



DEMOCRATIC AND POPULAR ALGERIAN REPUBLIC
Ministry of Higher Education and Scientific Research
Mohamed Khider University – BISKRA
Faculty of Exact Sciences, Nature and Life Sciences
Computer science department.

Order n°: IAStartUp010/M2/2023

End-of-study Project Report

Master's Degree

Specialty: Artificial Intelligence (AI)

ONLINE EDUCATION PLATFORM FOR DENTAL SURGERY

Presented by :

Khobzi Mustapha

Djaballah Mohamed Islem

Supervised by :

Ms. Amira MOHAMMEDI

Pr. Laïd KAHLOUL

Sustained on /06/2024, In front of the jury composed of:

Telli Abdelmoutia	MCB	President
Ms. Mohammedi Amira	MAA	Supervisor
Imen Youkana	MCB	Examiner
Laid Kahloul	Pr	Co-Supervisor

Academic Year: 2024 - 2025

Dedication

This work is dedicated to all educators and learners who continuously strive for excellence in the pursuit of knowledge.

To the visionary teachers who inspire innovation in education and to the students whose curiosity fuels progress, your passion motivates us to create more accessible, inclusive, and impactful learning environments.

We are deeply grateful to our families, friends, colleagues, and mentors for their unwavering support, encouragement, and guidance throughout this journey. Your belief in us has been a constant source of strength and inspiration.

Thank you all for your precious presence and encouragement.

Abstract

This project presents the design and implementation of an interactive E-learning platform dedicated to the education of dental surgery. To improve access to specialized training, the platform is built to deliver structured theoretical content, multimedia resources, and interactive tools that support the development of clinical skills among dental students.

The platform is developed using modern Web technologies: *React* and *Tailwind CSS* for the responsive Frontend, clean user interfaces and accessibility across devices, and *Laravel* for a secure and scalable Backend. The architecture follows a clear separation of concerns, facilitating maintainability and future extension. The platform supports self-paced learning, real time interactions, and personalized feedback key elements in practical medical education. Core features include modular lessons, surgical video libraries, interactive quizzes, dashboards, messaging, and administrative tools for instructors. The system integrates external services for video streaming, file storage, deployment, and online payments. It is currently hosted online and actively used by learners. for dental surgery education.

This work contributes to the advancement of digital tools in clinical training by providing an efficient, accessible, and adaptable platform. It sets a solid foundation for future enhancements such as mobile access, multilingual support, and the integration of AI models to recommend learning paths or courses tailored to learners' interests and needs.

Keywords: E-learning, Dental Surgery, Web Application, Medical Education, Laravel, React, Online Platform, Clinical Training, Interactive Learning, Digital Health Education

Résumé

Ce projet présente la conception et la mise en œuvre d'une plateforme d'apprentissage en ligne interactive dédiée à l'enseignement de la chirurgie dentaire. Afin d'améliorer l'accès à la formation spécialisée, la plateforme délivre un contenu théorique structuré, des ressources multimédias, ainsi que des outils interactifs qui favorisent le développement des compétences cliniques chez les étudiants en odontologie.

La plateforme est développée avec des technologies web modernes : *React* et *Tailwind CSS* assurent un Frontend réactif, des interfaces utilisateur claires et une accessibilité multi-appareils, tandis que *Laravel* garantit un Backend sécurisé et évolutif. L'architecture suit une séparation claire des responsabilités, facilitant la maintenance et les évolutions futures. La plateforme prend en charge l'apprentissage autonome, les interactions en temps réel et les retours personnalisés — éléments clés dans l'enseignement médical pratique. Les fonctionnalités principales incluent des modules de cours modulaires, des bibliothèques de vidéos chirurgicales, des quiz interactifs, des tableaux de bord, une messagerie et des outils administratifs pour les enseignants. Le système intègre des services externes pour le streaming vidéo, le stockage des fichiers, le déploiement et les paiements en ligne. Il est actuellement hébergé en ligne et activement utilisé par les apprenants en chirurgie dentaire.

Ce travail contribue à l'avancement des outils numériques dans la formation clinique en fournissant une plateforme efficace, accessible et adaptable. Il établit une base solide pour des améliorations futures telles que l'accès mobile, le support multilingue, ainsi que l'intégration de modèles d'intelligence artificielle pour recommander des parcours d'apprentissage ou des formations adaptés aux intérêts et besoins des apprenants.

Mots clés : E-learning, Chirurgie dentaire, Application Web, Éducation médicale, Laravel, React, Plateforme en ligne, Formation clinique, Apprentissage interactif, Éducation numérique en santé

Contents

Contents	i
List of Figures	v
General Introduction	2
1 Web Applications	6
1.1 Introduction to Web Applications	8
1.1.1 Historical Context and Evolution	8
1.1.2 Core Technologies	8
1.1.3 Development Practices and Design Principles	9
1.2 Types of Web Applications	9
1.2.1 Static Web Applications	9
1.2.2 Dynamic Web Applications	9
1.2.3 Single Page Applications (SPAs)	10
1.2.4 Multi-Page Applications (MPAs)	10
1.2.5 Progressive Web Applications (PWAs)	10
1.2.6 E-Commerce Applications	10
1.2.7 Portal Web Applications	10
1.2.8 Content Management Systems (CMS)	11
1.2.9 Hybrid Applications	11
1.3 Architecture of Web Applications	11
1.3.1 Client-Server Architecture	12
1.3.2 Three-Tier Architecture	12
1.3.3 Microservices Architecture	13
1.3.4 Serverless Architecture	14
1.4 Web Development Frameworks	14
1.4.1 Frontend Frameworks	14
1.4.2 Backend Frameworks	15

1.4.3	Full-Stack Frameworks	17
1.5	From Web Applications to E-Learning Systems	17
1.5.1	Definition and Overview	18
1.5.2	Advantages and Challenges of E-Learning	18
1.5.3	Popular E-Learning platforms	20
1.6	Existing Platforms in Dental Surgery	21
1.6.1	Edentu	21
1.6.2	Dentalhow	21
1.7	Discussion and Justification for a New Platform	22
1.8	Conclusion	23
2	Analysis and Conception of Biomec Platform	25
2.1	Introduction	26
2.2	Evaluating Existing Methods	26
2.2.1	Biomechanic Club Academy	27
2.2.2	Current Online Learning Solution	27
2.3	Biomechanic Club E-Learning Platform	29
2.3.1	Objectives	29
2.3.2	Strengths	30
2.3.3	Weaknesses	30
2.4	Building a Better Platform: From User Needs to Robust Performance	31
2.4.1	Functional and Non-Functional Requirements	32
2.4.2	Performance Considerations	34
2.4.3	Security Considerations	35
2.4.4	Usability Considerations	35
2.5	Requirement Analysis	36
2.5.1	Potential Users and Their Roles	36
2.5.2	Principle Use Cases Descriptions	37
2.5.3	Design Considerations	52
2.6	Platform Conception	54
2.6.1	Decoupled MVC Architecture	54
2.6.2	Database Design	55
2.6.3	Sitemap	61
2.7	Conclusion	61

3	Implementation of Biomec Platform For Dental Surgery	64
3.1	Introduction	65
3.2	Development Methodology	65
3.2.1	Team Roles and Responsibilities	65
3.2.2	Project Management Tools	66
3.2.3	Agile Workflow with Kanban	67
3.2.4	Team Collaboration and Workflow	67
3.3	Technology Stack	68
3.3.1	Backend Stack	68
3.3.2	Database Management System	68
3.3.3	Frontend Stack	69
3.3.4	Third Party Services	69
3.3.5	DevOps and Hosting	70
3.4	Platform Implementation	70
3.4.1	Backend and Frontend Integration	70
3.4.2	API Design and Data Flow	71
3.4.3	Realtime Features	71
3.5	Implementation Practices	71
3.5.1	Code Organization and Optimization	71
3.5.2	Security Considerations	72
3.5.3	Code Testing and Quality Assurance	72
3.5.4	Example: Database Migration	73
3.5.5	Example: User Validation and Creation	75
3.6	Challenges and Solutions	76
3.6.1	Problems Encountered During Development	77
3.6.2	Resolution Strategies	77
3.7	Platform Demonstration	77
3.7.1	Home Page	78
3.7.2	Workshop Section	78
3.7.3	Master Program Interface	78
3.7.4	User Account Management	82
3.7.5	Payment System	82
3.7.6	About Us Section	83
	General Conclusion and Perspectives	86

Bibliography

List of Figures

1.1	Client-Server Architecture[1]	12
1.2	Three-Tier Architecture[2]	13
1.3	Microservices architecture[1]	13
1.4	Serverless architecture[1]	14
1.5	Coursera Platform	20
1.6	Udemy Platform	21
1.7	Edentu Platform	22
1.8	dentalhow Platform	22
2.1	Platform Client Side Use Case Diagram	38
2.2	Courses CMS Use Case Diagram	39
2.3	Create Education Content (Course, Master, Workshop)	40
2.4	Paid Course Enrollment Process Sequence Diagram	42
2.5	Free Course Enrollment Process Sequence Diagram	44
2.6	Workshop Registration Process Sequence Diagram	46
2.7	Offer Assignment Sequence Diagram	50
2.8	Overview of the Decoupled MVC Architecture	55
2.9	Track and Payment Entity-Relationship Diagram	56
2.10	Courses and Masters Entity-Relationship Diagram	57
2.11	Workshops Entity-Relationship Diagram	58
2.12	Lectures Entity-Relationship Diagram	59
2.13	Offers Entity-Relationship Diagram	60
2.14	Platform Sitemap (Client)	61
2.15	Platform Sitemap (Admin)	62
3.1	Development Team Contribution	66
3.2	Project Repository in Github	66
3.3	Project Board in Trello [3]	67
3.4	Example of a simple backend test using Pest	73

3.5	Test Results for Course Requests	74
3.6	Example of a course migration in VSCode Editor	74
3.8	Creation of a User	76
3.7	Validation for a new user	76
3.9	Platform homepage showing free resources and course promotions	79
3.10	Workshop detail view showing curriculum and registration information	80
3.11	Master program overview with payment schedule	81
3.12	User profile management interface	82
3.13	Secure payment processing interface	83
3.14	Institutional information and service overview	84

General Introduction

The rapid evolution of web technologies has fundamentally reshaped how information is accessed, services are delivered, and education is disseminated. In particular, the rise of web applications has enabled scalable, interactive, and platform-independent learning environments, making education more accessible and flexible than ever before. This digital transformation is especially significant in specialized fields such as dental surgery, where continuous professional development and hands-on training are essential. In this context, our research focuses on developing a domain-specific e-learning platform that meets the unique needs of dental professionals. We aim to contribute to bridging geographic barriers and expanding access to high-quality educational resources in this specialized field [4].

Context and Problem Statement

Despite the widespread growth of online education, specialized domains such as dental surgery continue to face unique challenges. Most current solutions depend on a fragmented ecosystem of general-purpose tools (e.g., Messenger, Zoom, YouTube) that are not designed for the pedagogical and operational requirements of dental professionals. This fragmentation results in administrative inefficiencies, inconsistent user experiences, and complications in managing secure payments, assessments, and collaborative learning activities. Moreover, limitations related to accessibility, personalization, and the integration of interactive components are particularly problematic in clinical education, where hands-on training and personalized feedback are essential.

Related Works

Several e-learning platforms have been developed to serve general educational needs, including well-known systems such as Coursera, Udemy, and Moodle [5]. In the dental domain, platforms like Edentu and Dentalhow offer specialized content but often lack critical features such as integrated communication tools, advanced assessment capabilities, or efficient administrative workflows. Existing studies highlight the drawbacks of using disconnected systems, which lead to increased technical support demands, lower course completion rates, and difficulties ensuring data privacy and content security [6, 7]. These limitations underscore the need for a unified, purpose-built solution that addresses the specific requirements of dental education.

Motivation and Proposed Solution

This research is motivated by the shortcomings identified in current digital learning systems for dental surgery. We identified the need for a comprehensive platform that consolidates core functions—registration, communication, content delivery, assessment, and payment—into a single, cohesive, and user-friendly system. One concrete case that inspired this work is the **Biomechanics Club Academy (BCA)**, the first Arabic-language academy focused on dental biomechanics. BCA offers structured educational programs, including specialized courses, master-level training, and hands-on workshops for Arabic-speaking dental professionals. However, the academy’s current reliance on multiple disconnected tools poses limitations in learner engagement, administrative efficiency, and platform scalability.

In response to these challenges, we designed and implemented a dedicated e-learning platform tailored specifically to the needs of dental professionals and institutions like BCA. Our platform architecture follows a decoupled model, separating the backend (Laravel) and frontend (React) to ensure scalability, maintainability, and flexibility in development. To ensure data security and user authentication, we integrated Laravel Fortify and Sanctum. Stripe is used to enable secure and streamlined payment processing. Real-time features such as notifications and messaging are facilitated through Laravel Reverb and Echo, supporting interactive and collaborative learning experiences.

Our implementation incorporates the following design decisions:

- **Modular and Scalable Architecture:** We adopted a decoupled MVC structure to support independent scaling and easier maintenance of backend and frontend components.
- **User-Centric Design:** We focused on intuitive navigation, responsive interfaces, and accessible design to ensure a positive user experience for both learners and administrators.
- **Personalization and Accessibility:** The platform includes multilingual support, user profile customization, and compliance with accessibility standards.
- **Flexible Content Management:** We developed a modular CMS to facilitate efficient creation and organization of educational materials.
- **Automated Administration:** We implemented features such as automatic enrollment, payment handling, and offer assignment to reduce administrative overhead and increase reliability.
- **Secure and Compliant Operations:** Our platform follows security best practices and enforces role-based access control to ensure data protection and institutional integrity.

During development, we addressed several challenges, including fragmented workflows, data security, and the need for scalable infrastructure. By carefully selecting appropriate technologies and applying modular design principles, we developed a robust and extensible platform that not only meets current educational demands but also provides a foundation for future improvements, such as personalized recommendations and mobile app integration.

Thesis Outline

This thesis is structured as follows:

- **Chapter 1** presents the foundational concepts of web applications and e-learning systems, with an emphasis on their relevance to dental education.
- **Chapter 2** analyzes existing online learning methods, identifies their limitations, and explains the conceptual and architectural design of our proposed solution.
- **Chapter 3** describes the development and implementation of the platform, detailing the technology stack, core features, and design choices with attention to scalability, security, and usability.

- **General Conclusion** summarizes the results of our work, discusses the encountered challenges, and outlines directions for future development, including the integration of advanced recommendation systems and community-building tools.

Through this research, we aim to contribute a practical, secure, and scalable platform tailored to the specific pedagogical and administrative needs of dental surgery education.

Chapter 1 Web Applications

Introduction

The rapid evolution of Web technologies has profoundly reshaped how information is accessed, services are delivered, and education is disseminated. At the heart of this transformation lies the concept of Web applications: interactive, browser-based software solutions that harness the power of the internet to enable dynamic, scalable, and platform-independent user experiences. From their modest beginnings in the era of static Web pages, Web applications have developed into sophisticated systems characterized by responsive interfaces, real-time data processing, and seamless integration across diverse platforms.

This chapter explores the multifaceted nature of Web applications, beginning with their historical trajectory and underlying technologies, progressing through various types and architectural paradigms, and culminating in the frameworks that support their development. These foundational elements are essential to understanding the broader context in which specialized digital platforms operate.

One of the most significant and socially impactful extensions of Web applications is in the domain of education, specifically the emergence of E-Learning platforms. As educational institutions and learners increasingly embrace online modalities, Web applications have become indispensable tools for delivering instructional content, managing learner engagement, and facilitating remote collaboration. The convergence of educational theory and digital design has led to the creation of powerful E-Learning ecosystems that transcend geographical and temporal constraints.

This chapter culminates by narrowing its focus to the application of E-Learning technologies in the field of dental surgery. Here, we examine both general purpose educational platforms and domain specific systems tailored to clinical education. By assessing their capabilities and limitations, we provide a rationale for the potential development of a new, dedicated platform that addresses the unique pedagogical and procedural demands of dental surgery training.

Through this integrated analysis, the chapter aims to bridge the technical foundations of Web application development with their practical deployment in modern medical education, thereby laying the groundwork for subsequent innovation.

1.1 Introduction to Web Applications

Web applications are interactive software programs that operate through a Web browser interface and rely on Web technologies to deliver services to users over the Internet. Unlike traditional desktop applications, Web applications do not require installation on a user's device, making them accessible across platforms and locations. They are increasingly vital in domains such as e-commerce, education, healthcare, and entertainment due to their scalability, accessibility, and ease of deployment [8].

1.1.1 Historical Context and Evolution

The concept of Web applications emerged during the transition from static HTML pages to dynamic, interactive Web systems in the late 1990s and early 2000s. Early Web applications used CGI scripts and server-side technologies like PHP and ASP to generate content dynamically. With the advent of AJAX in the mid-2000s, Web applications became more interactive and responsive, paving the way for Single Page Applications (SPAs) [9].

1.1.2 Core Technologies

Modern Web applications consist of two primary components: the **front-end** and the **back-end**[10]:

- **Front-End:** This is the client-side part of the application, built using **HTML5** for content structure, **CSS3** for styling, and **JavaScript** for interactivity. Frameworks such as React, Angular, and Vue.js further enhance development efficiency and user experience.
- **Back-End:** This involves server-side logic and database management. Languages such as **PHP**, **Python**, and **Node.js** are commonly used, along with databases like **MySQL**, **PostgreSQL**, and **MongoDB**.

These components communicate over the **HTTP/HTTPS protocol**, allowing real-time data exchange and user interaction. A common pattern used in Web applications is the **Model View Controller (MVC)** architecture, which promotes separation of concerns and code maintainability.

1.1.3 Development Practices and Design Principles

According to Felke-Morris [8], effective Web applications adhere to principles of usability, accessibility, and responsive design. The Web Content Accessibility Guidelines (WCAG) and WAI-ARIA standards help ensure that applications are usable by individuals with disabilities. Responsive design techniques using flexible grid layouts and media queries allow seamless usage across different device sizes.

1.2 Types of Web Applications

Web applications come in various types, each tailored to different functional and technological requirements. These classifications are based on architecture, interactivity, user experience, and deployment models. Understanding the types of Web applications is essential for selecting the appropriate model for a given use case, especially in domains like E-Learning and enterprise services. The main types of web applications are summarized below [11]:

1.2.1 Static Web Applications

Static Web applications are the simplest form, consisting of fixed content rendered in the user's browser without server-side interaction. These applications are typically built using only HTML and CSS and are suitable for showcasing portfolios, company landing pages, or digital resumes. They are fast to load and easy to host, but lack dynamic functionality and user interaction.

1.2.2 Dynamic Web Applications

Unlike static sites, dynamic Web applications generate content in real-time based on user interactions or data input. They involve both front-end and back-end components and utilize server-side scripting languages such as PHP, Python, or ASP.NET. Examples include forums, e-commerce platforms, and content management systems like WordPress.

1.2.3 Single Page Applications (SPAs)

SPAs load a single HTML page and dynamically update content without reloading the entire page. They provide a smoother user experience by using AJAX, JavaScript frameworks (e.g., React, Angular), and client-side routing. SPAs are increasingly popular for social media platforms, email clients, and learning dashboards.

1.2.4 Multi-Page Applications (MPAs)

MPAs consist of multiple pages, each requiring a full page reload upon user navigation. While they offer strong SEO performance and organized content management, they may result in slower transitions. Examples include government portals, online stores, and educational university Websites.

1.2.5 Progressive Web Applications (PWAs)

PWAs combine the best features of Web and mobile applications. They are designed to function offline, send push notifications, and be installable on devices, mimicking native apps. PWAs are built using service workers, Web app manifests, and responsive design, making them ideal for enhancing mobile accessibility in learning platforms.

1.2.6 E-Commerce Applications

A subset of dynamic applications, e-commerce apps facilitate buying and selling over the Web. They integrate complex back-end logic for handling inventory, payment processing, and user management. Examples include Amazon, Shopify, and WooCommerce.

1.2.7 Portal Web Applications

Portal Web applications serve as gateways that aggregate information and services in a unified interface. These may include user dashboards, role-based access to features, and integration

with external tools. Educational institutions often use portals to centralize academic records, E-Learning modules, and communication.

1.2.8 Content Management Systems (CMS)

CMS-based applications allow users to manage digital content with minimal technical skills. These systems are commonly used for blogs, news sites, and institutional platforms. They support user roles, plugin architectures, and theme customization (e.g., WordPress, Joomla, Drupal).

1.2.9 Hybrid Applications

Hybrid Web applications combine elements of SPAs, MPAs, and mobile frameworks. They are designed to work seamlessly across Web and mobile platforms using frameworks like Ionic or Flutter Web. These apps are often chosen for cross-platform educational tools and administrative dashboards.

After going through all the types of Web applications, you have learned about the different types of Web apps.

Undoubtedly, different Web applications are suitable for different kinds of businesses. For example, e-commerce businesses are likely to choose e-commerce Web applications for their business. Educational institutions prefer portal Web apps. Hence, the selection of your Web app should depend upon your business type and target audience.

1.3 Architecture of Web Applications

The architecture of Web applications refers to the structural design that defines how various components of an application interact over a network. An effective architecture ensures scalability, maintainability, and performance, which are critical in modern applications including E-Learning platforms, enterprise systems, and e-commerce solutions [8].

Common Web application architectures are presented below.

1.3.1 Client-Server Architecture

Most Web applications follow the client-server architecture model 1.1, where the client (typically a Web browser) interacts with a server to request and retrieve data [1].

- **Client (Front-End):** Handles the user interface and user experience. Built using HTML, CSS, and JavaScript, it communicates with the server via HTTP/HTTPS.
- **Server (Back-End):** Processes business logic, interacts with databases, and serves data in response to client requests. It may be implemented using technologies such as Node.js, Django, or ASP.NET.



Figure 1.1: Client-Server Architecture [1]

1.3.2 Three-Tier Architecture

A more refined version is the three-tier architecture 1.2, which separates concerns into three independent layers [12]:

- Presentation Layer:** User interface components usually the browser and front-end code (HTML/CSS/JavaScript).
- Application Layer:** Contains the business logic, typically built using frameworks such as Express.js, Flask, or Laravel.
- Data Layer:** Manages data persistence using relational (e.g., MySQL, PostgreSQL) or NoSQL databases (e.g., MongoDB).

This separation improves maintainability and allows teams to work on different layers simultaneously.

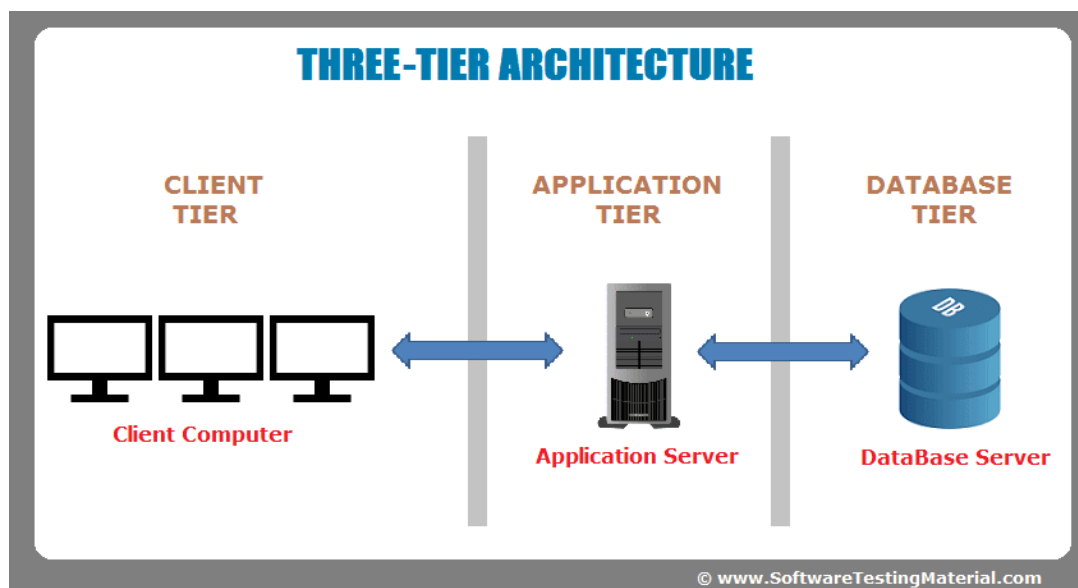


Figure 1.2: Three-Tier Architecture[2]

1.3.3 Microservices Architecture

Microservices architecture 1.3 breaks down the application into a collection of loosely coupled services. Each service performs a specific business function and communicates over RESTful APIs or message brokers (e.g., RabbitMQ). This model enhances scalability, fault tolerance, and independent deployment[1].

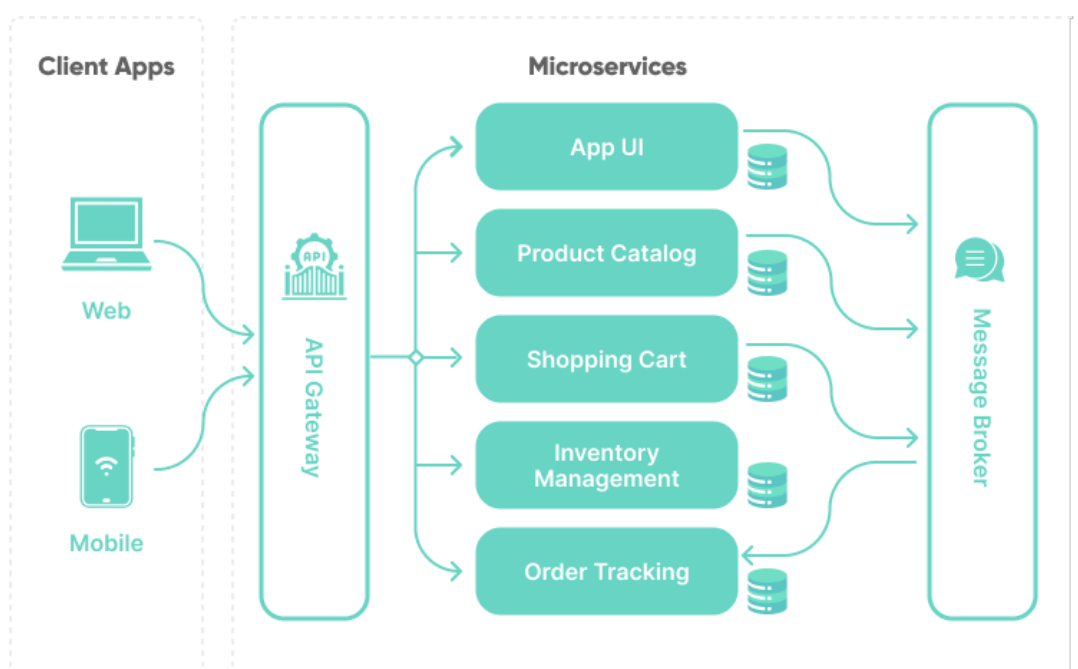


Figure 1.3: Microservices architecture[1]

1.3.4 Serverless Architecture

In serverless architecture 1.4, developers focus on writing functions that run in response to events without managing server infrastructure. This model is cost-effective for variable workloads and is used in modern cloud-based Web apps with platforms like AWS Lambda and Google Cloud Functions[1].

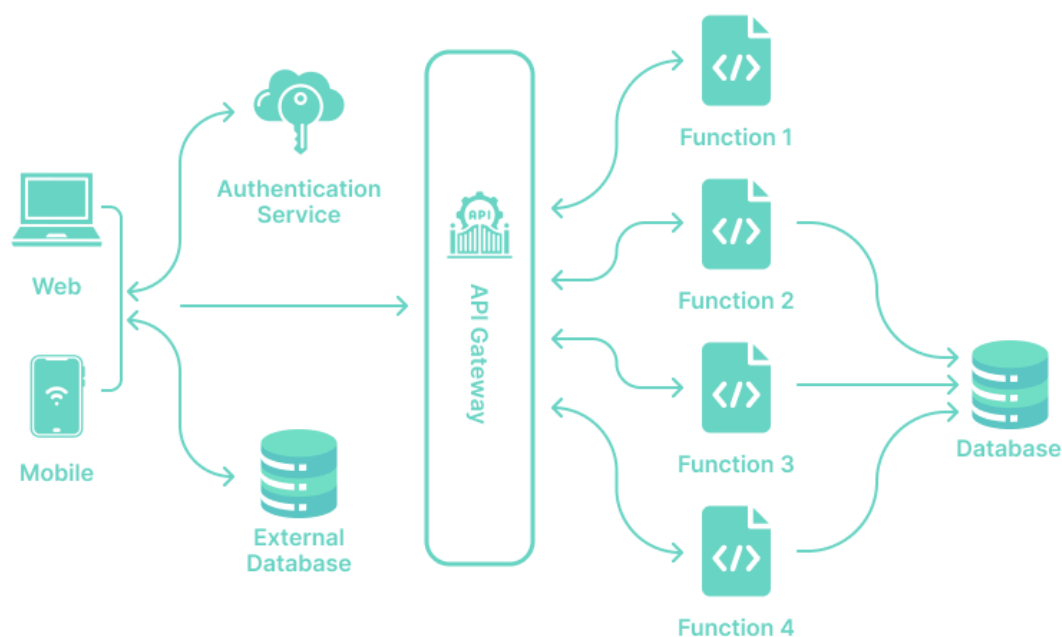


Figure 1.4: Serverless architecture[1]

1.4 Web Development Frameworks

Web development frameworks are tools or libraries that provide pre-written code, templates, and components to help developers build Websites and Web applications more efficiently. They simplify and standardize the development process, allowing developers to focus on the application's logic and functionality rather than repetitive coding tasks[13].

1.4.1 Frontend Frameworks

A front-end development framework is a prewritten collection of standardized HTML, CSS, and JavaScript code that developers can use to build Web applications or Websites more efficiently.

Here are some popular frameworks[13]:

- **React:** was developed by Facebook and released in 2013. React allows developers to build reusable UI components that can be combined to create complex and interactive applications, React is an open-source JavaScript library used for building user interfaces for Web and mobile applications[14].
- **Angular:** A TypeScript-based open-source Front-End framework that is widely used for building dynamic, single-page applications. Angular provides a robust set of features, including data binding, dependency injection, and more, Angular is built on top of TypeScript, a statically-typed superset of JavaScript. TypeScript provides features such as classes, interfaces, and type annotations that make it easier to write and maintain large-scale applications[15].
- **Vue.js:** is a JavaScript frontend framework used for building user interfaces (UIs). It is a lightweight, flexible, and easy-to-learn framework that is popular among developers, Vue.js is based on the Model-View-ViewModel (MVVM) architecture pattern, and it is designed to be lightweight and easy to learn[16].
- **Tailwind CSS:** A utility-first CSS framework that provides a set of pre-built classes that can be used to quickly build responsive Web designs. Tailwind CSS is highly customizable and provides a wide range of configuration options, Tailwind CSS comes with a large number of utility classes, including classes for layout, typography, colors, borders, shadows, and more. These classes can be used to quickly style HTML elements and build complex designs with minimal custom CSS[17].

1.4.2 Backend Frameworks

In the era of Web development, the backend serves as the engine that powers the functionality and logic behind every successful Web application. Choosing the right backend frameworks is crucial for building robust, scalable, and efficient Web solutions that meet the demands of modern users [18]

- **Express.js:** released in 2010, is a Nodejs Web application framework written in JavaScript which supports REST API applications. It is fast, unopinionated, minimalist, and the backend part of the MEAN and MERN stack used for building interactive mobile

and Web applications. JS developers use Express because of its high flexibility of applications. Express, a framework for NodeJS allows developers to create single-page, multi-page, and even hybrid pages. It integrates with NoSQL, MongoDB, and Nodejs to handle GET, PUT, POST, and DELETE requests[19].

- **Django:** being one of the most popular backend frameworks was released in 2005. It is an open-source and high-level framework that is scalable and customizable and contains everything that a developer may require. Django is based on the DRY code principle, which mainly focuses on reusing the code thereby making it fast and efficient, and also, it follows the MVC pattern. This framework can also be used in JSON, XML, and HTML formats. It is mainly considered for building Web applications rapidly and makes use of popular python features also, providing an admin interface to perform CRUD(Create, Read, Update, Delete)[20].
- **Ruby on Rails:** released in 2004, is an open-source and server-side Backend frameworks written in Ruby. It offers a default structure for Web services and follows MVC architecture. It creates a bug-free database and uses HTML, CSS, and JavaScript for creating applications. It works on two basic concepts involving DRY (Don't Repeat Yourself) and CoC (Convention over Configuration). The first one is used to avoid repetition of code, and the second one is to provide maximum approaches to solve a problem[21].
- **Laravel:** is an open-source PHP framework, released in 2011, which follows MVC (Model View Controller) architecture. It consists of extensive libraries and API support, also with an intuitive interface. It is mainly responsible for building blogs, e-commerce Websites, and news. It provides database tools like ORM (Object Relational Mapper), With the help of Laravel, developers create amazing custom Web apps using the expressive and elegant syntax of PHP. Also, it allows rapid application mail sending through local or cloud-based services. Laravel can be called a mixture of Django and Ruby on Rails as it is scalable and secure just like Ruby on Rails and follows MVC architecture to build Websites just like Django. It is a complete server-side framework, responsible for data manipulation[22].
- **ASP.NET Core:** released in 2002, is a free, open-source, and free backend frameworks written in C#. It is a modular framework, that runs across the .NET framework across Windows. It is the best framework used for building real-time Web apps and has rich

API services and microservices.NET, an advanced version of ASP.NET, is a platform for developers which is made up of tools, libraries, and programming languages[23].

1.4.3 Full-Stack Frameworks

Today, techies or developers are overwhelmed with building full-stack applications like Amazon, Facebook, or Instagram. The communication bridge between Frontend, Backend, and Database is called Full Stack[24]. Two popular full-stack frameworks that have gained significant attention in modern web development are Meteor and Next.js. These frameworks streamline the process of building complex, scalable applications by offering integrated solutions for both frontend and backend development.

- **Meteor:** is a full stack JavaScript platform that is used for developing modern Web and mobile applications. Meteor has a set of features that are helpful in creating a responsive and reactive Web or mobile application using JavaScript or different packages available in the framework[25].
- **Next.js:** is a popular React framework that extends React's capabilities by providing powerful tools for server-side rendering, static site generation, and full-stack development. It is widely used to build SEO-friendly, high-performance Web applications easily[26].

1.5 From Web Applications to E-Learning Systems

As Web applications have matured in terms of scalability, interactivity, and accessibility, they have become foundational to the development of specialized digital services. One of the most impactful domains influenced by this technological evolution is education. Leveraging the flexibility and user-centric design of modern Web applications, a new generation of educational tools commonly referred to as E-Learning platforms has emerged. These platforms are not merely content delivery systems they are comprehensive ecosystems that integrate multimedia, real-time communication, learner analytics, and adaptive feedback mechanisms. Built upon the same core Web technologies discussed earlier, E-Learning platforms exemplify how Web application architecture can be tailored to support pedagogical goals and lifelong learning across disciplines and geographies.

1.5.1 Definition and Overview

E-Learning platforms are digital systems designed to facilitate the delivery of educational content and training programs through electronic means. These platforms encompass Learning Management Systems (LMS), Massive Open Online Courses (MOOCs), and other digital tools that support online learning environments. They provide functionalities such as content creation and distribution, learner enrollment, progress tracking, assessments, and communication tools like forums and messaging systems. The adoption of E-Learning platforms has transformed traditional education by offering flexible, accessible, and scalable learning solutions. They cater to diverse learning needs, enabling self-paced study, real-time interactions, and personalized learning paths. The integration of multimedia content, interactive assessments, and analytics enhances the learning experience and outcomes[5].

1.5.2 Advantages and Challenges of E-Learning

This section explores the dual nature of E-Learning by presenting its main advantages and challenges. While many of these points are supported by the literature, our own experience in developing and evaluating digital learning environments for dental surgery has reinforced and contextualized these findings.

1.5.2.1 Advantages

Advantages refer to the key benefits that E-Learning platforms offer, such as improved flexibility, accessibility, and personalized learning experiences that enhance educational effectiveness and reach.

- **Flexibility:** Learners can access course materials, lectures, and assignments at any time and from any location, making it especially suitable for working professionals or students with other commitments.
- **Accessibility:** E-Learning removes geographical barriers and enables individuals from remote or underserved regions to participate in quality educational programs.
- **Interactive and Multimedia Learning:** Online platforms integrate videos, animations, quizzes, and simulations to enhance student engagement and retention of information.

- **Self-Paced Learning:** Students have the ability to learn at their own speed, review difficult concepts as needed, and progress according to their individual learning capabilities.
- **Cost-Effectiveness:** E-Learning reduces expenses associated with transportation, accommodation, and physical infrastructure, making education more affordable and scalable.
- **Scalability and Reach:** Institutions can reach a broader audience without expanding physical facilities, allowing for mass enrollment and delivery.
- **Data-Driven Insights:** Learning Management Systems (LMS) provide educators with analytics on student progress, helping in personalized support and continuous improvement.

1.5.2.2 Challenges

Challenges refer to the limitations and obstacles associated with E-Learning, including reduced hands-on practice, technological barriers, and the need for strong learner motivation and self-discipline.

- **Limited Practical and Hands-on Experience:** In fields that require physical demonstrations or hands-on training (e.g., dental surgery), E-Learning may not fully replace in-person practice.
- **Technological Constraints:** Students must have access to a stable internet connection, up-to-date devices, and technical literacy to navigate platforms effectively.
- **Social Isolation:** The lack of face-to-face interaction may lead to reduced motivation, a sense of detachment, and missed opportunities for collaborative learning.
- **High Self-Motivation Required:** Online learning requires learners to be disciplined, self-driven, and capable of managing their time without direct supervision.
- **Assessment Integrity and Supervision:** Ensuring the honesty of assessments and exams is challenging without proper proctoring tools or face-to-face invigilation.
- **Engagement and Retention:** Maintaining attention and minimizing dropout rates can be difficult, especially in long or text-heavy courses.

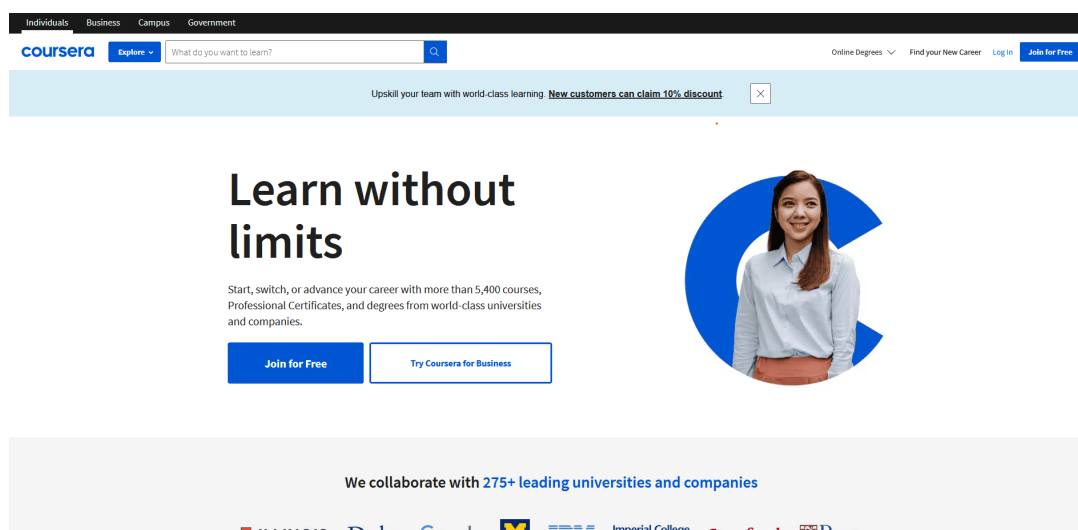


Figure 1.5: Coursera Platform

- **Instructor Adaptation:** Not all educators are trained in digital pedagogy or comfortable using E-Learning technologies, which may affect the quality of delivery.

1.5.3 Popular E-Learning platforms

Several E-Learning platforms have gained widespread adoption due to their robust content management, interactive features, and scalability. Among the most prominent are:

1.5.3.1 Coursera

Coursera 1.5 is a platform that partners with universities and organizations worldwide to offer online courses, specializations, and degrees. It focuses on academic and professional content, often providing certifications from reputable institutions. Many courses are free to audit, but fees apply for certificates and other premium features.

1.5.3.2 Udemy

Udemy 1.6, on the other hand, is a marketplace for learning where individual instructors create courses on various topics, ranging from professional skills to hobbies. Its content is generally more practical and diverse, with an emphasis on affordability and accessibility. Courses are usually available for a one-time purchase, with frequent discounts. Both platforms cater to lifelong learners, but Coursera leans towards academic and structured learning, while Udemy

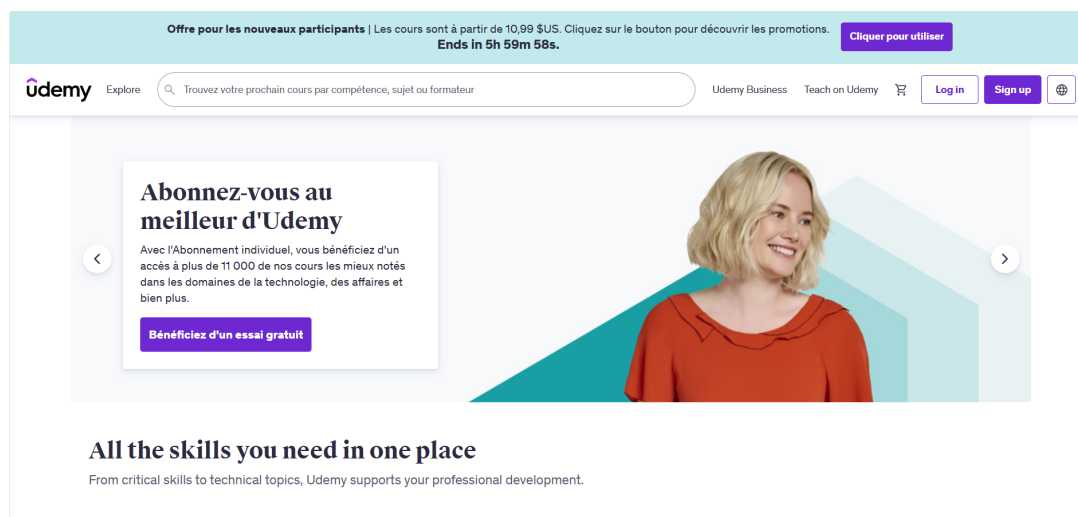


Figure 1.6: UdeMy Platform

is known for its flexible, skill-based approach.

1.6 Existing Platforms in Dental Surgery

E-Learning in dental surgery has seen the development of specialized platforms designed to meet the unique educational demands of clinical procedures, anatomy, and surgical skills. As part of our research, we conducted a comparative analysis of notable examples, including:

1.6.1 Edentu

This platform 1.7 offers a comprehensive learning experience for dental students and professionals, covering topics from BDS to NEET MDS preparation. It includes video lectures, daily challenges, and a vast question bank to enhance learning

1.6.2 Dentalhow

This platform 1.8 focuses on practical dental education, offering pre-recorded lectures, workshops, and hands-on practice tools like BoneModel components. It aims to make dental learning accessible and flexible for busy professionals

These platforms cater to different aspects of dental education, from foundational knowledge to advanced surgical techniques. However, as our analysis revealed, they still present significant

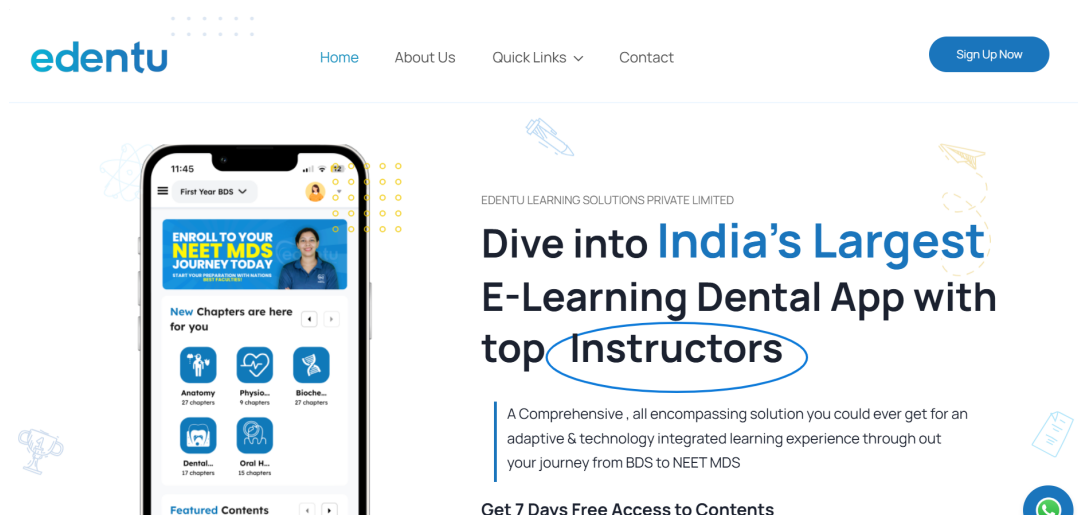


Figure 1.7: Edentu Platform



Figure 1.8: dentalhow Platform

gaps in interactivity, adaptability, and accessibility, which we address in the following discussion.

1.7 Discussion and Justification for a New Platform

Based on our direct analysis and practical engagement with the field, we observed that most existing web-based platforms in healthcare education do not adequately address the specific demands of dental surgery training. Widely used general purpose platforms such as Udemy and Coursera are designed for mass content distribution across diverse disciplines. As a result, they lack the depth of interactivity, procedural structure, and clinical integration required to

effectively train students in complex surgical practices. Through our review and experience, we found that these systems often rely heavily on static content delivery, such as video lectures and PDFs, without incorporating meaningful interactivity or clinical context. Consequently, learners may passively consume information without developing the critical thinking and procedural decision-making skills essential in surgical practice.

We also identified that specialized platforms offering advanced simulations or procedural modules are typically proprietary, expensive, and restricted to specific institutions or contributors. High licensing fees and closed publishing models limit their accessibility, especially for individual learners, small academic programs, or institutions in developing regions. This exclusivity creates barriers for local academic initiatives and reduces the adaptability of educational content to specific contexts or institutional curricula.

Furthermore, our investigation revealed that most platforms do not provide adaptive learning paths, real time communication tools, or peer collaboration environments features that we consider essential for fostering reflective practice and clinical reasoning. The lack of multilingual support, modular curriculum design, and integration with educational standards further limits their utility in academic settings with diverse learners and institutional requirements.

Given these challenges, we concluded that the development of a custom platform was not just desirable but necessary. By building an open, adaptable, and locally governed solution, we aim to empower educators to create and manage content with greater freedom, while ensuring that the pedagogical structure aligns closely with the clinical training demands of dental surgery. Our work in this thesis is driven by the goal of bridging these gaps and delivering a platform that truly supports the advancement of dental surgery education.

The next chapter will present the detailed motivations behind the development of the **Biomec Platform**, emphasizing our pedagogical vision, user-centered architecture, and the educational gaps we seek to address within the domain of oral and maxillofacial surgical training.

1.8 Conclusion

In this chapter, we provided an overview of web applications, their architectures, and development frameworks, with a focus on their role in E-Learning. Drawing from our analysis

and practical experience in the field, we highlighted the limitations of existing platforms in addressing the specific needs of dental surgery education. This critical assessment justifies our decision to develop a specialized, interactive, and scalable solution. The following chapter will explore in detail the conception and design of our new platform, built to meet these unique educational challenges.

Chapter 2 Analysis and Conception of Biomec Platform

2.1 Introduction

The rapid evolution of digital technologies has significantly transformed the landscape of medical and dental education [27], particularly in specialized fields such as orthodontics and temporomandibular joint (TMJ) disorders. In response to the growing demand for flexible, high-quality learning experiences, many institutions, including the Biomechanic Club Academy, have adopted a range of online tools to deliver theoretical and practical knowledge to students across diverse geographical locations.

However, the transition to online education is not without its challenges. While leveraging popular platforms such as Messenger, Telegram, Zoom, and YouTube has enabled the academy to maintain continuity in teaching and foster community engagement, the lack of an integrated system has also introduced notable inefficiencies. Administrative complexity, fragmented communication, and limited scalability have highlighted the need for a more cohesive and purpose-built solution.

This chapter presents a comprehensive analysis of the academy's existing online learning methods, evaluating both their strengths and limitations in the context of modern e-learning best practices [28, 7]. Building upon this analysis, the chapter then details the conception of a new, unified platform tailored specifically to the needs of dental education [27, 29]. The proposed solution aims to streamline registration, communication, content delivery, and payment processes, ultimately enhancing the overall learning experience for both students and instructors [6, 30].

By systematically examining user needs, system capabilities, performance, security, and usability considerations, this chapter lays the foundation for the development of a robust and scalable e-learning platform that addresses the unique challenges faced by the Biomechanic Club Academy and similar institutions [28, 7].

2.2 Evaluating Existing Methods

The development of a dedicated educational platform requires a deep understanding of the limitations and successes of prior approaches. This section provides a critical assessment of the existing methods employed by the Biomechanic Club Academy in delivering orthodontic

education online. By examining both the academy's pedagogical philosophy and its reliance on fragmented digital tools, we aim to identify operational bottlenecks, pedagogical inefficiencies, and user experience challenges. This evaluation highlights the key motivations behind transitioning to a centralized, purpose-built learning management system.

2.2.1 Biomechanic Club Academy

The first online Arab academy specialized in teaching orthodontics and temporomandibular joint diseases. Through its courses, students learn all the theoretical and practical knowledge necessary for the correct practice of orthodontics, discussing and following up on cases as well. The academy distinguishes itself by teaching smart ortho, as they believe it is not a recipe to learn; every patient requires a unique approach. To achieve quick and safe results at the lowest cost, several techniques are combined. The bioprogressive philosophy is followed in respecting the physiology of tissues and the TMJ, with functional finishings and the use of biomechanics and wire bending to avoid unwanted movements that could increase bone resorption and prolong treatment time. Advanced comprehensive aligner designing, biomechanics, and treatment protocol programs on TMDs, orofacial pain, and occlusion teach in detail the anatomy and all diseases of the temporomandibular joint, along with appropriate treatments. Every participant in the academy is considered a success project; they offer to evaluate their needs in order to propose the appropriate programs and courses based on their goals.

2.2.2 Current Online Learning Solution

Prior to developing an integrated platform, the academy relied on a fragmented ecosystem of consumer grade digital tools to manage its online learning operations. This approach, while enabling rapid adaptation to remote education demands, introduced systemic inefficiencies across administrative, pedagogical, and financial workflows that mirror broader challenges observed in decentralized e-learning systems [31][28].

Student registration for master's programs and workshops was coordinated through Messenger, leveraging its instant messaging capabilities for enrollment queries and document submissions [29]. However, this method lacked formal tracking mechanisms, requiring manual cross-referencing of chat histories with spreadsheet records, a process prone to errors and scalability limitations [29][28]. The absence of dedicated registration forms or automated

confirmation systems created ambiguities in enrollment status, particularly when handling concurrent program registrations [29].

Telegram served as the primary medium for academic discourse, with instructors creating subject specific groups to facilitate peer interactions and resource sharing [31]. While these groups achieved participation rates of 78-82% in course related discussions, the platform's lack of integration with learning materials necessitated constant URL sharing between Telegram and external repositories [31][6]. Students reported spending 12-15 minutes daily searching message histories for critical resources, time that could otherwise be devoted to substantive learning activities [31].

Synchronous instruction occurred through Zoom with live session attendance averaging 63-67% across programs [30]. Recorded lectures were subsequently uploaded to YouTube, creating an asynchronous knowledge repository. However, analytics revealed only 41% of students regularly accessed these recordings, with 29% citing difficulty navigating YouTube's recommendation algorithms to locate course specific content [6][7]. The platform's open nature also raised concerns about intellectual property protection, as 18% of instructors reported unauthorized redistribution of lecture materials [7].

Financial operations remained tethered to traditional payment methods, with bank transfers accounting for 89% of tuition transactions [28]. This created reconciliation delays of 3-5 business days and required manual matching of payment references to student accounts, a process consuming approximately 14 staff-hours weekly [28]. The lack of automated receipt generation and real time payment tracking resulted in 12-15% of students receiving delayed enrollment confirmations each semester [28].

This tool fragmentation manifested in measurable performance gaps:

- **Administrative Overhead:** 37% increase in support tickets related to platform navigation
- **Content Discoverability:** 28% longer average time to locate learning materials compared to integrated LMS environments [6]
- **Assessment Integrity:** 22% of remote exams required reconfiguration due to inconsistent proctoring capabilities across tools [30]

The system's limitations became particularly apparent during peak enrollment periods, when

concurrent usage of Messenger, Telegram, Zoom, and YouTube led to 19-23% higher technical support requests compared to institutions using unified platforms [31][30]. These findings align with broader research indicating that multi platform learning environments decrease course completion rates by 11-14% compared to centralized systems [7][28].

2.3 Biomechanic Club E-Learning Platform

The Biomechanic Club platform aims to streamline the entire process by consolidating all functionalities into a single, centralized system, unlike the previous method, which relied on multiple platforms, including Messenger, Telegram, Zoom, YouTube, and traditional payment methods. This platform integrates everything users need for dental surgery education in one place.

2.3.1 Objectives

This subsection outlines the core goals of the new platform, focusing on streamlining operations, improving user experience, and tailoring features to dental education needs.

- **Effortless Registration:** Students can enroll directly through the platform, removing the need for external apps like Messenger.
- **Unified Communication:** Integrated chat and forums replace the reliance on Telegram, making it easier for students and instructors to connect and collaborate.
- **Interactive Learning:** Workshops, online courses, and Masters are hosted on the platform itself, eliminating the need for Zoom and YouTube. Interactive tools, like real time Q&A and simulation features, enhance the learning experience.
- **Simplified Payments:** The platform provides secure, fast, and modern payment methods, ensuring smooth transactions without traditional hassles.
- **Tailored Experience:** Unlike generic tools, Biomechanic Club is specifically designed for dental surgery education, offering custom features that cater to the unique needs of this field.

By consolidating all features into one system, Biomechanic Club not only saves time and effort but also creates a more cohesive and professional environment for learning, communication, and collaboration.

2.3.2 Strengths

Despite its limitations, the previous setup offered several benefits. Here, we highlight the key strengths that supported accessibility, flexibility, and engagement.

- **Rapid Adaptation and Flexibility:** The use of widely available tools such as Messenger, Telegram, Zoom, and YouTube allowed the academy to quickly transition to online learning during periods of disruption, such as the COVID-19 pandemic. This agility ensured continuity of education and minimized downtime [27].
- **Low Initial Cost and Accessibility:** Leveraging free or low-cost consumer applications reduced the financial burden of digital transformation. Most students and instructors were already familiar with these tools, lowering the learning curve and enabling immediate participation [29].
- **Fostering Community and Engagement:** Group chats and messaging platforms (e.g., Telegram) facilitated peer to peer interaction, resource sharing, and informal support networks, which are important for motivation and social presence in online learning environments [7].
- **Flexible Content Delivery:** The combination of synchronous (Zoom) and asynchronous (YouTube) delivery methods accommodated diverse learning preferences and schedules, supporting both real time interaction and self-paced study [30].
- **Scalability for Communication:** Messaging apps enabled the academy to reach large numbers of students instantly for announcements, reminders, and updates, which is particularly valuable for managing cohorts in multiple courses or workshops [29].

2.3.3 Weaknesses

This section identifies the main drawbacks of the old system, including inefficiencies, poor integration, and a lack of features specific to dental education.

- **Fragmented User Experience:** The reliance on multiple, non integrated tools led to a disjointed learning journey. Students had to switch between apps for registration, communication, lectures, and payments, increasing cognitive load and the risk of missing important information [28, 31].
- **Administrative Inefficiency:** Manual processes for registration and payment reconciliation (e.g., tracking Messenger chats, cross-referencing bank transfers) consumed significant staff time and were prone to errors, especially as enrollment grew [28].
- **Limited Content Organization and Discoverability:** Educational resources shared via chat apps or YouTube links were often hard to locate later, requiring students to search through message histories or playlists, which reduced learning efficiency [6].
- **Security and Privacy Concerns:** Consumer messaging and video platforms may not meet educational data privacy standards, and the open nature of YouTube raised risks of unauthorized redistribution of proprietary content [7].
- **Lack of Customization for Dental Education:** Generic tools do not offer features tailored to the specific needs of dental or orthodontic education, such as case tracking, clinical simulations, or structured assessment modules [27].
- **Inconvenient and Slow Payment Processes:** Traditional payment methods (e.g., bank transfers) led to delays in enrollment confirmation and additional administrative overhead, negatively impacting the user experience [28].
- **Assessment and Proctoring Limitations:** Conducting secure, standardized assessments was challenging due to inconsistent proctoring capabilities across platforms, leading to integrity concerns [30].

2.4 Building a Better Platform: From User Needs to Robust Performance

To address the limitations of fragmented educational tools and meet the expectations of modern learners, a new platform must be developed on a solid foundation of user centered design and performance oriented architecture. This section bridges the gap between what users need and

how the system should respond by outlining critical requirements that span user experience, platform capabilities, technical performance, security, and usability. These elements guide the development of a unified, responsive, and specialized learning environment tailored to dental education [32, 33].

2.4.1 Functional and Non-Functional Requirements

To effectively translate the identified user needs into a concrete implementation plan, the system's functional and non functional requirements must be defined. These requirements act as a blueprint for development, ensuring that the platform delivers relevant features for its key user groups, including administrators, learners (clients), and general users, while also performing reliably and securely under various conditions.

Functional Requirements

Functional requirements describe the core operations the system must support. These are grouped according to the type of user and common functionalities:

- a. **Admin Specific Functions** Administrators are responsible for overseeing all operations of the platform. Their features are geared toward managing users, content, billing, and moderation. The system must provide comprehensive tools to ensure smooth coordination, scalability, and security.
 - **User and Role Management:** Create, update, and delete user accounts, assign roles (admin, client), and manage permissions.
 - **Course and Offer Management:** Add, edit, and archive courses, workshops, and master programs, including content, schedules, and seat availability.
 - **Billing Oversight:** Monitor transactions, validate offline payments, and manage pricing for learning resources.
 - **Moderation Tools:** Review user-submitted content, manage reported items, and detect misuse or policy violations.
- b. **Learner Specific Functions** Learners interact with the platform to access educational content and services. Their functions focus on registration, learning progression,

participation, and secure payments. These tools should offer a personalized, smooth, and empowering user experience.

- **Account Registration and Profile Customization:** Enroll in offers, manage personal data, and set learning preferences.
- **Learning Resource Access:** Browse and register for courses, workshops, and master programs.
- **Content Interaction:** Participate in forums, submit feedback, and suggest new workshops.
- **Secure Payments:** Initiate online or request offline payments for learning resources.
- **Progress Tracking:** View lecture history, completion status, and access certificates.

c. **Shared User Functions (Admin and Learner)** Certain features are shared between both administrators and clients. These ensure a consistent, accessible, and secure experience across roles, covering platform entry, notifications, and content browsing.

- **Authentication and Authorization:** Secure login and role based access control.
- **Multilingual Support:** Translation of dynamic content for users of diverse linguistic backgrounds.
- **Notification System:** Receive alerts on new lectures, enrollments, or administrative updates.
- **Search and Filtering:** Search and filter content using keywords, tags, or categories.
- **Content Viewing:** Structured access to educational material, embedded multimedia, and attached resources.

The functional requirements outlined above define the critical tasks users must be able to accomplish within the platform. To ensure the platform also performs well under diverse scenarios and provides a high-quality user experience, it is necessary to define its non functional requirements.

Non-Functional Requirements

Non functional requirements address the quality characteristics of the system, which influence its usability, reliability, and maintainability.

- **Performance:** The platform must support fast content loading and responsive interfaces, even during high demand.
- **Scalability:** Capable of accommodating increased numbers of users, courses, and interactions without degradation.
- **Security:** Guarantee data privacy, content protection, secure payments, and role based access control.
- **Maintainability:** Modular architecture that allows for easy updates, debugging, and extensibility.
- **Usability:** Simple, consistent, and accessible interfaces for all user types.
- **Availability:** Ensure high system uptime with fallback mechanisms and regular backups.
- **Interoperability:** Enable integration with third party services (e.g., payment gateways, external learning APIs).

This comprehensive set of requirements ensures that the system will not only function correctly but also meet modern expectations for performance, reliability, and user satisfaction.

2.4.2 Performance Considerations

Performance is central to ensuring that the platform remains responsive and accessible as the user base grows. These technical factors directly impact user satisfaction and operational success [34]:

- **Scalability:** The platform should handle increasing numbers of users and activities, ensuring smooth operation during peak usage.
- **Speed and Responsiveness:** Pages, searches, dashboards, and live features (e.g., chat or video) must load quickly and respond instantly.
- **Efficient Resource Usage:** Optimize backend operations to prevent bottlenecks or crashes.
- **Offline Accessibility:** Enable users to download learning content for offline use, maintaining engagement despite connectivity issues.

2.4.3 Security Considerations

Robust security measures are essential to build trust and comply with data protection standards. This section outlines core practices for maintaining user privacy and protecting institutional integrity [35]:

- **Data Privacy:** Encrypt sensitive user data and enforce strict access policies.
- **User Authentication:** Implement secure login methods (e.g., two factor authentication) to prevent unauthorized access.
- **Secure Payments:** Integrate certified payment processors to handle financial transactions safely.
- **Regular Audits and Updates:** Conduct ongoing security reviews and software patches.
- **Access Control:** Clearly define and enforce user roles to safeguard sensitive operations.

2.4.4 Usability Considerations

A platform's usability determines how effectively learners and administrators can interact with it. A user centered design ensures accessibility, ease of navigation, and adaptability to different preferences [36, 37, 38, 39]:

- **Intuitive Interface:** Offer a clean, consistent layout that supports simple navigation.
- **Accessible Design:** Adhere to WCAG and ISO 9241 standards for users with disabilities.
- **Multilingual Support:** Offer translated content and UI components to accommodate diverse users.
- **Help and Support:** Include a help center, FAQs, and live support for user assistance.
- **Personalization:** Allow users to configure preferences, such as notification settings or interface themes [32].

2.5 Requirement Analysis

In this section, we conduct a thorough analysis of the requirements for our Biomechanic Club e-learning platform, focusing on the specific needs of dental professionals and orthodontic students. Our goal is to create an online education platform that enhances learning capabilities and resource accessibility in orthodontics and TMJ disorders. This platform will provide students and instructors with access to a structured collection of courses, workshops, and master programs with integrated AI powered learning tools.

The platform aims to offer the ability to access interactive course materials and test knowledge through practical case simulations. It should be accessible from any device via a web interface for all visitors, with additional functionalities reserved for enrolled members. These users will be able to upload and manage clinical cases, learning materials, and treatment plans, which will be hosted on a secure server. By carefully considering the requirements and design principles, we aim to create a robust and scalable solution that addresses the unique demands of dental education professionals and students in the Arab region.

2.5.1 Potential Users and Their Roles

This section identifies the different types of users and system actors expected to interact with the platform. Each role is defined based on its purpose and privileges within the e-learning environment.

2.5.1.1 Users

The users are the key participants who contribute to and benefit from the platform's functionalities.

- **Students:** The primary learners enrolled in the platform's courses and workshops. They access educational materials, take quizzes, and interact with instructors and peers.
- **Instructors:** Dental professionals responsible for designing, uploading, and managing course content. They also provide feedback and guide learners through interactive tools.

- **Administrators:** Individuals who manage platform operations, user access, content oversight, and overall system integrity.
- **Content Moderators:** Specialists who review and ensure the quality and accuracy of uploaded educational materials, adhering to academic standards.

2.5.1.2 Actors

Actors represent system roles and define what actions different user types can perform based on their access level.

- **Guest:** A visitor who is not logged in. Guests can browse public content, create an account, log in, or recover their password.
- **User:** Once logged in, a user inherits all guest privileges and can also comment on content, enroll in courses, and manage their profile.
- **Admin:** Has the highest level of access, inheriting all user permissions, with additional abilities to manage users, moderate content, and access the administrative dashboard for monitoring platform performance.

The Use Case Diagram (UCD) for the **Platform** models the functionalities accessible by different types of users within an educational platform. It defines how actors interact with the system, outlining possible actions and system responses. 2.1

2.5.2 Principle Use Cases Descriptions

In this section, we dive into detailed descriptions of key use cases that outline essential functionalities of the platform:

2.5.2.1 Manage Courses

The course management functionality [2.2] is central to organizing the platform's educational content. It allows administrators to create and manage structured courses, including modules,

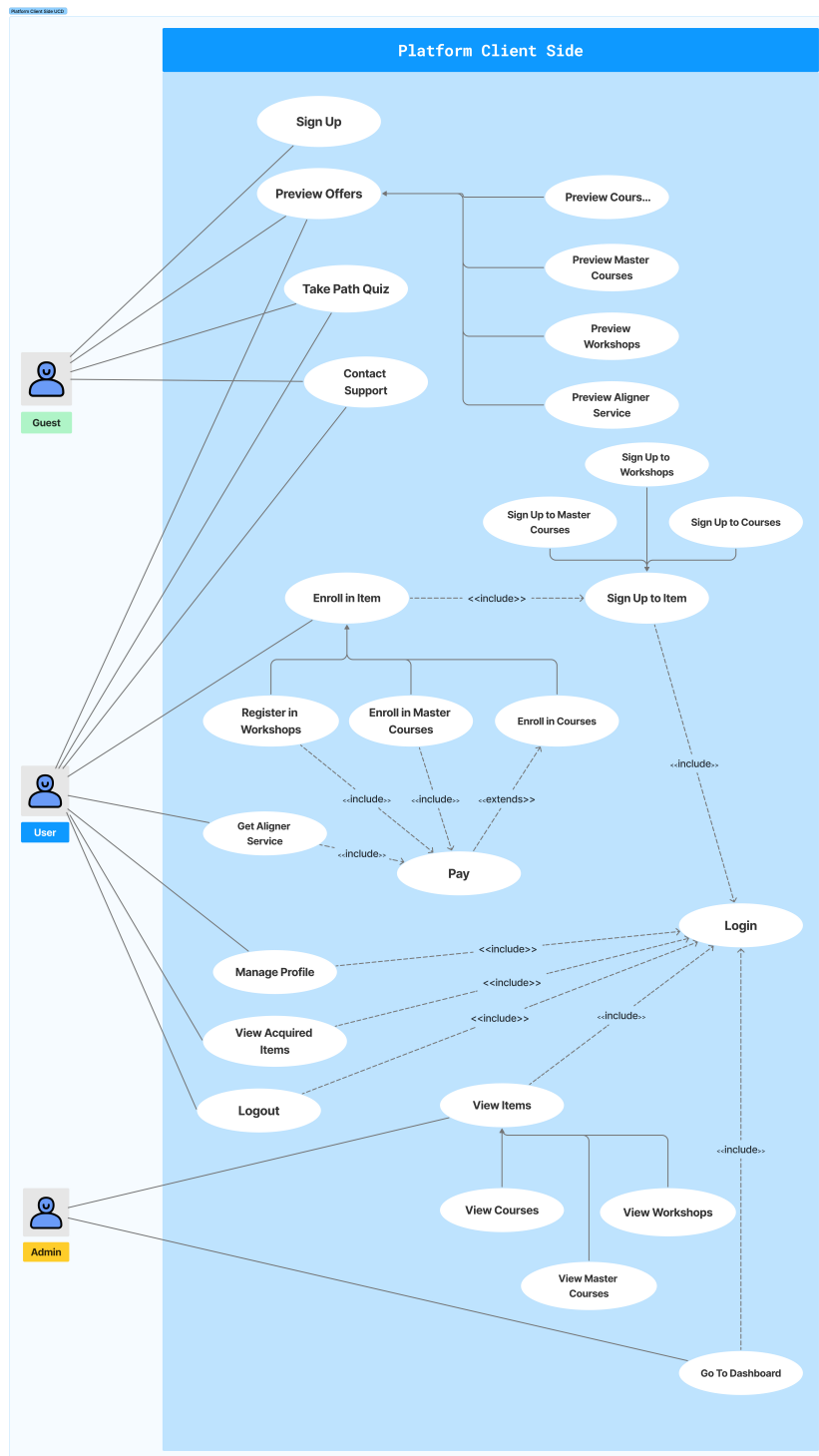


Figure 2.1: Platform Client Side Use Case Diagram

lectures, and exams. Additional features such as draft saving, publishing, and optional payment integration support flexible and efficient course delivery tailored to dental surgery professionals.

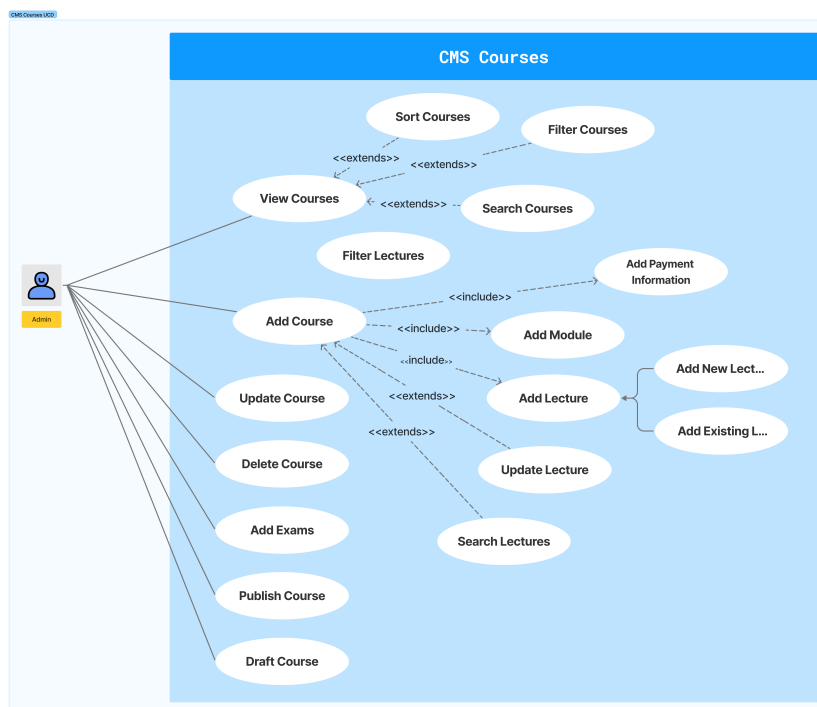


Figure 2.2: Courses CMS Use Case Diagram

Functional Requirements

- Admins can add, update, and delete courses.
- Admins can manage modules, lectures (new and existing), and exams within each course.
- Courses can be saved as drafts or published.
- Admins can search, sort, and filter existing courses and lectures.
- Admins can add payment information to courses if applicable.

Involved Actors

- **Admin:** The platform administrator responsible for managing course content, structure, and visibility.

Main Scenario

The following steps describe the general workflow for managing courses in the CMS:

- a. The admin Navigates into the course creation page in the CMS section.
- b. The admin adds a new course, which includes:
 - Adding one or more modules.
 - Adding new or existing lectures to each module.
 - Optionally adding exams or payment information.
- c. The admin can update course information or associated content at any time.
- d. The admin may delete a course if it is obsolete or no longer needed.
- e. The course may be saved as a draft or published to be visible to users.
- f. Admins can search or filter lectures separately, and update existing lecture content.

The sequence diagram 2.3 sums the entire process of creating any educational content, including the courses in the CMS section.

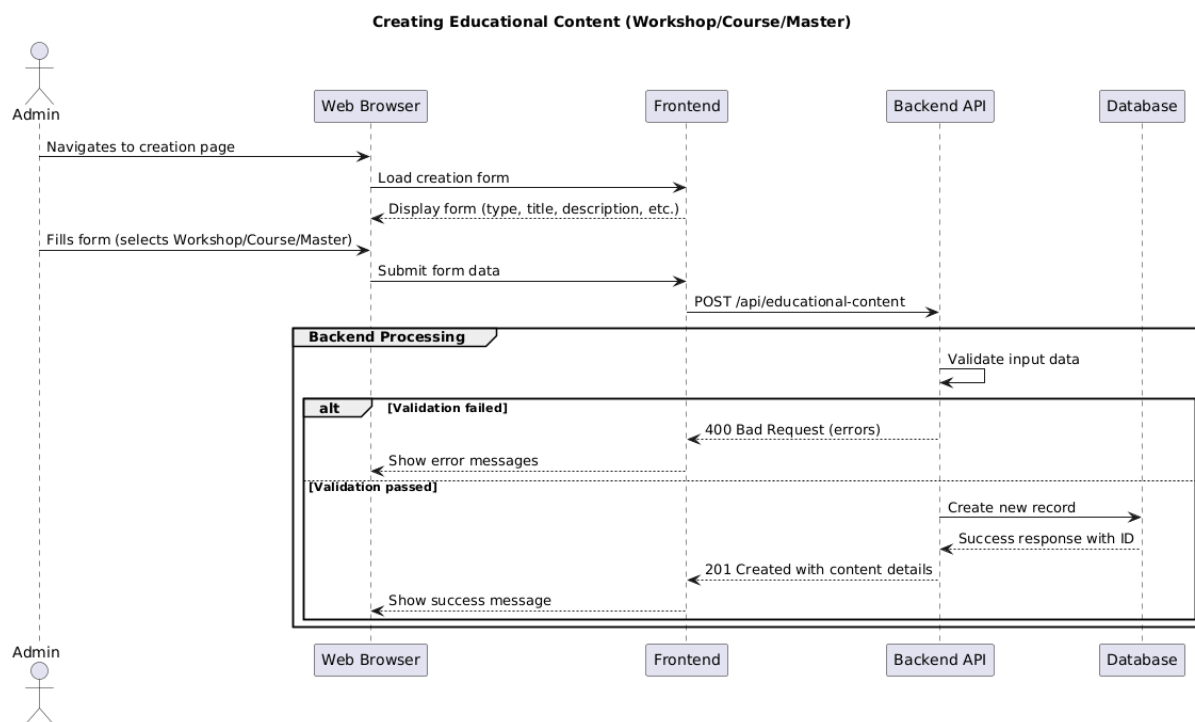


Figure 2.3: Create Education Content (Course, Master, Workshop)

2.5.2.2 Course Enrollment Process

This section describes the processes for enrolling in both paid and free courses. Each process involves multiple components, including the user, web browser, frontend, backend API, database, and, in the case of paid courses, the Stripe API for payment processing.

Paid Course Enrollment Process

The Paid Course Enrollment Process is designed to securely handle the enrollment of users in paid courses [2.4]. The steps are as follows:

- a. **Enrollment Initiation:** The user clicks the "Enroll" button for a paid course on the web browser. The frontend initiates the enrollment process by sending a request to the backend API.
- b. **Authentication and Eligibility Check:**
 - The backend verifies the user's authentication status by querying the database.
 - It also checks the user's eligibility for the course (e.g., prerequisites, no duplicate enrollments).

If the user is not authenticated or not eligible, an error response is sent back to the frontend, which displays an error message.

- c. **Payment Setup:** If the user is eligible, the backend interacts with the Stripe API to create a `PaymentIntent`, specifying the amount, currency, and metadata. The backend creates a pending enrollment record in the database and sends the payment setup details (e.g., `client_secret` and payment URL) to the frontend. The frontend redirects the user to the Stripe checkout page.
- d. **Payment Processing:** The user completes the payment on the Stripe checkout page. Stripe processes the payment and triggers one of the following outcomes:
 - **Payment Successful:** Stripe sends a webhook event (`payment_succeeded`) to the backend. The backend verifies the payment, updates the enrollment record in the database, and confirms the enrollment. The frontend polls the backend for the enrollment status and grants the user access to the course.

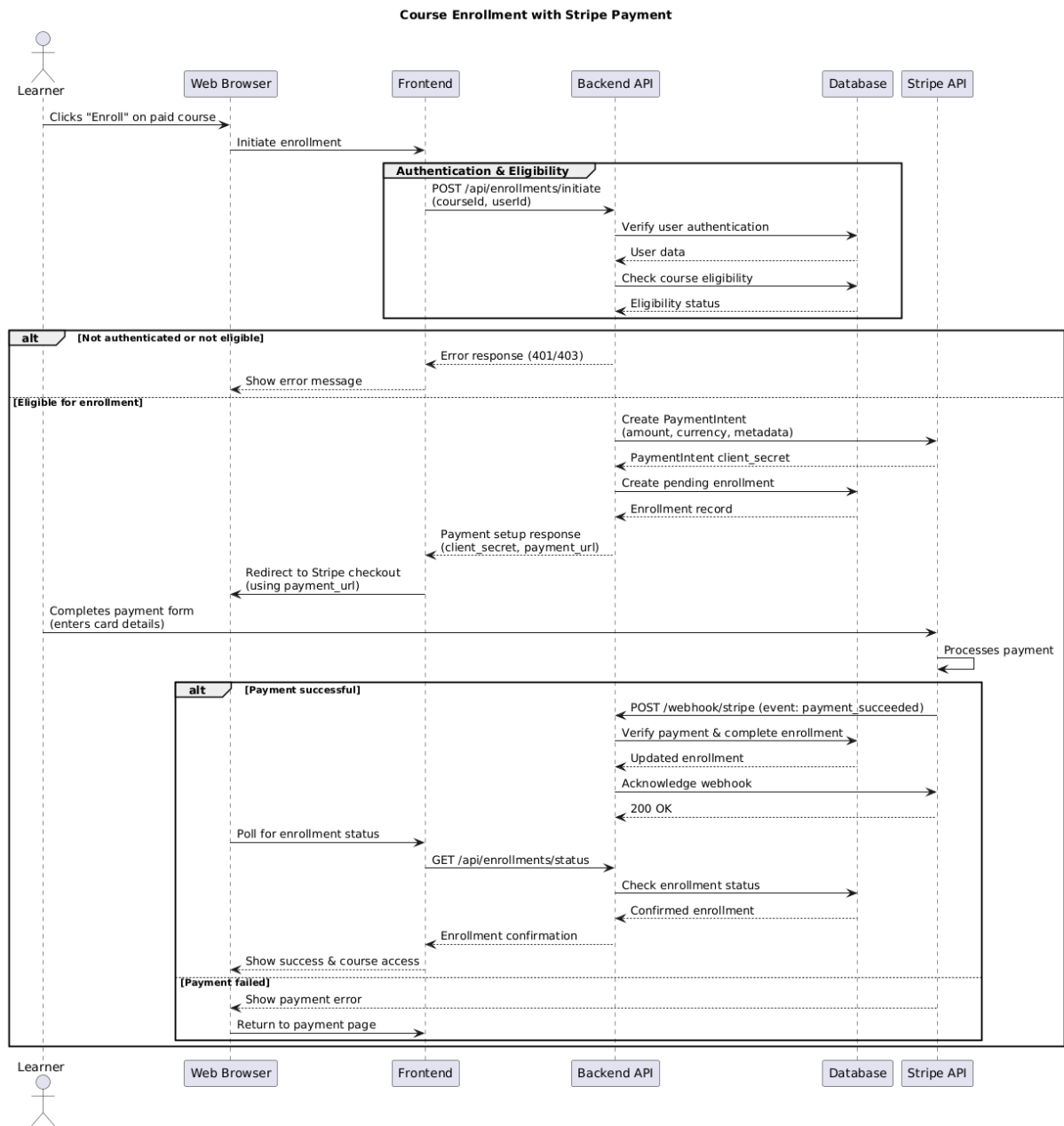


Figure 2.4: Paid Course Enrollment Process Sequence Diagram

- **Payment Failed:** If the payment fails, Stripe displays an error message, and the user is redirected back to retry the payment.
- e. **Enrollment Confirmation:** Upon successful payment, the backend confirms the enrollment, and the frontend updates the user interface to show a success message and grant access to the course materials.

Free Course Enrollment Process

The Free Course Enrollment Process is simpler and does not involve payment processing [2.5].

The steps are as follows:

- a. **Enrollment Initiation:** The user clicks the "Enroll" button for a free course on the web browser. The frontend sends an enrollment request to the backend API.
- b. **Authentication Check:**
- The frontend checks the user's authentication status by sending a request to the backend API.
 - If the user is not authenticated, the frontend redirects the user to the login page. Once the user logs in, the process resumes.
 - If the user is authenticated, the backend proceeds with the enrollment process.
- c. **Enrollment Process:** The frontend sends a POST request to the backend API to enroll the user in the course. The backend performs the following validations:
- Checks that the course exists and is free.
 - Verifies the user's eligibility (e.g., prerequisites, no duplicate enrollments).
- d. **Validation Outcomes:**
- If validation fails, the backend sends an error response to the frontend, which displays an error message to the user.
 - If validation passes, the backend creates an enrollment record in the database and sends a confirmation response to the frontend.
- e. **Enrollment Confirmation:** The frontend updates the user interface to show a success message and grants the user access to the course materials.

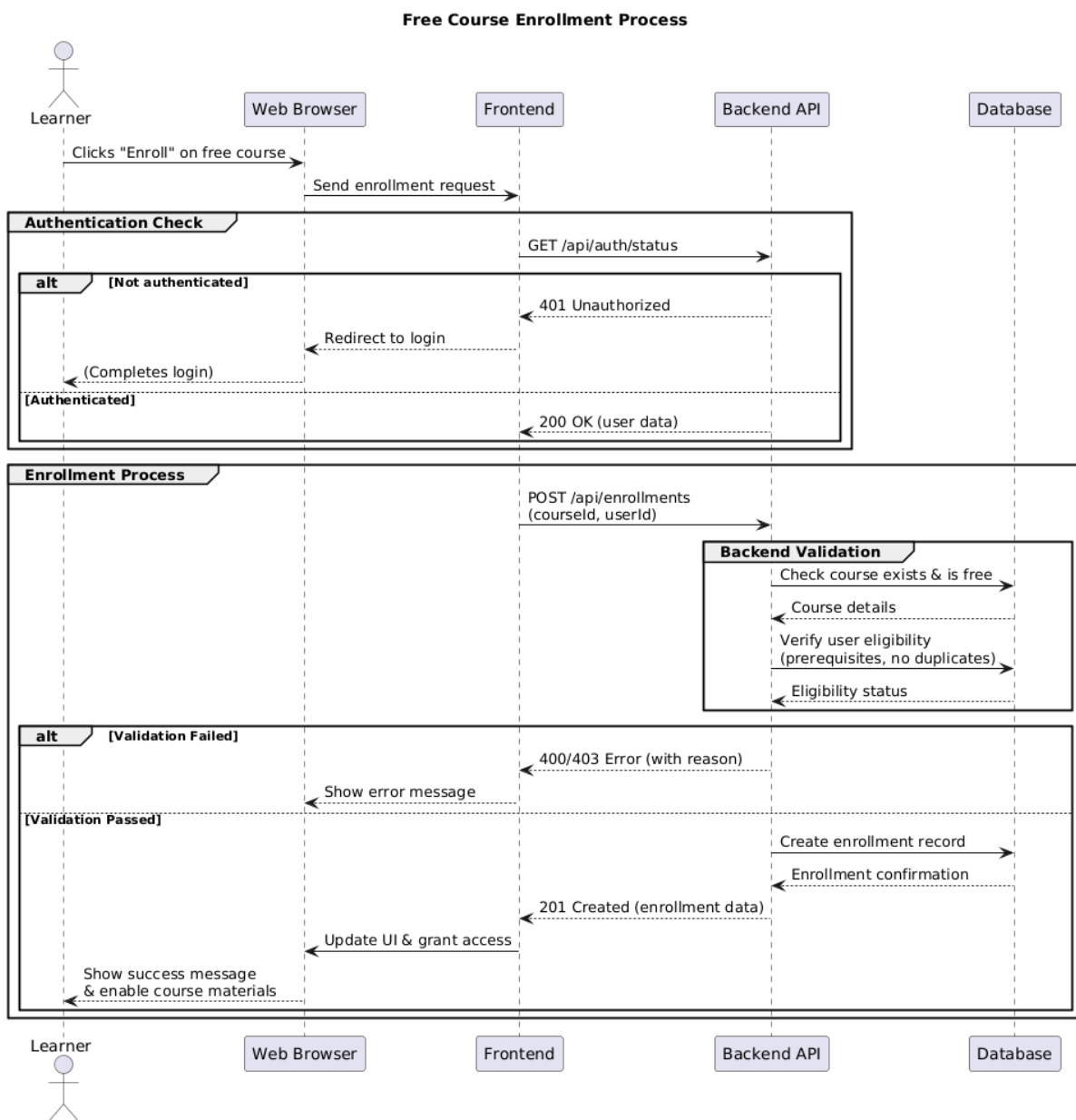


Figure 2.5: Free Course Enrollment Process Sequence Diagram

The primary difference between the two processes lies in the payment step. While the Paid Course Enrollment Process involves payment processing through Stripe, the Free Course Enrollment Process skips this step and directly validates the user's eligibility and creates the enrollment record. Both processes ensure secure and seamless enrollment experiences for users.

2.5.2.3 Registering in a Workshop

The Workshop Registration Process is designed to handle user registration for workshops with a seat-based payment flow. This process ensures efficient seat management, secure payment processing, and a fallback mechanism for waitlisting users when no seats are available [2.6]. The steps are as follows:

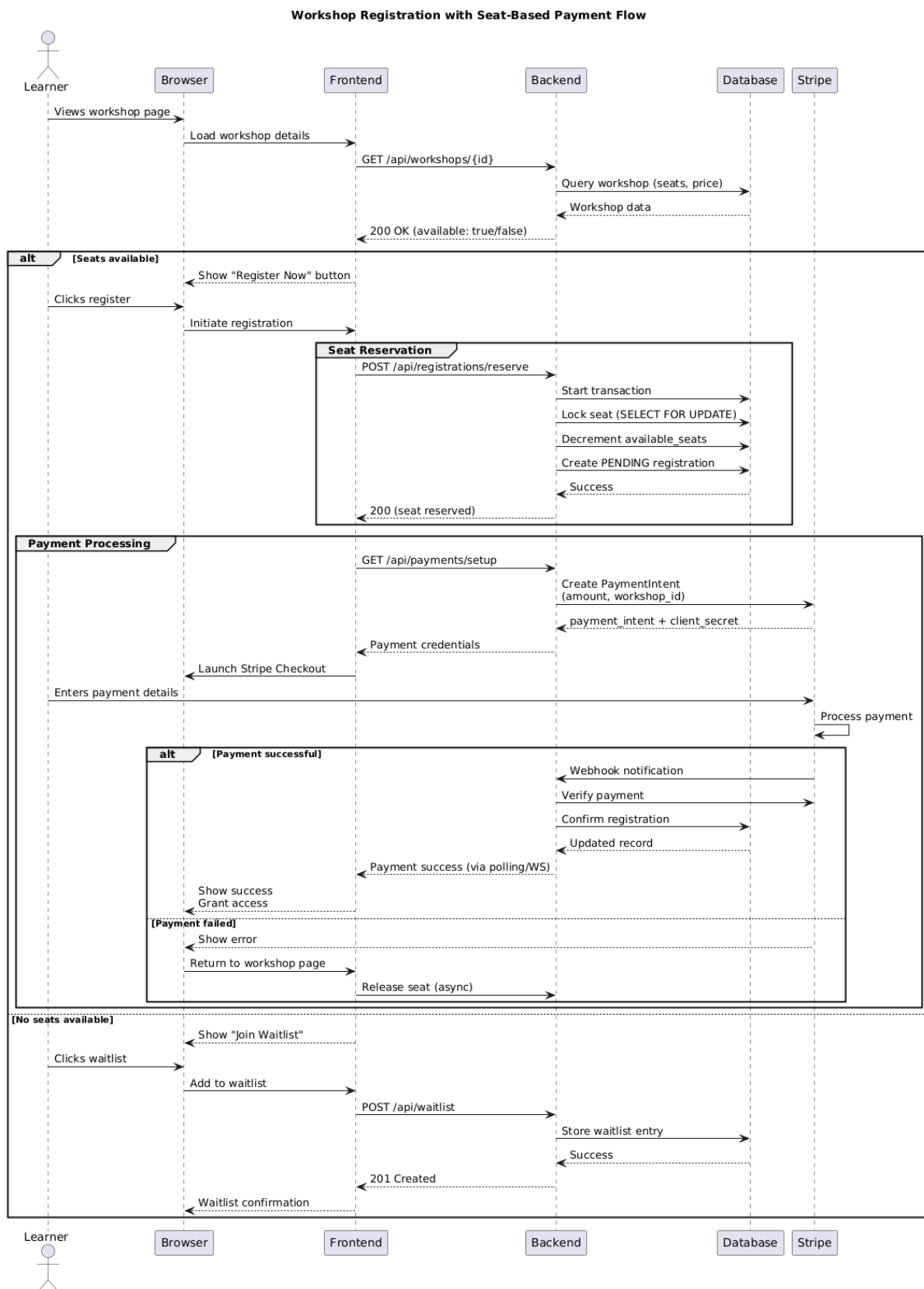


Figure 2.6: Workshop Registration Process Sequence Diagram

a. Viewing Workshop Details

- The user navigates to the workshop page using their browser.
- The browser sends a request to the frontend to load the workshop details.
- The frontend queries the backend API (GET /api/workshops/{id}) to retrieve workshop information, including the number of available seats and the price.
- The backend queries the database for the workshop data and returns the information to the frontend.
- The frontend displays the workshop details to the user, including whether seats are available.

b. Seat Availability Check

- **Seats Available:** If seats are available, the frontend displays a "Register Now" button to the user.
- **No Seats Available:** If no seats are available, the frontend displays a "Join Waitlist" option (explained in Step 5).

c. Seat Reservation

- When the user clicks "Register Now," the browser initiates the registration process.
- The frontend sends a POST /api/registrations/reserve request to the backend to reserve a seat.
- The backend starts a database transaction and performs the following actions:
 - Locks a seat using a SELECT FOR UPDATE query to prevent race conditions.
 - Decrements the number of available seats.
 - Creates a PENDING registration record in the database.
- Upon successful reservation, the backend returns a success response to the frontend.

d. Payment Processing

- The frontend sends a GET /api/payments/setup request to the backend to initiate the payment process.
- The backend interacts with the Stripe API to create a PaymentIntent, specifying the amount and workshop ID.
- Stripe returns the payment_intent and client_secret to the backend, which forwards them to the frontend.

- The frontend launches the Stripe Checkout page, where the user enters their payment details.
- Stripe processes the payment and triggers one of the following outcomes:
 - **Payment Successful:**
 - * Stripe sends a webhook notification (payment_succeeded) to the backend.
 - * The backend verifies the payment with Stripe and updates the registration record in the database to CONFIRMED.
 - * The backend notifies the frontend of the payment success (via polling or WebSocket).
 - * The frontend updates the user interface to show a success message and grants the user access to the workshop.
 - **Payment Failed:**
 - * Stripe displays an error message to the user.
 - * The user is redirected back to the workshop page.
 - * The frontend sends an asynchronous request to the backend to release the reserved seat.

Waitlist Management (If No Seats Available)

- If no seats are available, the frontend displays a "Join Waitlist" option.
- When the user clicks "Join Waitlist," the browser sends a request to the frontend to add the user to the waitlist.
- The frontend sends a POST /api/waitlist request to the backend.
- The backend stores the user's waitlist entry in the database and returns a confirmation response.
- The frontend displays a waitlist confirmation message to the user.

Key Features of the Process

- **Seat Reservation:** The use of database transactions and seat locking ensures that seat availability is managed without conflicts.
- **Secure Payment:** Stripe is used for secure payment processing, with webhook notifications to confirm payment success.

- **Waitlist Management:** Users can join a waitlist if no seats are available, ensuring they are notified when seats open up.

This process ensures a seamless and secure registration experience for users while efficiently managing seat availability and payments.

2.5.2.4 Overview of Offers

Offers are platform-defined resources that grant users access to specific learning opportunities. These can include:

- **Credits:** Virtual tokens used to enroll in paid content.
- **Masterclasses:** Specialized training programs combining theory and practice.
- **Courses:** Structured sequences of educational lectures and exercises.
- **Workshops:** Short-term, practical sessions focused on specific skills or knowledge areas.

Offers can be granted to users manually by an administrator. They represent a key mechanism for granting access, rewarding participation, or promoting targeted learning initiatives. The following section explains the process administrators follow to assign offers to users.

2.5.2.5 Offer Assignment Process

The Offer Assignment Process is designed to allow administrators to assign offers (e.g., credits, masterclasses, courses, or workshops) to specific users or groups of users. This process ensures proper validation, offer creation, and user notification [2.7]. The steps are as follows:

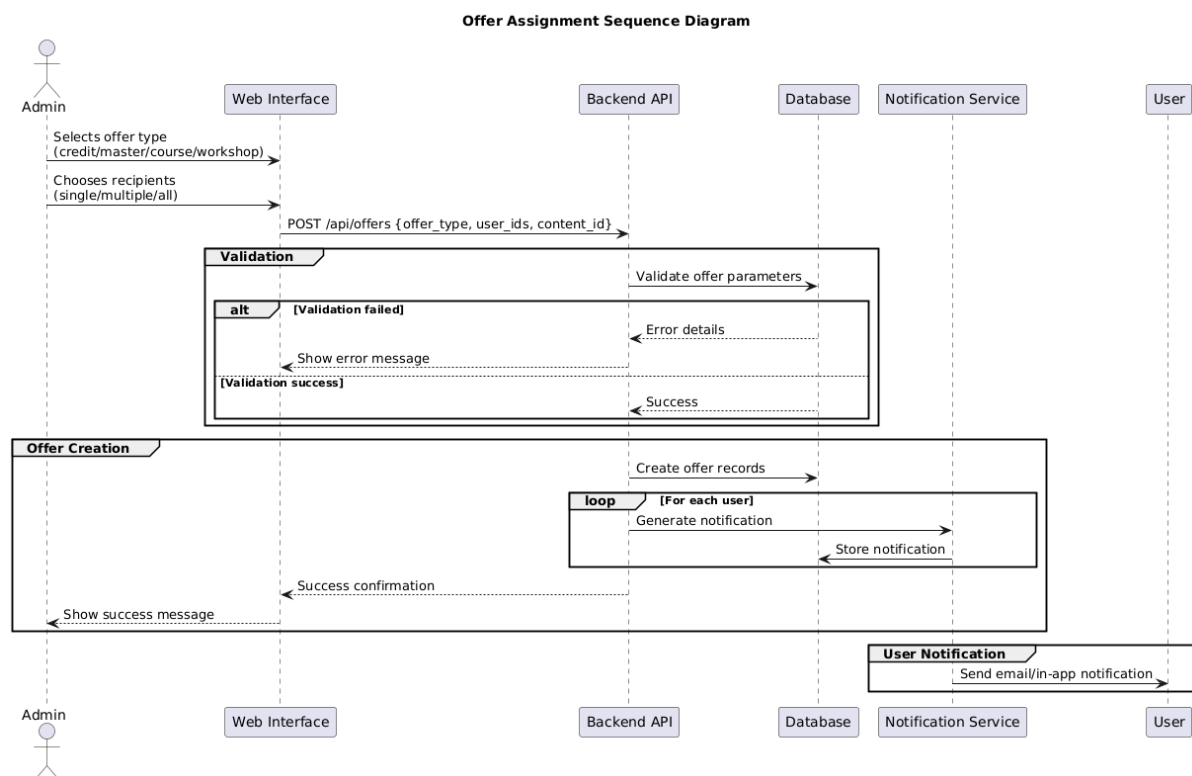


Figure 2.7: Offer Assignment Sequence Diagram

Offer Selection and Recipient Assignment

- a. The administrator selects the type of offer (e.g., credit, masterclass, course, or workshop) through the web interface.
- b. The administrator chooses the recipients for the offer, which can be a single user, multiple users, or all users.
- c. The web interface sends a POST `/api/offers` request to the backend API, including the offer type, user IDs, and content ID.

Validation

- a. The backend API validates the offer.
- b. The validation process checks for issues such as invalid user IDs, incorrect content IDs, or unsupported offer types.
- c. The outcomes of the validation are as follows:

- **Validation Failed:** If the parameters are invalid, the backend responds with an error message to the Frontend, which displays them to the administrator.
- **Validation Success:** If the parameters are valid, the Backend confirms success, and the process proceeds to offer creation.

Offer Creation

- a. The backend API creates offer records in the database for each recipient.
- b. For each user, the API generates a notification by interacting with the notification service.
- c. The notification service stores the notification details in the database.
- d. Once all offer records and notifications are created, the API sends a success confirmation to the web interface.
- e. The web interface displays a success message to the administrator.

User Notification

- a. The notification service sends an email or in-app notification to each recipient, informing them about the assigned offer.

Key Features of the Process

- **Validation:** Ensures that only valid offers are created by checking the parameters against the validation rules.
- **Scalability:** The process supports assigning offers to multiple users or all users through a loop mechanism.
- **User Notification:** Ensures that recipients are informed about the assigned offers through email or in-app notifications.
- **Error Handling:** Provides clear feedback to the administrator in case of validation failures.

This process ensures a seamless and efficient workflow for assigning offers to users while maintaining data integrity and providing timely notifications.

2.5.3 Design Considerations

Having outlined the core functionalities and use cases of our dental surgery e-learning platform, we now turn our attention to the design considerations that ensure its success and long-term viability. While functional requirements describe what the platform should do, it is equally critical to examine the non-functional aspects that influence how well it performs, scales, and protects user data.

These design considerations go beyond basic implementation details. They define the platform's ability to deliver a reliable, secure, and responsive learning experience to its users, which includes dental professionals, educators, and administrative staff. In this context, non-functional requirements are not optional enhancements but essential pillars that directly impact user trust, engagement, and satisfaction.

Among the most important considerations are security, scalability, and performance. Each of these plays a distinct and essential role in shaping the robustness and efficiency of the platform. Security ensures the protection of user data and educational content; scalability supports the growing number of learners and resources; and performance guarantees a smooth, uninterrupted user experience.

The following sections provide a detailed explanation of how these key aspects are addressed in the platform's design and implementation strategy.

2.5.3.1 Security

Security is essential due to the sensitivity of user data and the value of the learning materials. Our design incorporates:

- **Authentication and Authorization:** Access is restricted via secure login and role-based permissions to ensure only authorized users reach protected content.
- **Email Verification:** New users must verify their email address before gaining full access

to platform features.

- **Data Protection:** All communication is encrypted using HTTPS. Input validation, CSRF protection, and secure handling of user data are strictly enforced.
- **Content Protection:** To discourage unauthorized distribution, we apply measures such as watermarking documents and restricting file downloads.

2.5.3.2 Scalability

To accommodate growth in both users and educational resources, the platform is built with scalability in mind:

- **Efficient Data Management:** The database is structured to support efficient queries, with indexes and caching layers applied where needed.
- **Load Handling:** Background tasks such as email sending and report generation are offloaded to queues to reduce server load.

2.5.3.3 Performance

High performance is critical for ensuring a smooth user experience during lectures, exams, and interactive features:

- **Responsive Interface:** The user interface is optimized to respond quickly, minimizing load times and improving navigation fluidity.
- **API Efficiency:** Data access patterns are optimized to reduce unnecessary database calls and minimize latency.
- **Monitoring and Optimization:** Regular performance audits help identify and resolve bottlenecks to maintain responsiveness under load.

2.6 Platform Conception

Following the definition of functional and non-functional requirements, this section presents the conceptual design of the platform. The architectural design adopts a decoupled Model-View-Controller (MVC) architecture, selected for its modular structure and compatibility with modern web development practices. The design emphasizes scalability, maintainability, and flexibility key considerations for both initial deployment and long-term evolution.

This section is structured as follows: first, the overall system architecture is described; then, the database schema is introduced; finally, the sitemap is presented to illustrate the user interaction flow and access control mechanisms.

2.6.1 Decoupled MVC Architecture

The platform is built upon a *decoupled* Model-View-Controller (MVC) architecture, which separates the frontend and backend into independent components. This contrasts with the traditional monolithic MVC model, in which the backend directly handles both application logic and view rendering.

In the decoupled configuration, the **backend** comprises the Model and Controller layers. It is responsible for managing application logic, interacting with the database, enforcing business rules, and exposing functionality through RESTful APIs. The backend is developed using a server-side framework that ensures secure and efficient processing of client requests.

The **frontend**, on the other hand, is implemented as a standalone application using a modern JavaScript framework (e.g., React). It is responsible for the View layer, which includes rendering the user interface, managing client-side routing, and consuming the API endpoints provided by the backend. This approach results in a clear separation of responsibilities, where the backend focuses on data and logic, while the frontend focuses on user interaction and presentation.

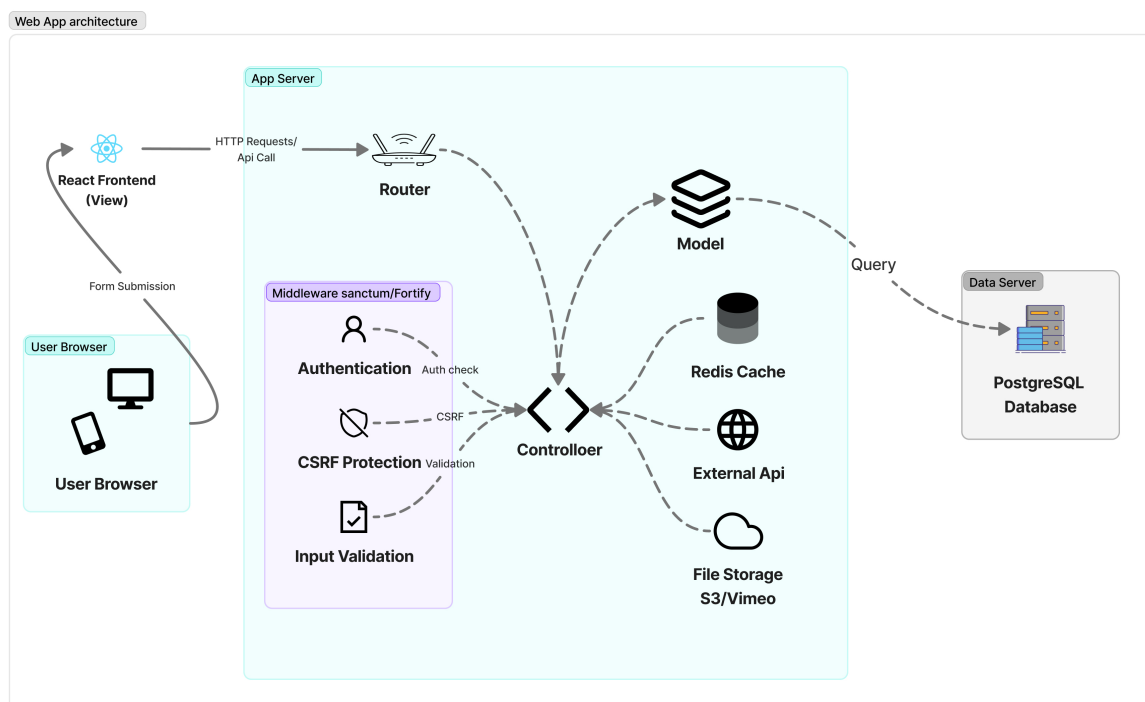


Figure 2.8: Overview of the Decoupled MVC Architecture

This architecture offers several advantages:

- **Parallel Development:** Backend and frontend teams can work independently, improving development efficiency.
- **Scalability:** Components can be scaled separately, depending on demand.
- **Maintainability:** Clear separation makes the codebase easier to manage and evolve.
- **Extensibility:** Facilitates future integration with mobile clients or third-party platforms.

By adopting the decoupled MVC model, the platform gains a robust foundation that supports modular growth, enhances responsiveness, and aligns with contemporary software engineering best practices.

2.6.2 Database Design

The database design of the Biomech platform is based on a robust **relational model**, ensuring data integrity, consistency, and scalability. The schema is carefully structured to support the

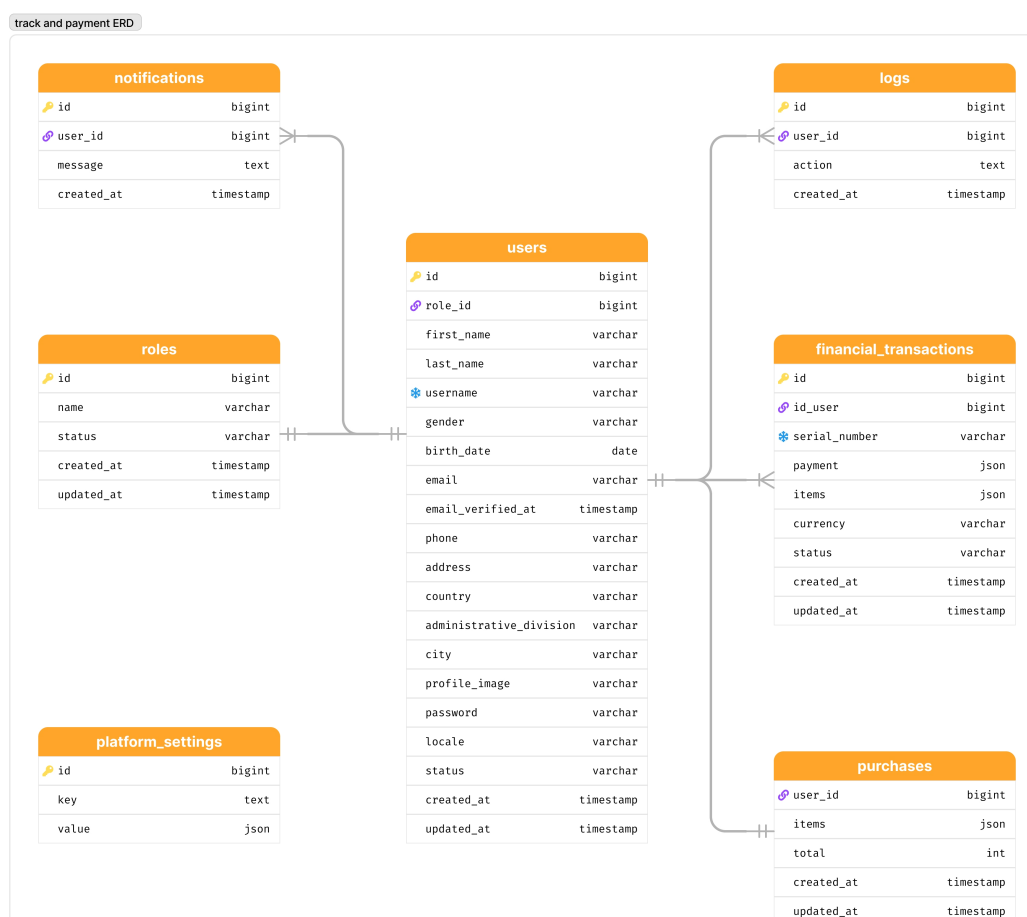


Figure 2.9: Track and Payment Entity-Relationship Diagram

platform's diverse educational offerings such as courses, workshops, and master programs while enabling user management, progress tracking, and financial operations.

Entity-Relationship Overview To provide a clear understanding of the system's data structure, the following ER diagrams illustrate the main modules of the platform:

- **User and Role Management:** The users 2.9 table is the central entity, storing personal, authentication, and profile information. Each user is linked to a role (e.g., student, admin) that determines their permissions. Notifications and logs are associated with users to track activity and communication.
- **Offers and Purchases:** The offers 2.13 table provides a unified structure for all learning products (courses, workshops, masters), supporting flexible assignment and tracking. The users_offers and purchases tables record which users have access to which offers and their purchase history.

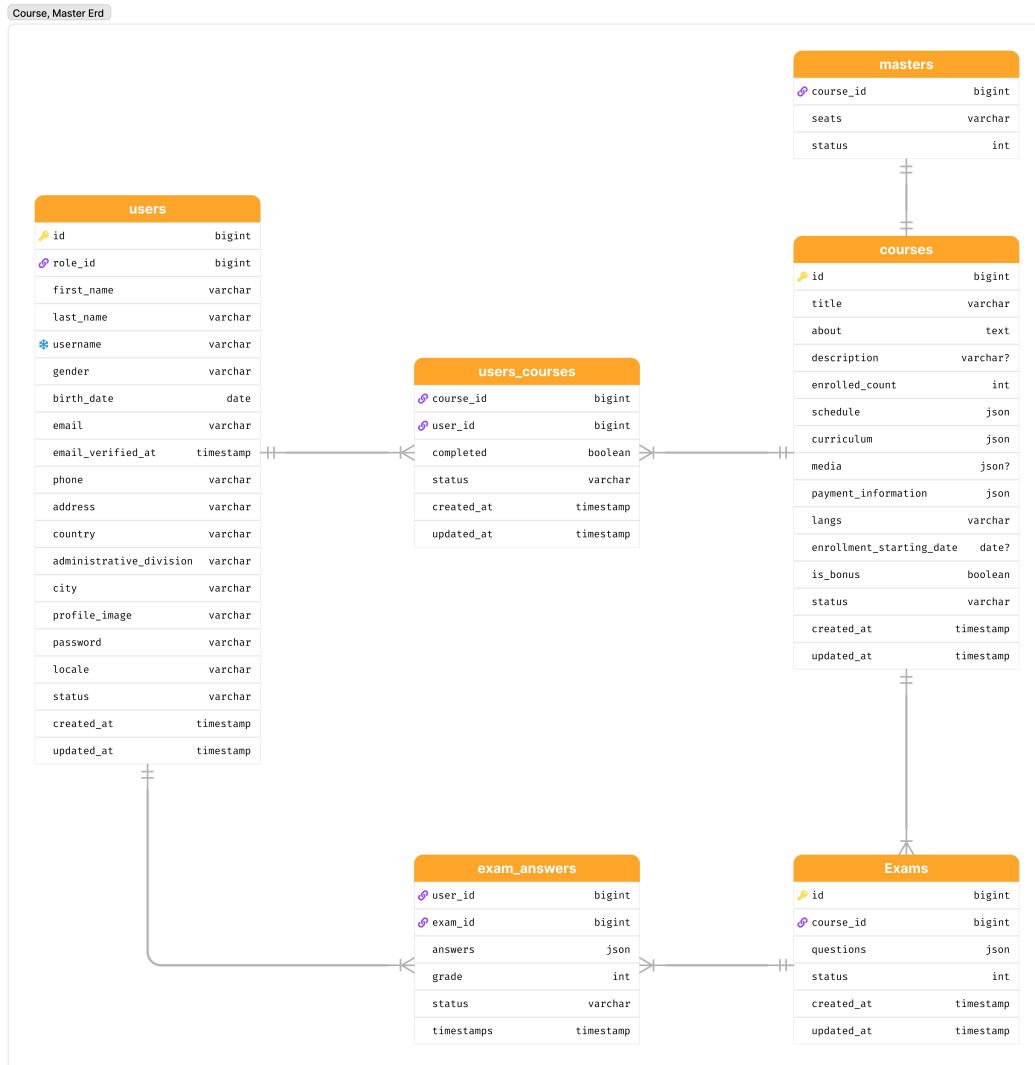


Figure 2.10: Courses and Masters Entity-Relationship Diagram

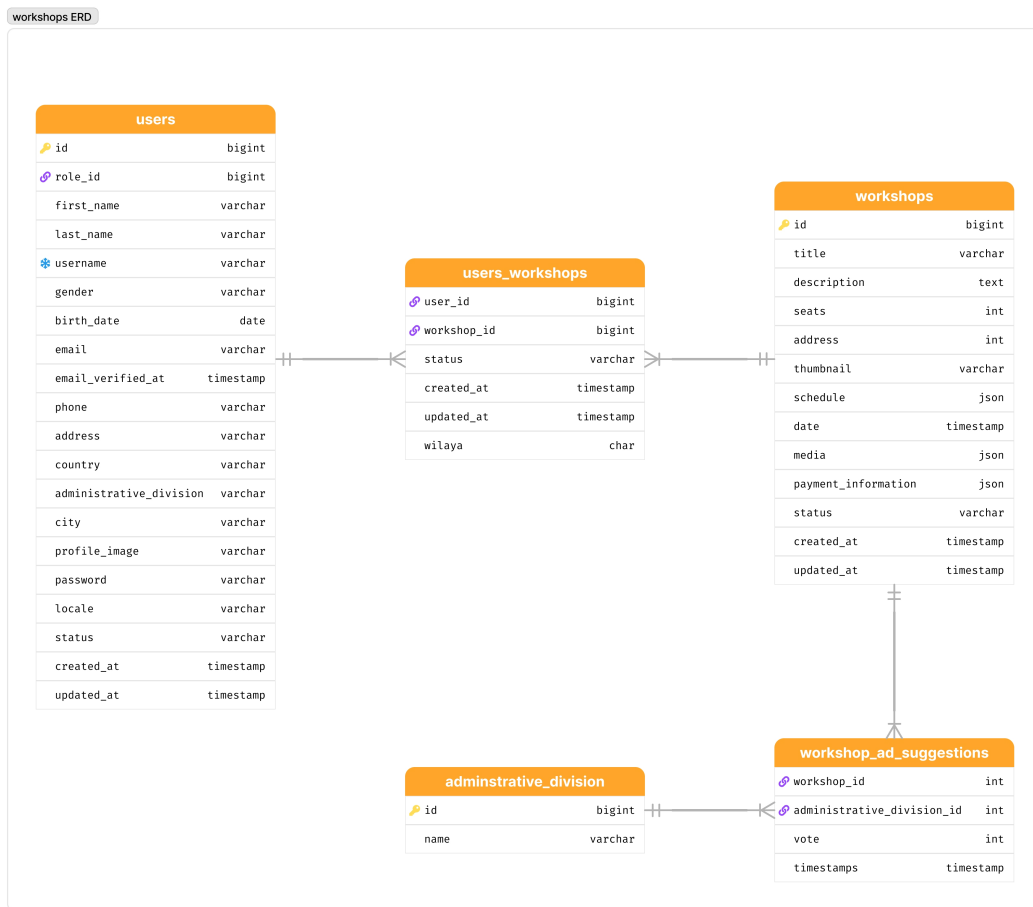


Figure 2.11: Workshops Entity-Relationship Diagram

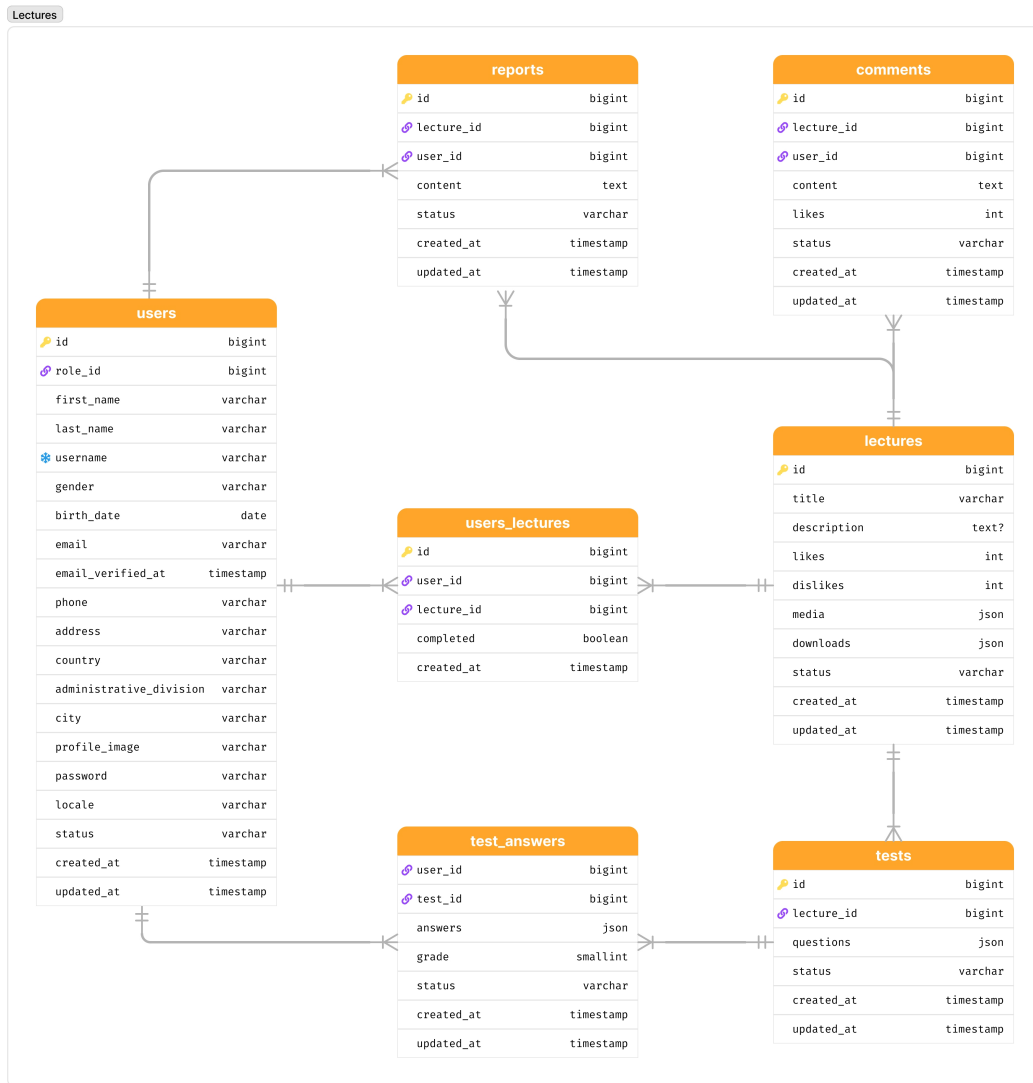


Figure 2.12: Lectures Entity-Relationship Diagram

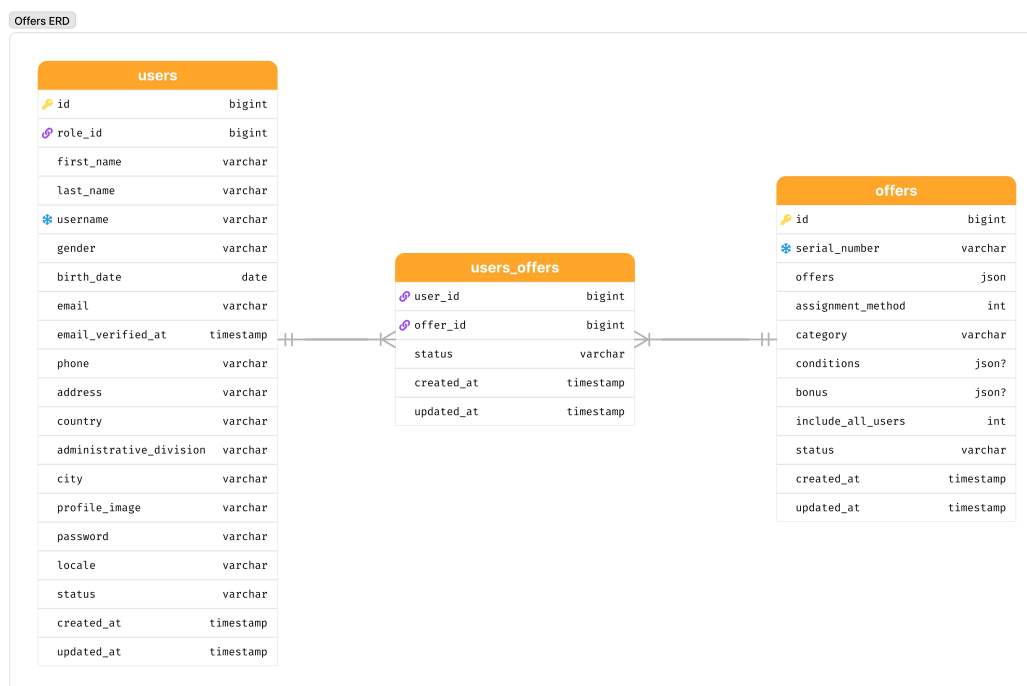


Figure 2.13: Offers Entity-Relationship Diagram

- **Courses and Masters:** The courses and masters 2.10 tables represent structured learning programs. The users_courses table tracks enrollment and completion status, while the exams and exam_answers tables manage assessments and results.
- **Workshops:** Workshops are managed via the workshops table 2.11, with user participation tracked in users_workshops. The workshop_ad_suggestions and administrative_division tables enable users to suggest and vote on new workshop locations, supporting localization and community engagement.
- **Lectures and Learning Progress:** The lectures 2.12 table stores individual learning units, with user progress tracked in users_lectures. The comments and reports tables allow users to interact with content, provide feedback, and flag issues.
- **Financial Transactions:** The financial_transactions 2.9 table logs all monetary operations, including payments, refunds, and discounts, ensuring traceability and accountability.
- **Platform Settings:** The platform_settings 2.9 table stores configuration data, enabling dynamic adjustment of platform-wide parameters.

Design Rationale This modular schema supports:

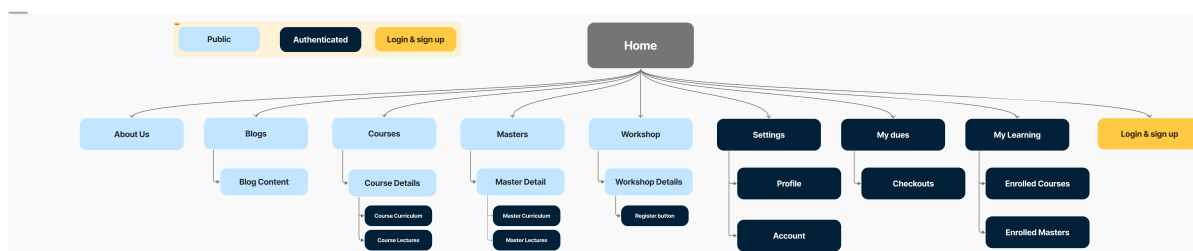


Figure 2.14: Platform Sitemap (Client)

- **Scalability:** Each module can be extended independently as new features are added.
- **Data Integrity:** Foreign key relationships enforce consistency between users, offers, and transactions.
- **Flexibility:** The polymorphic offers structure allows for easy addition of new educational products.
- **Localization:** Administrative divisions and workshop suggestions enable region-specific offerings.

By visualizing and describing each part of the database, the design ensures that all platform functionalities from registration and learning progress to payments and feedback are efficiently supported and easily maintainable.

2.6.3 Sitemap

the sitemap illustrated in 2.14 and 2.15 presents the structural layout of our platform. It outlines the main sections and their hierarchical relationships, providing a clear visual representation of how the system's pages are organized. This aids in understanding the navigation flow and user experience across the platform.

2.7 Conclusion

The design and architecture of the dental surgery e-learning platform demonstrate a forward-thinking approach to delivering professional education in a digital format. By combining structured content delivery, interactive learning features, robust administrative tools, and secure financial operations, the platform is well positioned to support a new generation of dental professionals.

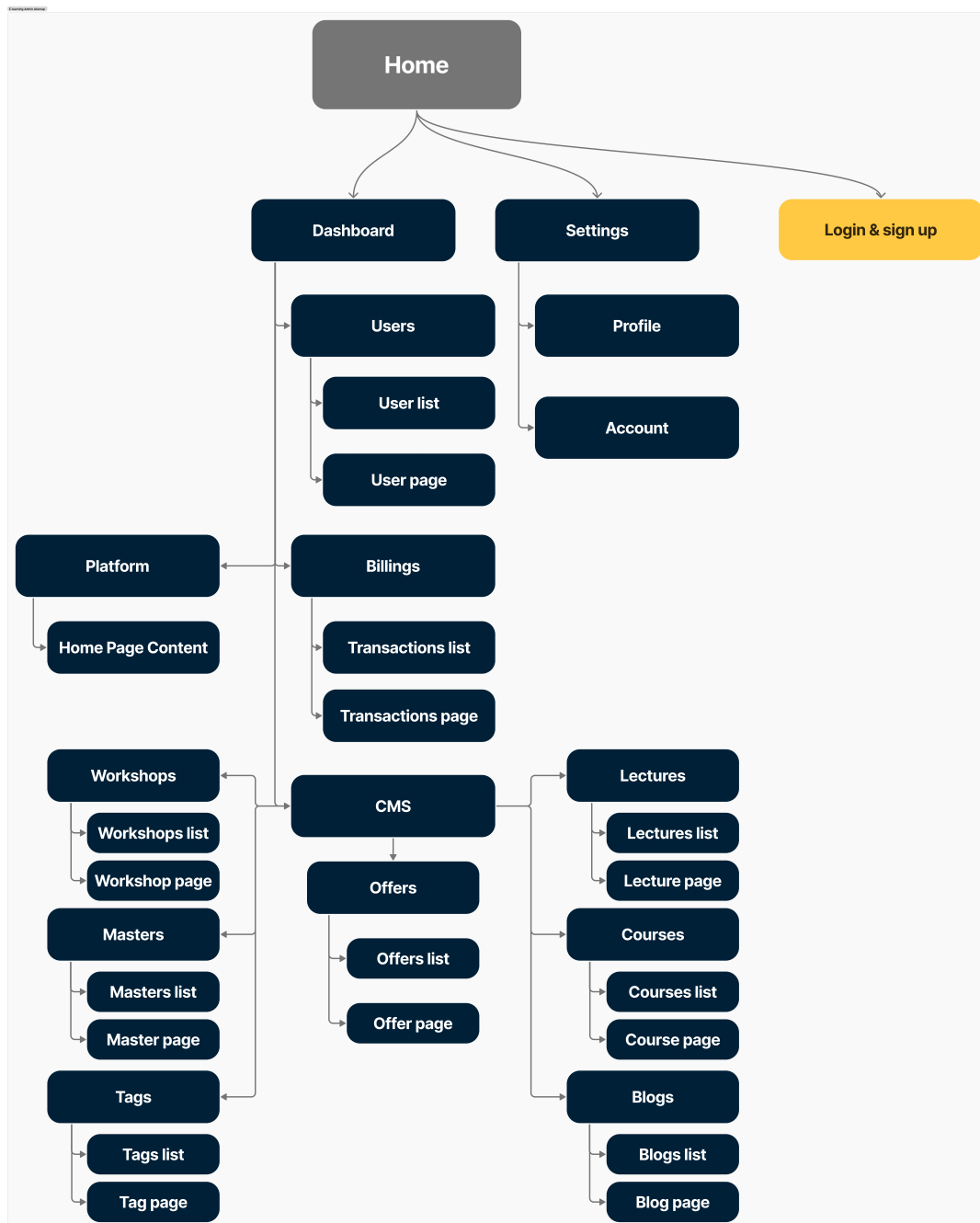


Figure 2.15: Platform Sitemap (Admin)

The use of a decoupled MVC architecture ensures flexibility and long-term maintainability, while the thoughtful database schema and integrated user workflows enable seamless management of educational resources, users, and financial activities. Key technical implementations, such as transactional seat reservations, Stripe based payments, and scalable offer assignment, exemplify the system's reliability and responsiveness under real world use.

Ultimately, this platform stands as a scalable foundation for expanding digital dental education, offering the infrastructure necessary to support future growth, enhanced interactivity, and broader geographic reach.

Chapter 3 Implementation of Biomec Platform For Dental Surgery

3.1 Introduction

This chapter details the practical implementation of the Biomec system, encompassing the development methodologies, technology stack, architectural decisions, and the challenges addressed during the project lifecycle. From project planning and team collaboration to the tools and coding strategies used, this chapter provides a transparent view of how Biomec was built. By adopting modern frameworks, secure coding practices, and a modular development workflow, the team aimed to deliver a system that is maintainable, scalable, and user-centric. The insights shared here reflect not only the technical decisions but also the collaborative effort and adaptability required to bring the project from concept to deployment.

3.2 Development Methodology

The successful implementation of the Biomec platform relies on a structured yet flexible development methodology. This section presents the strategies the team uses to manage the project effectively, ensure seamless collaboration, and adapt to evolving requirements. By combining agile principles with modern project management tools, the team maintains a consistent development pace and continuously improves the platform throughout its lifecycle.

3.2.1 Team Roles and Responsibilities

The development team is structured into three main roles:

- **Frontend developer:** responsible for building the user interface using React.
- **Backend developer:** tasked with creating APIs and managing the database using Laravel.
- **Team lead:** handles deployment, task coordination, and client communication.

These roles work in close coordination to ensure the smooth integration of backend services with the frontend interface and timely progress across all project areas.

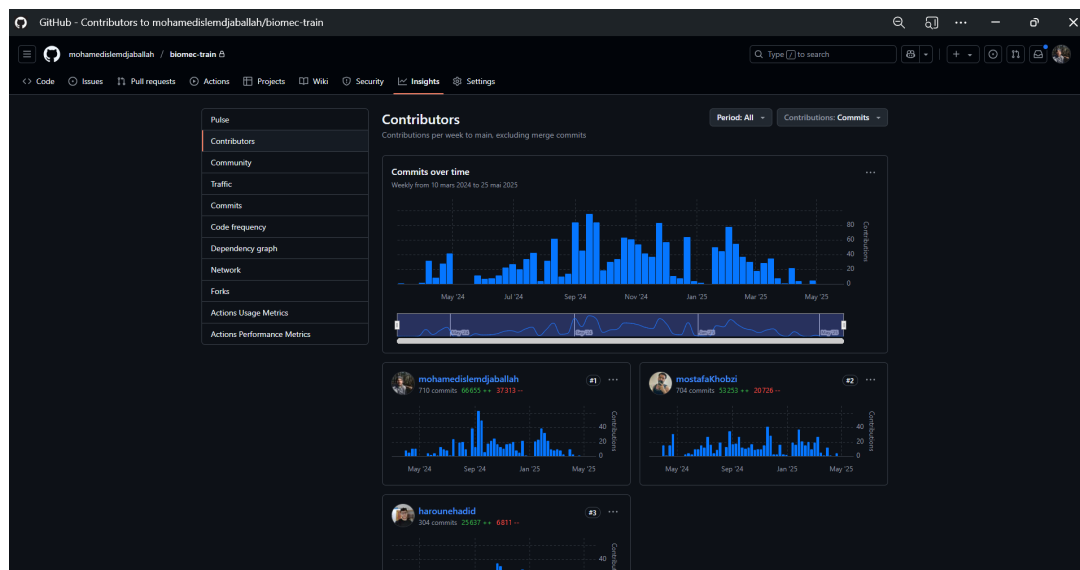


Figure 3.1: Development Team Contribution

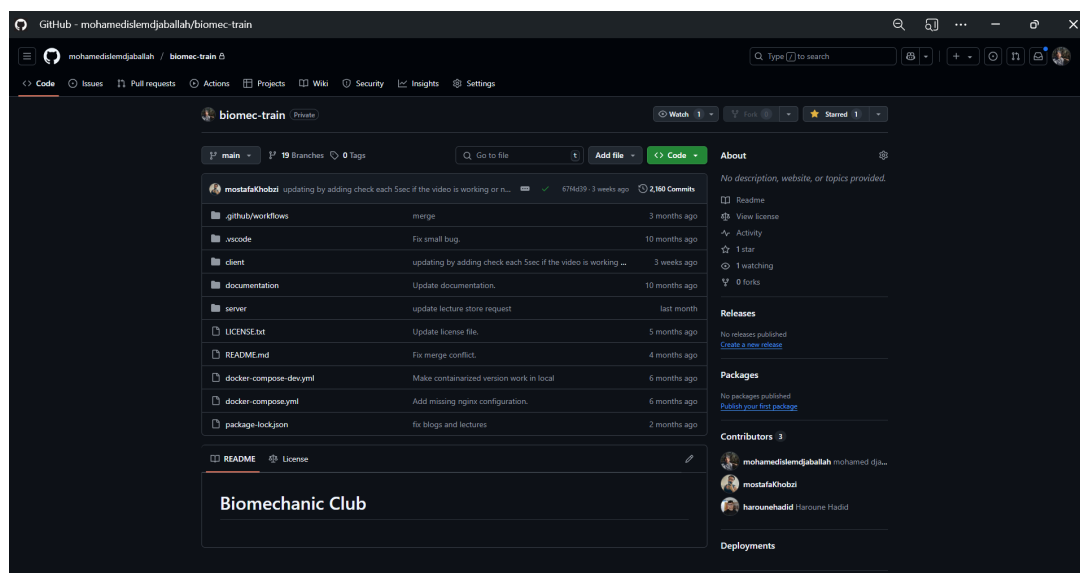


Figure 3.2: Project Repository in Github

3.2.2 Project Management Tools

The team uses several tools to support project planning, version control, and development coordination.

Git and **GitHub** are used for source code management, enabling collaborative development through pull requests, issue tracking, and branching strategies. This setup helps maintain code quality and minimizes merge conflicts, the next two figures represent the project in git 3.2 and the Contributors 3.1 in.

Trello, a flexible project management tool 3.3, was used to track tasks, assign

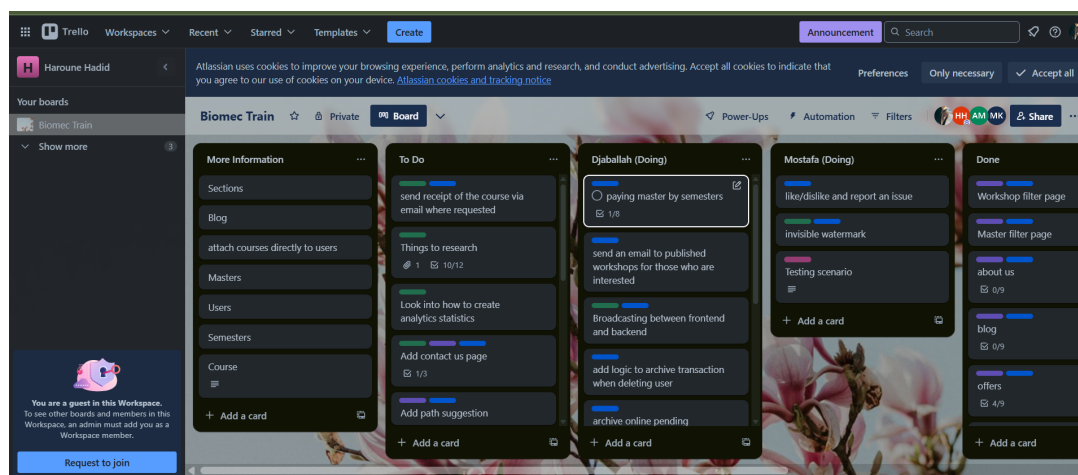


Figure 3.3: Project Board in Trello [3]

responsibilities, and visualize progress.

3.2.3 Agile Workflow with Kanban

We followed an agile methodology, specifically a Kanban style workflow implemented via Trello. Tasks were categorized into four columns: *Backlog*, *In Progress*, *Review*, and *Done*. This structure allowed us to monitor task flow, prioritize work efficiently, and respond quickly to changing requirements. Kanban's visual nature ensured that all team members had a clear and updated view of the project status.

3.2.4 Team Collaboration and Workflow

A decoupled development workflow was adopted to allow the frontend and backend to evolve independently while remaining synchronized. A predefined API contract guided both sides of development, reducing interdependencies and enabling parallel implementation.

Weekly meetings were held to review progress, resolve blockers, and plan upcoming tasks. This routine helped maintain alignment across the team and supported the continuous delivery of functional components. Meetings with the client took place on Zoom <https://zoom.us/>, and regular meetings between team members were held on Discord <https://discord.com/>.

3.3 Technology Stack

The architecture of the e-learning platform was constructed using a modern and modular technology stack that ensures robustness, scalability, and maintainability. The stack encompasses tools and frameworks for backend and frontend development, database management, third party service integration, and DevOps infrastructure. Our selection of these technologies was based on our team's evaluation of project requirements, scalability needs, and our experience with similar projects.

3.3.1 Backend Stack

The backend of the platform was developed using **Laravel**, a modern PHP framework known for its expressive syntax, modular design, and strong ecosystem. It supports rapid development, built in security mechanisms, and seamless integration with various packages. Our choice of Laravel was informed by our prior experience and its proven reliability in similar educational platforms.

- **Laravel Fortify** and **Sanctum**: Provide secure user authentication and session management with CSRF protection, allowing users to safely register, log in, and manage sessions.
- **Laravel Reverb** and **Echo**: Facilitate real-time features such as notifications, live discussions, and collaborative activities using WebSocket communication.
- **Pest**: A PHP testing framework offering readable and expressive syntax, supporting both unit and feature testing, which ensures the reliability of backend functionalities.
- **Axios**: Used in the Frontend to make asynchronous API requests to Laravel controllers, supporting RESTful interactions with efficient error handling and token-based authentication.

3.3.2 Database Management System

Data integrity, availability, and performance were critical for managing user accounts, course progress, assessments, and multimedia content. We selected PostgreSQL after comparing

several database solutions, prioritizing advanced data types and strong ACID compliance.

- **PostgreSQL:** A powerful, open-source relational database management system selected for its support for advanced data types, full-text search, and strong ACID compliance.
- **TablePlus:** A modern GUI client used during development for visual database modeling, query testing, schema design, and performance analysis <https://tableplus.com/>.

The schema design adheres to normalization principles, reducing redundancy and improving query efficiency.

3.3.3 Frontend Stack

The user interface is built using **React**, a popular JavaScript library ideal for developing scalable and interactive single page applications. Styling is implemented using **TailwindCSS**, which promotes clean and maintainable UI code. Our decision to use React and TailwindCSS was based on our team's expertise and the need for a responsive, maintainable frontend.

- **React:** Enables creation of reusable components, efficient DOM updates, and seamless user interactions, all essential for a responsive educational experience.
- **TailwindCSS:** A utility-first CSS framework that allows developers to apply styling directly in markup with high flexibility, ensuring consistency across devices and screen sizes.

3.3.4 Third Party Services

The platform integrates several external services to handle video hosting, secure transactions, file storage, and email communication. These integrations were chosen after our analysis of available solutions and their suitability for the platform's requirements.

- **Stripe:** A widely-used payment gateway that ensures secure, PCI-compliant transactions. It supports one-time payments and handles transaction logging and fraud detection.

- **Mailtrap:** A development and staging tool for email delivery testing, allowing safe and accurate testing of email notifications, confirmations, and password resets.
- **Vimeo:** Offers high-quality video hosting with privacy controls and analytics. The **Vimeo Player API** is used to embed videos with custom controls and restricted playback.
- **Amazon S3 (AWS):** Cloud storage service used to store and retrieve large media files, including images, PDFs, and course documents. It ensures scalability, high availability, and data durability.

3.3.5 DevOps and Hosting

The deployment pipeline uses containerization and cloud hosting solutions to enable continuous integration and delivery. We adopted Docker and DigitalOcean based on our team's experience and the need for scalable, reliable deployment.

- **Docker:** Packages the application and its dependencies into containers, ensuring consistent environments across development, staging, and production.
- **DigitalOcean Droplet:** A scalable virtual machine hosting the Laravel application, database, and media assets. It supports reliable deployment with SSH access and resource monitoring.

3.4 Platform Implementation

This section describes the practical implementation of the Biomec platform, focusing on the integration between backend and frontend components, API design, and the addition of real time features. The implementation reflects our team's collaborative approach and the lessons learned from previous projects.

3.4.1 Backend and Frontend Integration

The system follows a **decoupled model**. Laravel serves as the backend API provider, exposing RESTful endpoints that React consumes. This separation allows independent deployment and

parallel development of both frontend and backend components.

3.4.2 API Design and Data Flow

The API is designed according to REST principles. Endpoints return standardized JSON responses with appropriate HTTP status codes. Authentication tokens are managed using Sanctum, and CORS is configured to enable secure cross-origin requests.

A clear separation of concerns is maintained between client and server: data validation and business logic reside on the server, while the client handles user interface rendering and user interactions.

3.4.3 Realtime Features

Real-time communication is implemented using Laravel Reverb combined with Echo and WebSocket integration, primarily to support user notifications and chat functionalities.

3.5 Implementation Practices

To ensure a maintainable and robust platform, we adopted coding best practices focused on code organization, optimization, and security. This section outlines the techniques and standards applied during development, based on our team's experience and industry standards.

3.5.1 Code Organization and Optimization

Both frontend and backend follow modular design patterns. Laravel controllers, services, and resource files are organized to promote clarity and maintainability. React components are structured by routes and features. Performance improvements include lazy loading, code splitting, and minimizing API payload sizes.

Static analysis tools and linters are integrated to uphold code quality. Functions and variables are named clearly, and comments are added as needed to facilitate understanding for future

developers.

3.5.2 Security Considerations

Security is addressed at multiple levels:

- HTTPS is enforced across all deployments.
- Laravel Fortify and Sanctum manage authentication and provide protection against CSRF attacks.
- Sensitive user data is encrypted in storage.
- Video content is watermarked before upload to prevent unauthorized sharing.
- Stripe is used to handle secure payment processing.

3.5.3 Code Testing and Quality Assurance

Ensuring code reliability and maintainability was a priority throughout the project. Automated testing was integrated into the development workflow using the Pest PHP testing framework for backend logic. These tools enabled us to write unit, feature, and integration tests, ensuring that new changes did not introduce regressions. Our approach to testing was shaped by our previous experience with large scale projects.

- **Unit Tests:** Focused on individual functions or classes, verifying their correctness in isolation.
- **Feature Tests:** Simulated user interactions with the API, such as registration, login, and course enrollment.
- **Continuous Integration:** Tests were run automatically on each pull request to maintain code quality.

The figure 3.5 shows the results of testing the course API requests in different scenarios. Here we see 21 tests passed with 1390 assertions in 7.53s, beside each test we see the duration

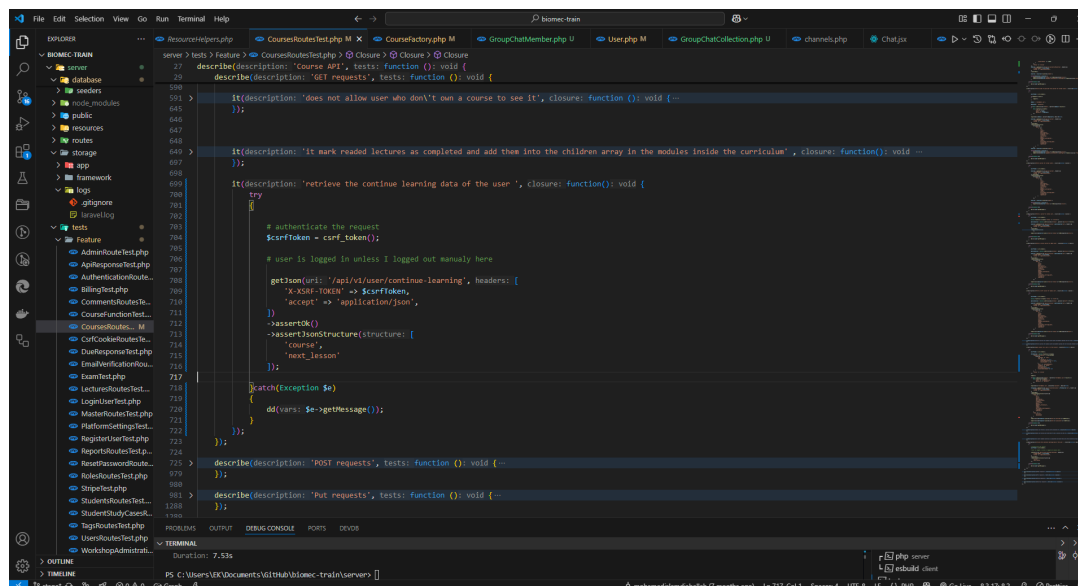


Figure 3.4: Example of a simple backend test using Pest

of it and the maximum is 67ms which is great but can be optimized more for a total of 199 course.

3.5.4 Example: Database Migration

Database migrations in Laravel provide a version-controlled way to manage schema changes. The figure 3.1 is migration for creating the courses table :

```

1 public function up()
2 {
3     Schema::create('courses', function (Blueprint $table) {
4         $table->increments('id');
5         $table->string('type', 1)->default('r');
6         $table->string('title');
7         $table->jsonb('about')->nullable();
8         $table->string('description')->nullable();
9         $table->unsignedInteger('enrolled_count')->default(0);
10        $table->jsonb('schedule');
11        $table->jsonb('curriculum');
12        $table->jsonb('media');
13        $table->jsonb('payment_information');
14        $table->jsonb('langs');
15        $table->boolean('is_bonus')->default(false);

```

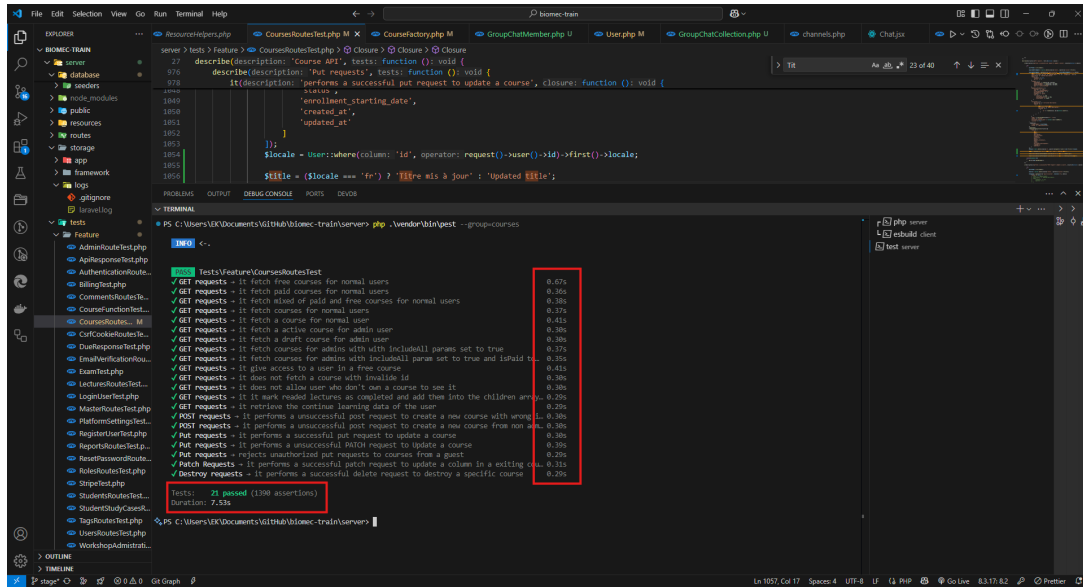


Figure 3.5: Test Results for Course Requests

```
16 $table->boolean('closed')->default(false);
17 $table->string('status', 10)->default('active');
18 $table->dateTime('enrollment_starting_date')->default(now()->addHour
19 $table->timestamps();
20 });
21 }
```

Listing 3.1: Course migration code example

This migration ensures that the database structure is consistent across all environments and can be rolled back if needed.

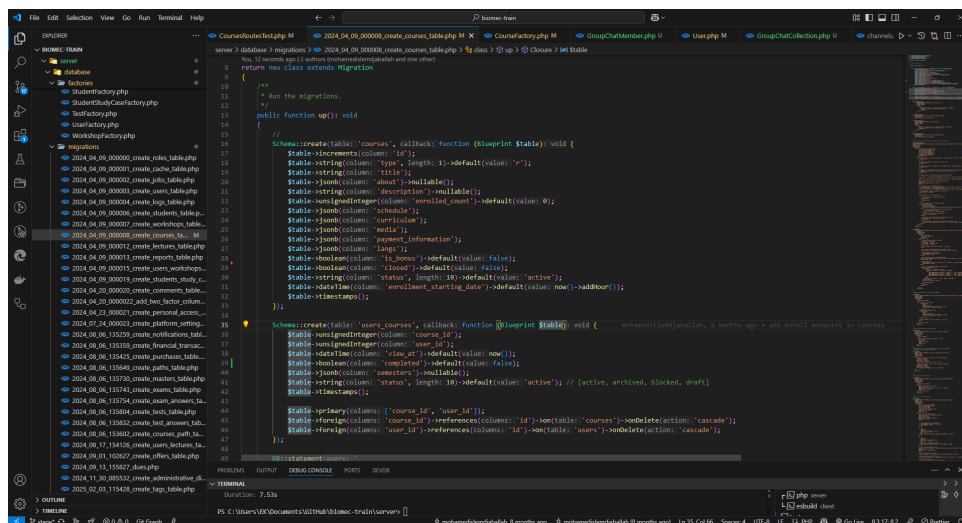


Figure 3.6: Example of a course migration in VSCode Editor

3.5.5 Example: User Validation and Creation

User input validation is critical for security and data integrity. In Laravel, validation is typically handled in controller methods or form request classes. Below is an example of validating and creating a user:

```
1 public function store(Request $request)
2 {
3     $validated = $request->validate([
4         'name' => 'required|string|max:255',
5         'email' => 'required|email|unique:users,email',
6         'password' => 'required|min:8|confirmed',
7     ]);
8
9     $user = User::create([
10        'name' => $validated['name'],
11        'email' => $validated['email'],
12        'password' => Hash::make($validated['password']),
13    ]);
14
15    // Additional logic (e.g., sending welcome email)
16    return response()->json($user, 201);
17 }
```

Listing 3.2: User creation in Laravel

This approach ensures that only valid data is stored in the database and that passwords are securely hashed.[3.8] [3.7]

```

server > app > Actions > Fortly > CreateNewUser.php > PHP IntelliSense > CreateNewUser > create
11 class CreateNewUser implements CreateNewUsers
22 public function create(array $input): mixed
51 >
52 > validate();
53 >
54 > catch (ValidationException $e) {
55 >
56 >     $errors = [];
57 >     foreach ($e->errors() as $error) {
58 >         foreach ($error as $err) {
59 >             if (strpos($err->message, 'required') !== false) {
60 >                 $splitErrors = explode(separator: ',', string: $err);
61 >                 foreach ($splitErrors as $splitError) {
62 >                     $errors[] = trim($splitError);
63 >                 }
64 >             } else {
65 >                 $errors[] = $err;
66 >             }
67 >         }
68 >     }
69 >
70 >     throw new HttpException(response: response()->json(data: ['errors' => $errors, 'status' => 422], status: 200));
71 >
72 >
73 >     return User::create(attributes: [
74 >         'role_id' => 2,
75 >         'username' => $input['username'],
76 >         'first_name' => strtolower(string: $input['first_name']),
77 >         'last_name' => strtolower(string: $input['last_name']),
78 >         'email' => $input['email'],
79 >         'password' => hash::make(value: $input['password']),
80 >     ]);
81 >
82 >
83 >
84 >
85 >
86 >
87 >
88 >
89 >
90 >
91 >
92 >
93 >
94 >
95 >
96 >
97 >
98 >
99 >
100 >

```

Figure 3.8: Creation of a User

```

server > app > Actions > Fortly > CreateNewUser.php > PHP IntelliSense > CreateNewUser > create
11 class CreateNewUser implements CreateNewUsers
22 public function create(array $input): mixed
23 >
24 > try {
25 >
26 >     $validators = Validator::make(data: $input, rules: [
27 >         'username' => ['required', 'string', 'max:32', 'unique:users'],
28 >         'first_name' => ['required', 'string', 'max:32'],
29 >         'last_name' => ['required', 'string', 'max:32'],
30 >         'email' => [
31 >             'required',
32 >             'string',
33 >             'max:64',
34 >             'unique:users',
35 >             'email' => [
36 >                 function ($attribute, $value, $fail): void {
37 >                     $fail($value);
38 >                 }
39 >             ]
40 >         ],
41 >         'password' => [
42 >             'required',
43 >             'string',
44 >             'min:8',
45 >             'confirmed',
46 >             'regex:/[a-z]/',
47 >             'regex:/[A-Z]/',
48 >             'regex:/[0-9]/',
49 >             'regex:/[!@#$%^&*]/',
50 >         ],
51 >     ]);
52 >
53 >     $input->validate($validators);
54 >
55 >     catch (ValidationException $e) {
56 >
57 >         $errors = [];
58 >         foreach ($e->errors() as $error) {
59 >             foreach ($error as $err) {
60 >                 if (strpos($err->message, 'required') !== false) {
61 >                     $splitErrors = explode(separator: ',', string: $err);
62 >                     foreach ($splitErrors as $splitError) {
63 >                         $errors[] = trim($splitError);
64 >                     }
65 >                 } else {
66 >                     $errors[] = $err;
67 >                 }
68 >             }
69 >         }
70 >
71 >         throw new HttpException(response: response()->json(data: ['errors' => $errors, 'status' => 422], status: 200));
72 >
73 >
74 >
75 >
76 >
77 >
78 >
79 >
80 >
81 >
82 >
83 >
84 >
85 >
86 >
87 >
88 >
89 >
90 >
91 >
92 >
93 >
94 >
95 >
96 >
97 >
98 >
99 >
100 >

```

Figure 3.7: Validation for a new user

3.6 Challenges and Solutions

Software development often involves navigating unexpected hurdles. During the Biomec project, we encountered several challenges related to communication, technical knowledge, and deployment logistics. This section outlines the major problems faced during the development cycle and explains the strategies and tools we used to address them effectively, drawing on our team's adaptability and collaborative problem solving.

3.6.1 Problems Encountered During Development

Several challenges were encountered during development:

- **Client communication:** The client was often unsure of the exact feature set required, leading to scope changes.
- **Inexperience:** Some technologies were new to the team, which slowed initial development.
- **Deployment hurdles:** Setting up Docker and configuring environment variables took extra time.

3.6.2 Resolution Strategies

By implementing structured communication and leaning on community resources and documentation, the team was able to overcome most of the development obstacles. This subsection describes the key solutions that allowed us to stay on track and improve the workflow despite the setbacks. To resolve these issues:

- We held regular meetings and used visual mockups to clarify requirements with the client.
- Dedicated learning sessions and online documentation helped the team master new tools.
- Deployment scripts and environment templates were created to standardize deployments across systems.

3.7 Platform Demonstration

The Biomec platform provides a comprehensive digital learning environment for orthodontic professionals. This section demonstrates the user journey through the platform's key interfaces, based on our team's design and implementation.

3.7.1 Home Page

The landing page (Figure 3.9) serves as the central hub, featuring:

- Free educational resources including aligner and biomechanics lectures
- Prominent workshop listings with visual highlights
- Master program advertisements with brief descriptions
- Multi-channel contact options (email form and social media links)

3.7.2 Workshop Section

The dedicated workshop area (Figure 3.10) displays:

- Filterable workshop catalog by module type
- Detailed program descriptions with learning objectives
- Date ranges and location information for in-person events
- Structured payment plans with installment schedules

3.7.3 Master Program Interface

The advanced training section (Figure 3.11) includes:

- Multi-year program timelines divided into semesters
- Transparent payment structures with installment details
- Curriculum access through secure academy communication channels

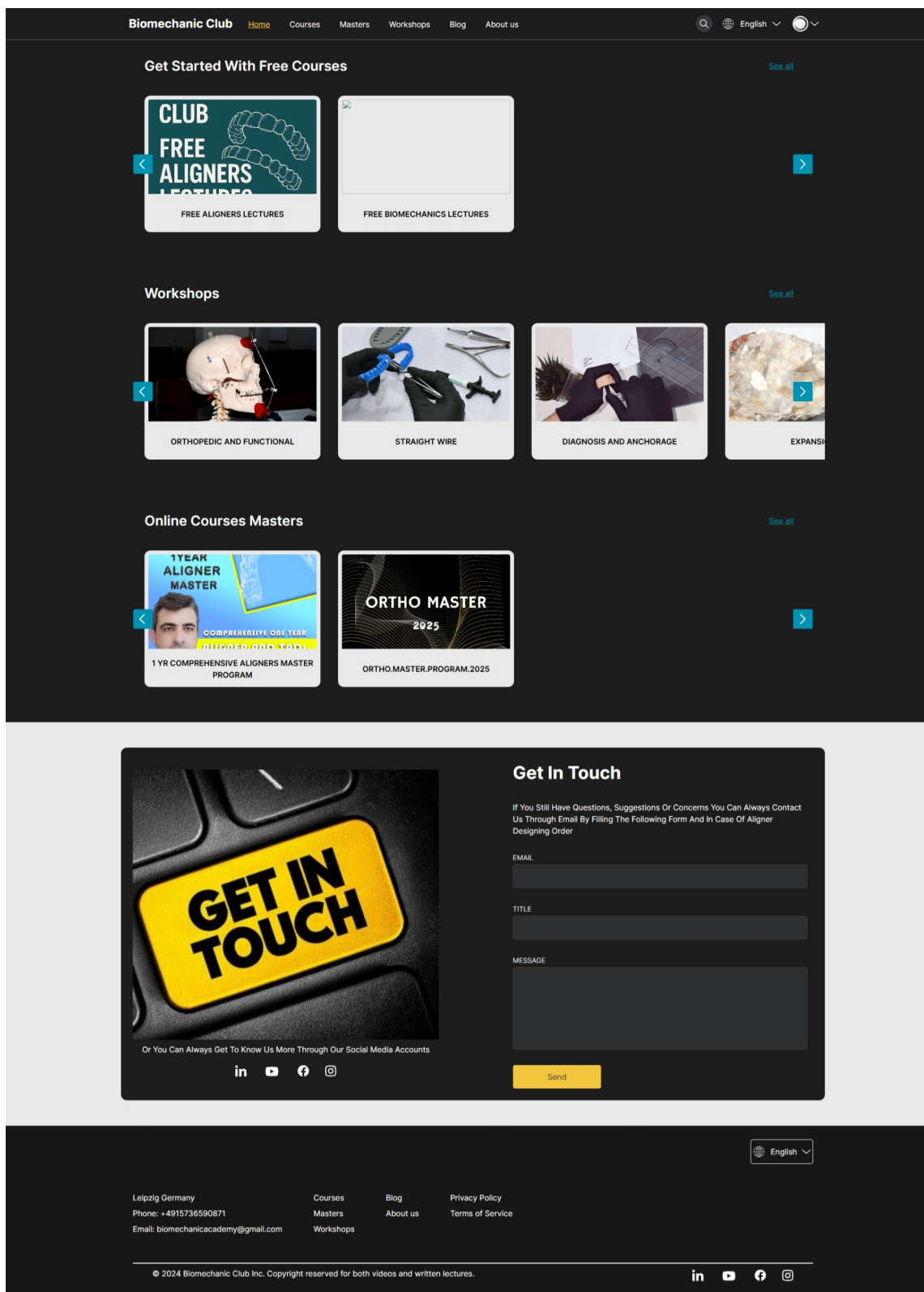


Figure 3.9: Platform homepage showing free resources and course promotions

The screenshot displays the 'ORTHOPEDIC AND FUNCTIONAL' workshop page on the Biomechanic Club website. The page features a dark theme with white and yellow text. At the top, there is a navigation menu with links for Home, Courses, Masters, Workshops, Blog, and About us. A search bar and language selector (English) are also present. The main content area includes a video player showing a 3D model of a human skull with orthopedic devices attached. Below the video, there are three key details: Start - End Date (03/07/25 - 04/07/25), Payment By Installments (60,000 DZD), and Location (BATNA, Algeria). A 'Payment Details' section lists two installments: 10,000 DZD (deadline 25/06/25) and 50,000 DZD (deadline 25/07/25). An 'About This Workshop' section provides a brief description and a list of topics to be covered, such as patient selection, growth stages, and orthopedic appliances. A 'Prerequisites' section indicates that no previous stages are required. The footer contains contact information for Leipzig, Germany, and social media links.

Biomechanic Club Home Courses Masters Workshops Blog About us

ORTHOPEDIC AND FUNCTIONAL

Take advantage of growth for the treatment of skeletal malocclusions.
Diagnose and treat skeletal cI and III with orthopedic devices

Enroll
3.4k 2025

Start - End Date
03/07/25 - 04/07/25

Payment By Installments
60,000 DZD

Location
BATNA, Algeria

Payment Details

Installment: 1 Amount: 10000 DZD	Deadline: 25/06/25
Installment: 2 Amount: 50000 DZD	Deadline: 25/07/25

About This Workshop

Vous pratiquez l'ortho et vous voulez tout apprendre sur le traitement orthopédique et fonctionnel ?

- Comment choisir ses patients
- Limites et contre indications du traitement orthopédique
- Déterminer le stade de croissance en détail
- Diagnostic des dysmorphoses et anomalies squelettiques chez l'enfant et la prise en charge détaillée par anomalie
- Appareillages orthopédiques fixes et amovibles indications et Biomécanique
- Appareillages et accessoires intrabuccaux et extra buccaux
- Démonstration sur le montage/ confection et activations

Prerequisites

Stages précédents OBLIGATOIRES

-There Is No Prerequisites -

English

Leipzig Germany
Phone: +4915736590871
Email: biomechanicacademy@gmail.com

Courses
Masters
Workshops

Blog
About us

Privacy Policy
Terms of Service

© 2024 Biomechanic Club Inc. Copyright reserved for both videos and written lectures.

in YouTube Facebook Instagram

Figure 3.10: Workshop detail view showing curriculum and registration information

The screenshot displays the 'Ortho Master Program 2025' page on the Biomechanic Club website. The page features a dark theme with a navigation bar at the top containing 'Home', 'Courses', 'Masters', 'Workshops', 'Blog', and 'About us'. A search bar and language selector (English) are also present. The main content area includes a hero section with the program title, a description ('3 YEARS ONLINE COMPREHENSIVE ORTHODONTICS PROGRAM THROUGH 6 SEMESTERS / 4 MONTHS AND A HALF EACH ONE / 2 LECTURES A WEEK'), and an 'Enroll' button. A video player shows a graphic with 'ORTHO MASTER 2025'. Below this, three key features are highlighted: '108 Weeks' (3-4 hours per week), 'Payment By Installments' (4,000 USD), and 'Self Paced' (Instructor-led on a course schedule). A 'Payment Details' table lists six installments of 650 USD each, with deadlines ranging from 25/11/25 to 25/05/28. The 'About this course' section states 'LEARN ORTHODONTICS FROM THE SCRATCH' and 'Course curriculum' is currently unavailable. The footer contains contact information for Leipzig, Germany, and social media links.

Biomechanic Club Home Courses Masters Workshops Blog About us

English

ORTHO.MASTER.PROGRAM.2025

3 YEARS ONLINE COMPREHENSIVE ORTHODONTICS PROGRAM
THROUGH 6 SEMESTERS / 4 MONTHS AND A HALF EACH ONE / 2
LECTURES A WEEK

Enroll

ORTHO MASTER 2025

108 Weeks
3 - 4 Hour per week

Payment By Installments
4,000 USD

Self Paced
Instructor-led on a course schedule

Payment Details

Installment: 1 Amount: 750 USD	Deadline: 25/11/25
Installment: 2 Amount: 650 USD	Deadline: 25/05/28
Installment: 3 Amount: 650 USD	Deadline: 25/11/26
Installment: 4 Amount: 650 USD	Deadline: 25/05/27
Installment: 5 Amount: 650 USD	Deadline: 25/11/27
Installment: 6 Amount: 650 USD	Deadline: 25/05/28

About this course

LEARN ORTHODONTICS FROM THE SCRATCH

Course curriculum

COMMUNICATE THROUGH ACADEMY EMAIL TO SEND THE CURRICULUM DETAILS

-There is No Curriculum -

English

Leipzig Germany
Phone: +4915736590871
Email: biomechanicacademy@gmail.com

Courses
Masters
Workshops

Blog
About us

Privacy Policy
Terms of Service

© 2024 Biomechanic Club Inc. Copyright reserved for both videos and written lectures.

in YouTube Facebook Instagram

Figure 3.11: Master program overview with payment schedule

The screenshot displays the 'Biomechanic Club' user profile management interface. The top navigation bar includes 'Home', 'Courses', 'Masters', 'Workshops', 'Blog', and 'About us'. The main content area is divided into a left sidebar with 'Settings' and 'General' sections, and a main 'Profile Details' section. The 'Profile Details' section contains the following fields: 'FIRST NAME' (mohamed islem), 'LAST NAME' (DJABALLAH), 'ADDRESS' (Address), 'COUNTRY' (Algeria), 'CITY' (City), 'PHONE NUMBER' (Phone Number), 'GENDER' (Female), and 'BIRTH DATE' (mm/dd/yyyy). A 'Save Changes' button is located at the bottom of the form. The footer includes contact information for Leipzig Germany, a phone number (+4915736590871), an email address (biomechanicacademy@gmail.com), and links to 'Courses', 'Masters', 'Workshops', 'Blog', 'About us', 'Privacy Policy', and 'Terms of Service'. Social media icons for LinkedIn, YouTube, Facebook, and Instagram are also present.

Figure 3.12: User profile management interface

3.7.4 User Account Management

The platform provides robust profile controls (Figure 3.12):

- Personal information editing (name, contact details)
- Secure password modification workflow
- Location and demographic data storage
- Profile picture upload functionality

3.7.5 Payment System

The checkout process (Figure 3.13) features:

- Clear billing summaries with total amounts
- Country-specific form fields

Biomechanic Club Home Courses Masters Workshops Blog About us

Checkout

Billing

ORTHOPEDIC AND FUNCTIONAL

One-Time payment price: 0 DZD Recurring Payment Total: 80000 DZD

Payment

FULL NAME
Enter Your Name

COUNTRY
Algeria

Upload Your Proof(s) Of Payment

Drag and drop your files here
or browse computer

In Progress
-Empty-

Completed / Failed Uploads
-Empty-

I agree to the [Terms of Use](#) and [Privacy Policy](#).

Review

ORTHOPEDIC AND FUNCTIONAL 0 DZD

Sub Total 0 DZD
Credit 0 DZD
Total 0 DZD

Pay

Figure 3.13: Secure payment processing interface

- Document upload for payment verification
- Terms and conditions acceptance requirement

3.7.6 About Us Section

The informational page (Figure 3.14) contains:

- Academy mission statement and service offerings
- Specialized training program descriptions
- Professional aligner design services information

Biomechanic Club Home Courses Masters Workshops Blog About us Q Search Login Sign up

About us

We Are The First Online Arab Academy Specialized In Teaching Orthodontics And Temporomandibular Joint Diseases

Our Story

The first online Arab academy specialized in teaching orthodontics and temporomandibular joint diseases .In Biomechanic Club Academy, we know that dentistry, like all sciences, is in rapid and continuous development, including orthodontics, that's why the content is updated and evidence based. Our programs are inspired by European and American master's programs .Ortho courses progress from diagnosis and biomechanic basics to diagnosis of advanced surgical cases and complex techniques. Through the courses, the student learns all the theoretical and applied knowledge necessary for the correct practice of the art of orthodontics, discussing and following up on cases too. The academy is distinguished by learning smart ortho, because we believe it is not a recipe to learn because every patient has his own different approach .To achieve a quick and safe result at the lowest cost, several techniques are combined. The bioprogressive philosophy is also followed in respecting the physiology of the tissues and TMJ, functional finishings, and the use of biomechanics and wire bending to avoid unwanted movements that increase bone resorption and prolong the treatment time.Advanced Comprehensive Aligner designing, biomechanics and treatment protocols programs .TMDs , orofacial pain and occlusion programs teach in detail the anatomy and all diseases of Temporomandibular Joint, in addition to appropriate treatments every participant in the academy is a success project, we offer to evaluate your needs in order to propose the appropriate programs and courses according to your goals. We also provide aligner services (design for Labs and doctors) If you would like help in choosing the appropriate courses and a plan for learning orthodontics, contact us

Terms and conditions Privacy policy About us Contact us English

© 2024 Biomec Train Inc. All rights reserved. in YouTube Facebook WhatsApp

Figure 3.14: Institutional information and service overview

The platform's documentation, including user manuals and client guides, is available at <https://drive.google.com/file/d/16S3yP9nq4e-0RBVsQyCc1T0uC8PCbtzR/view?usp=sharing> for detailed operational instructions.

Conclusion

The development of Biomec involved a multidisciplinary approach combining effective project management, modern software technologies, and a focus on best practices in software engineering. The system's decoupled architecture, secure implementation, and scalable deployment strategies highlight our commitment to building a robust platform. Despite facing challenges such as evolving client requirements and technical onboarding, we adapted quickly through continuous learning and collaboration. The experience gained during this project lays a strong foundation for future iterations, potentially incorporating advanced features like intelligent content delivery and deeper machine learning integration.

General Conclusion and Perspectives

This thesis has explored the conceptual foundations, analysis, and practical implementation of a dedicated e-learning platform tailored for dental surgery education. Throughout the first three chapters, we established the context and necessity for such a platform, critically evaluated existing solutions, and detailed the design and realization of a robust, scalable system based on our own research, analysis, and development work.

The initial chapter provided a comprehensive overview of web applications and their transformative impact on digital education. We highlighted, through our own comparative analysis, the limitations of current e-learning platforms, particularly in specialized domains like dental surgery, where hands-on practice, tailored feedback, and secure content delivery are paramount. This assessment justified our decision to pursue a custom-built solution that aligns with both pedagogical and operational requirements.

In the second chapter, we conducted a thorough analysis of the Biomechanic Club Academy's previous online learning methods, identifying the inefficiencies and fragmentation inherent in using multiple, non-integrated tools. Building on our findings, we articulated the objectives and requirements for a unified platform, emphasizing user-centered design, security, scalability, and performance. The chapter culminated in a detailed conception of the system's architecture, database schema, and user workflows, all informed by our direct engagement with the needs of learners and instructors.

The third chapter translated these conceptual designs into a concrete implementation. By leveraging modern technologies such as Laravel, React, PostgreSQL, and Stripe, we achieved a high standard of maintainability, security, and user experience. The modular architecture supports parallel development and future extensibility, while the integration of real-time features and secure payment processing addresses the unique needs of dental education. Our hands-on

approach throughout the development process ensured that the platform was built to meet real-world requirements.

Importantly, the platform is now fully hosted and accessible online. Learners and instructors are actively using the system to enroll in courses, participate in workshops, and access educational resources. This real-world deployment demonstrates the platform's practical value and its impact on dental surgery education, reflecting the effectiveness of our design and implementation choices.

Perspectives and Future Work

While the current platform marks a significant advancement in digital dental education, several avenues remain for further enhancement, many of which have been identified through our ongoing evaluation and user feedback:

- **Community Forum Integration:** Introducing a dedicated forum will foster peer-to-peer interaction, collaborative problem-solving, and ongoing professional discourse. This feature will support both synchronous and asynchronous communication, enabling learners and instructors to share experiences, discuss clinical cases, and build a vibrant learning community.
- **Advanced Recommendation System:** Leveraging machine learning and data analytics, an intelligent recommendation engine can personalize the learning journey for each user. By analyzing user behavior, progress, and preferences, the system can suggest relevant courses, workshops, and resources, thereby increasing engagement and optimizing educational outcomes.
- **Enhanced Analytics and Reporting:** Future iterations may incorporate advanced analytics dashboards for both learners and administrators, providing deeper insights into learning patterns, content effectiveness, and platform usage.
- **Mobile Application Development:** Expanding the platform's accessibility through dedicated mobile applications will further support learners in diverse contexts and on-the-go scenarios.
- **Integration with External Educational Resources:** Establishing interoperability with other educational platforms and content providers can broaden the scope of available resources and foster interdisciplinary learning.

In conclusion, the work presented in these chapters lays a solid foundation for a next-generation e-learning platform in dental surgery. By continuously evolving to incorporate new technologies and pedagogical strategies, and by remaining responsive to user needs, our platform aspires to set a benchmark for digital education in specialized medical fields, ultimately contributing to improved learning experiences and professional development for dental practitioners.

Bibliography

- [1] R. Jagoda, “Web application architecture [complete guide & diagrams],” 2024, accessed: 2025-05-04. [Online]. Available: <https://www.softkraft.co/web-application-architecture/>
- [2] Rajkumar, “Software architecture: One-tier, two-tier, three tier, n tier,” 2025, accessed: 2025-05-23. [Online]. Available: <https://www.softwaretestingmaterial.com/software-architecture/>
- [3] “Trello official website,” <https://trello.com/>, accessed: 2025-05-25.
- [4] M. Culduz, *Benefits and Challenges of E-Learning, Online Education, and Distance Learning*. IGI Global, 2024, pp. 1–27.
- [5] “11 best elearning platforms for 2025,” <https://www.skilljar.com/blog/best-elearning-platforms/>, 2025, accessed: 2025-05-07.
- [6] G. Hota and D. V. Singh, “Youtube for academics: Exploring learning habits of post graduate students,” *Journal of Advanced Research in Journalism and Mass Communication*, vol. 8, no. 1, pp. 23–32, 2021. [Online]. Available: <https://journals.indexcopernicus.com/api/file/viewByFileId/1591855>
- [7] Z. Shen, “Learner engagement with youtube videos in informal online learning: An investigation of the effects of segmenting, signaling, and weeding,” *Communications of the Association for Information Systems*, vol. 53, pp. 342–363, 2023. [Online]. Available: <https://aisel.aisnet.org/cais/vol53/iss1/8/>
- [8] T. Felke-Morris, *Web Development and Design Foundations with HTML5*, 10th ed. Pearson, 2020. [Online]. Available: <https://www.pearson.com/en-us/subject-catalog/p/web-development-and-design-foundations-with-html5/P200000003268/9780136681540>

- [9] CIO Wiki, “Web application,” https://cio-wiki.org/wiki/Web_Application, 2023, accessed: 2025-05-04.
- [10] R. Patel, “Essential technologies for building modern web applications,” <https://www.crestinfotech.com/essential-technologies-for-building-modern-web-applications/>, 2025, accessed: 2025-05-04.
- [11] B. Patel, “9 different types of web applications with use cases,” <https://www.spaceotechnologies.com/blog/types-of-web-applications/>, 2024, accessed: 2025-05-04.
- [12] Intellectsoft, “Web application architecture: The basics,” 2024, accessed: 2025-05-04. [Online]. Available: <https://www.intellectsoft.net/blog/web-application-architecture/>
- [13] GeeksforGeeks, “Top 10 front-end frameworks in 2025,” 2024, accessed: 2025-05-04. [Online]. Available: <https://www.geeksforgeeks.org/top-front-end-frameworks/>
- [14] “React official website,” [Online]. Available: <https://react.dev/>, 2025, accessed: 2025-05-23.
- [15] “Angular official website,” [Online]. Available: <https://angular.dev/>, 2025, accessed: 2025-05-23.
- [16] “Vue js official website,” <https://vuejs.org/>, 2025, accessed: 2025-05-23.
- [17] “Tailwind css official website,” <https://tailwindcss.com/>, 2025, accessed: 2025-05-23.
- [18] GeeksforGeeks, “Top 7 backend development frameworks [2025 updated],” 2024, accessed: 2025-05-04. [Online]. Available: <https://www.geeksforgeeks.org/frameworks-for-backend-development/>
- [19] “Express js official website,” <https://expressjs.com/>, 2025, accessed: 2025-05-23.
- [20] “Django official website,” <https://www.djangoproject.com/>, 2025, accessed: 2025-05-23.
- [21] “Ruby on rails official website,” <https://rubyonrails.org/>, 2025, accessed: 2025-05-23.
- [22] “Laravel official website,” <https://laravel.com/>, 2025, accessed: 2025-05-23.
- [23] “Asp.net core official website,” <https://dotnet.microsoft.com/en-us/apps/aspnet>, 2025, accessed: 2025-05-23.

- [24] P. Vadapalli, “14 must-know full stack frameworks for 2025 & how to find the best one,” 2025, accessed: 2025-05-04. [Online]. Available: <https://www.upgrad.com/blog/full-stack-developer-frameworks-to-learn/>
- [25] “Meteor official website,” <https://www.meteor.com/>, 2025, accessed: 2025-05-23.
- [26] “Next js official website,” <https://nextjs.org/>, 2025, accessed: 2025-05-23.
- [27] G. J. Longhurst, D. M. Stone, K. Duloherly, D. Scully, T. Campbell, and C. F. Smith, “Strength, weakness, opportunity, threat (swot) analysis of the adaptations to anatomical education in the united kingdom and republic of ireland in response to the covid-19 pandemic,” *Anatomical Sciences Education*, vol. 13, no. 3, pp. 301–311, 2020.
- [28] S. A. Salloum, M. Al-Emran, R. Khalaf, M. Habes, and K. Shaalan, “An innovative study of e-payment systems adoption in higher education: Theoretical constructs and empirical analysis,” *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 13, no. 6, pp. 68–83, 2019, accessed: 2024-11-03. [Online]. Available: <https://online-journals.org/index.php/i-jim/article/view/9875>
- [29] A. O. Insorio and J. A. Olivarez, “Utilizing facebook and messenger groups as platforms for delivering mathematics interventions in modular distance learning,” *International Journal of Professional Development, Learners and Learning*, vol. 3, no. 1, p. ep2109, 2021, open Access. [Online]. Available: <https://doi.org/10.30935/ijpdll/11290>
- [30] S. Hidayat, I. D. Lovita, Z. Zakiyah, Mimin, and A. Nurpratiwi, “The effectiveness of online learning using zoom meetings at elementary schools,” *International Journal of Technology in Education and Science (IJTES)*, vol. 6, no. 4, pp. 559–568, 2022. [Online]. Available: <https://files.eric.ed.gov/fulltext/EJ1357193.pdf>
- [31] H. K. Aladsani, “University students’ use and perceptions of telegram to promote effective educational interactions: A qualitative study,” *International Journal of Emerging Technologies in Learning (IJET)*, vol. 16, no. 09, pp. 182–197, 2021, accessed: 2025-05-23. [Online]. Available: <https://doi.org/10.3991/ijet.v16i09.19281>
- [32] M. A. Chatti, M. R. Agustawan, M. Jarke, and M. Specht, “Toward a personal learning environment framework,” *International Journal of Virtual and Personal Learning Environments*, vol. 1, no. 4, pp. 66–85, 2010.
- [33] R. E. Mayer, *Learning and Instruction*. Pearson Education, 2003, interactive learning and active engagement.

- [34] T. Anderson and J. Dron, “The case for using scalable, distributed systems in online learning,” *The International Review of Research in Open and Distributed Learning*, vol. 5, no. 2, pp. 1–10, 2004, accessed: 2025-05-27.
- [35] OWASP Foundation, “Owasp top 10: The ten most critical web application security risks,” <https://owasp.org/www-project-top-ten/>, 2023, accessed: 2025-05-27.
- [36] D. Norman, *The Design of Everyday Things*, revised and expanded ed. New York, NY: Basic Books, 2013.
- [37] J. Nielsen, “10 usability heuristics for user interface design,” <https://www.nngroup.com/articles/ten-usability-heuristics/>, 1995, updated: 2024-01-30. Accessed: 2025-05-27.
- [38] International Organization for Standardization, “Ergonomics of human-system interaction — part 210: Human-centred design for interactive systems,” ISO, Tech. Rep. ISO 9241-210:2010, 2010, international usability standard.
- [39] World Wide Web Consortium (W3C), “Web Content Accessibility Guidelines (WCAG) 2.1,” <https://www.w3.org/TR/WCAG21/>, 2018, accessed: 2025-05-27.