

**Project titel:** **Educational Website for Students**

**Project Type:** Graduation Project – Final Year Submission

**Prepared by:**

* Sama Abdel Latif Mira
* Aya Ali Yahya
* Abeer Al-Zaini Ahmed
* Alaa mohamed Ahmed Al-Gharib
* Shaima Abdel Latif Musa

**Faculty:** Faculty of Specific Education

Mansoura university

**Supervisors:**

* Dr. Mohamed Ghanem
* Dr. Amani El Gamal  
   **2025**

## **📄 Table of Contents**

## **Introduction .............................................. Page 1**

## **User Stories ............................................. Page 3**

## **Project Structure ....................................... Page 5**

## **System Architecture ..................................... Page 7**

## **Database Design and Models Explanation .......... Page 10**

## **Implementation Details .................................. Page 13**

## **Testing and Results Analysis ............................ Page 16**

## **Challenges and Solutions ............................... Page 18**

## **Future Improvements and Enhancements .......... Page 20**

## **Conclusion ............................................ Page 22**

## **References ............................................ Page 23**

## **Appendix ............................................. Page 24**

## **List of Figures/Diagrams ............................. Page 25**

### **🟩 List of Figures/Diagrams**

### **This section provides an overview of the visual illustrations included in this project documentation. These diagrams and screenshots support the textual descriptions by clarifying the architecture, functionality, and user interface of the educational platform.**

| **Figure No.** | **Figure Name** | **Page** |
| --- | --- | --- |
| **Figure 1** | **Entity-Relationship Diagram (ERD)** | **Page 15** |
| **Figure 2** | **Student Dashboard Screenshot** | **Page 20** |
| **Figure 3** | **Admin Dashboard Screenshot** | **Page 21** |
| **Figure 4** | **Course Page Screenshot** | **Page 22** |

### 

### **Figure 1: Entity-Relationship Diagram (ERD)**

### **The Entity-Relationship Diagram (ERD) provides a visual representation of the database structure for the educational website project. It illustrates the key tables (entities) such as users, levels, courses, lessons, materials, and quizzes, along with their relationships. This diagram helps clarify how data flows within the system, ensuring a well-structured and scalable database design that supports the platform’s features and requirements.**



### 

### **Figure 2: Admin Dashboard and Login Screenshots**

| **Description** | **Details** |
| --- | --- |
| **Overview** | **This figure includes screenshots showcasing the Admin Login Page and the Admin Dashboard Home Page.** |
| **Key Screens Highlighted** | **- Admin Login Page: This page provides a secure login form for administrators, requiring an email and password.****- Admin Dashboard Home Page: The main interface for administrators to manage educational content, courses, levels, and students.** |
| **Purpose** | **To highlight the secure authentication process for administrators and the comprehensive control panel available after successful login.** |
| **Features Demonstrated** | **- Clean and secure login page for administrator access.****- Dashboard overview with navigation to manage key aspects of the educational website.** |

## **📄 1. Introduction**

In the era of digital transformation, education systems worldwide are increasingly turning to online platforms to provide flexible, interactive, and student-centered learning experiences. This project, titled **"Educational Website for Students,"** aims to create a modern, engaging, and accessible educational platform that addresses the needs of students across different age groups and learning levels.

The goal of this project is to bridge the gap between traditional classroom teaching and modern digital education by offering a robust and intuitive online platform. It provides a variety of high-quality educational materials tailored to the unique learning needs of each student group. Moreover, it empowers teachers and administrators with tools to manage content effectively and track student progress in real-time.

This document outlines the comprehensive plan and technical implementation details of the platform. It covers all stages, from problem analysis and system design to database architecture, development, and future improvements. Throughout this document, you will find detailed explanations of how the system was built using Laravel, Vue.js, and other modern technologies, as well as screenshots and diagrams to illustrate its functionality.

## **🚀 2. Project Structure**

The **project structure** is designed to ensure a systematic and organized approach to developing the educational website. It includes clear sections that address both technical and functional aspects of the platform, making sure it is intuitive, engaging, and aligned with modern educational needs.

This section is divided into several key components:

### **2.1 Analyzing the Problem and Determining the Goal**

* **The Problem:** Traditional educational methods often lack flexibility and fail to meet the diverse learning needs and paces of students. Many learners struggle to find quality supplemental resources beyond the classroom, and there is no single unified platform that caters to multiple educational levels and adapts to each student’s requirements.
* **The Project Goal:** The primary objective of this project is to build a comprehensive educational website that:  
   ✅ Offers an intuitive and interactive learning experience for students of all levels.  
   ✅ Provides engaging, age-appropriate educational content.  
   ✅ Allows teachers and administrators to manage and update content efficiently.  
   ✅ Includes tools to track student progress and adapt learning paths to individual needs.

### **2.2 Identifying Target Groups**

The platform caters to five main student groups, each with unique learning needs:

* **Primary School Students** – Basic literacy, numeracy, and foundational knowledge.
* **Preparatory Students** – A balance of foundational and advanced topics.
* **Secondary Students** – Specialized subjects preparing for higher education or careers.
* **University Students** – Advanced content to support higher studies and skills.
* **Professional Students** – Learners seeking advanced, professional-level training.

### **2.3 Site Function Planning**

The platform’s core features include:  
 ✅ **Student Registration:** A simple interface for new students to sign up with their name, email, birth date, and password.  
 ✅ **Determining Age and Educational Level:** Automatically suggests a level based on age or lets students choose manually.  
 ✅ **Educational Content:** Interactive, age-appropriate materials like videos, text explanations, and short quizzes.  
 ✅ **Student Control Panel:** A personalized dashboard showing progress, recommended content, and suggested lessons.

### **2.4 Technical Design and Development**

✅ **UI/UX Principles:** Simple, engaging, and responsive design for all devices.  
 ✅ **Technology Stack:**

* **Back-end:** Laravel (PHP) or Node.js.
* **Front-end:** Vue.js or React.
* **Database:** MySQL or MongoDB.

✅ **System Architecture:** Follows the **MVC (Model-View-Controller)** pattern, ensuring:

* Separation of concerns for maintainability.
* Efficient data retrieval and modular feature addition.
* Secure user authentication and role-based access control.

### **2.5 Educational Content Management**

✅ Content organized by educational level and type for easy access.  
 ✅ Short quizzes and tests to measure student progress.  
 ✅ Tools for admins to update or expand content regularly.

### **2.6 Flexibility and Customization**

✅ Students can:

* Let the system suggest their educational level automatically, or
* Choose a level manually during registration.

✅ **Bulk Student Data Import:** Admins can upload student data in bulk via Excel files to save time and improve onboarding.

### **2.7 Experimentation and Testing**

✅ Testing with a small group of educators to gather feedback on:

* Ease of registration
* Content quality
* Platform usability

✅ **Iterative Refinement:** Improving the platform based on real-world feedback to meet user needs.

### **2.8 Launch and Follow-Up**

✅ **Official Website Launch:** Releasing the platform after thorough testing.  
 ✅ **Marketing and Engagement:** Using social media and outreach to attract students of all ages.  
 ✅ **Ongoing Monitoring:** Continuously tracking user activity and fixing issues promptly.  
 ✅ **Feature Updates:** Adding new features and improvements based on user requests.

### **2.9 Summary of Project Structure**

This structured plan ensures that the educational platform meets modern standards for learning while staying adaptable to future updates and feedback.

## **🏗️ 3. System Architecture**

The **System Architecture** of this educational platform is designed to ensure scalability, maintainability, and a seamless experience for both students and administrators. It leverages modern software design principles and tools to create a robust and secure learning environment.

### **3.1 MVC Architecture Overview**

The project follows the **Model-View-Controller (MVC)** design pattern, which separates the application logic into three distinct layers:

* **Model:** Represents the data structure of the application (e.g., User, Course, Lesson models). It handles data retrieval, relationships, and business logic.
* **View:** Defines the user interface that displays data to the user. Views are built using Laravel Blade templates, providing a clean and dynamic presentation.
* **Controller:** Acts as the intermediary between the model and the view. It processes requests, manipulates data, and returns the appropriate response or view to the user.

✅ **Benefits:**

* **Separation of Concerns:** Makes it easier to maintain and update the application.
* **Scalability:** Facilitates adding new features without disrupting existing components.
* **Security:** Isolates user input and business logic for safer development.

### **3.2 System Components**

The platform includes the following key components:

✅ **Authentication & Middleware:**

* Uses Laravel’s built-in authentication for secure user login and registration.
* Role-based access control (RBAC) ensures that admins and students have appropriate permissions.
* Middleware layers protect sensitive routes and manage user sessions.

✅ **APIs & Routing:**

* RESTful routes for accessing various resources like courses, lessons, and quizzes.
* **Example Route:** /dashboard to show the student’s personalized dashboard.
* Routes are protected by middleware and use Laravel’s web routes file for clear organization.

✅ **Database Layer:**

* Uses **Eloquent ORM** to handle database interactions smoothly and efficiently.
* Relationships (e.g., hasMany, belongsTo) ensure data integrity across models.

### **3.3 System Diagrams and Visual Flow**

✅ **Entity-Relationship Diagram (ERD):** Illustrates the database structure and how tables like users, courses, lessons, and quiz\_results are interconnected.

✅ **System Flow Diagram (if included):** Shows how users move through the application, from registration to content viewing, quizzes, and progress tracking.

✅ **Figure 1: Entity-Relationship Diagram (ERD)** *(Place the ERD image here and include a caption like:)*

“This diagram illustrates the key tables in the platform’s database and their relationships, ensuring data consistency and scalability.”

### **3.4 Scalability and Flexibility**

✅ The architecture is built to support:

* **Future enhancements** (e.g., adding new educational resources or quizzes).
* **Multi-device compatibility** (responsive design for desktop, tablet, and mobile).
* **Efficient caching and queuing** (via Laravel’s built-in features like jobs and queues) for performance optimization.

### **3.5 Security Considerations**

✅ The platform integrates security best practices:

* **CSRF tokens** for form submissions.
* **Hashed passwords** using Laravel’s Hash facade.
* **Validation rules** for input fields, ensuring only clean, accurate data is stored.

## **🗂️ 4. Database Design and Models Explanation**

The **Database Design** is a crucial aspect of this educational platform, ensuring that all data is stored efficiently, securely, and in a way that supports scalability and future expansion.

### **4.1 Database Overview**

✅ The database uses a **relational design** (MySQL or MongoDB), managed by Laravel’s **Eloquent ORM**.  
 ✅ All tables have primary keys (id) and foreign keys where needed, ensuring data integrity and relational mapping.

### **4.2 Key Tables and Models**

Below is a breakdown of the **main tables** and their purposes:

#### **🔹 Admins and Admin Sessions**

* **admins**: Stores admin user data.  
   Columns: id, name, email, password, remember\_token, timestamps.  
   ✅ Used for admin login and management.
* **admin\_sessions**: Tracks admin sessions.  
   Columns: id, admin\_id (FK), ip\_address, user\_agent, payload, last\_activity.  
   ✅ Ensures secure session tracking and management.

#### **🔹 Users and Sessions**

* **users**: Stores student information.  
   Columns: id, name, birth\_date, level\_id (FK), email, password, remember\_token, timestamps.  
   ✅ Each student is linked to a level.
* **sessions**: Tracks active student sessions.  
   Columns: id, user\_id (FK), ip\_address, user\_agent, payload, last\_activity.

#### **🔹 Levels**

* **levels**: Represents educational stages.  
   Columns: id, name, description, order, min\_age, max\_age, timestamps.  
   ✅ Levels determine content access for students.

#### **🔹 Courses and Course Materials**

* **courses**: Stores course information.  
   Columns: id, title, description, image, pdf, level\_id (FK), timestamps.  
   ✅ Courses are linked to levels.
* **course\_materials**: Extra resources for courses.  
   Columns: id, course\_id (FK), type (image, video, link, summary), content, timestamps.

#### **🔹 Lessons and Materials**

* **lessons**: Each course has multiple lessons.  
   Columns: id, course\_id (FK), title, description, timestamps.
* **materials**: Content within lessons.  
   Columns: id, lesson\_id (FK), type (video, text, image), url, title, content, timestamps.

#### **🔹 Quizzes and Results**

* **quizzes**: Linked to lessons for assessments.  
   Columns: id, lesson\_id (FK), title, timestamps.
* **questions**: Questions within a quiz.  
   Columns: id, quiz\_id (FK), question\_text, timestamps.
* **question\_options**: Possible answers.  
   Columns: id, question\_id (FK), option\_text, is\_correct, timestamps.
* **quiz\_results**: Stores quiz results per student.  
   Columns: id, quiz\_id (FK), user\_id (FK), score, total, percentage, taken\_at, timestamps.

#### **🔹 System Tables**

* **cache / cache\_locks / jobs / job\_batches / failed\_jobs**:  
   Laravel system tables used for performance, caching, and job queue management.

### **4.3 Entity-Relationship Diagram (ERD)**

✅ The ERD provides a visual representation of the database structure, showing the relationships between key tables.

✅ **Figure 1: Entity-Relationship Diagram (ERD)** *(Place the ERD image here with a short caption like:)*

“This diagram visually depicts the relationships between core tables in the platform’s database, ensuring data integrity and supporting scalable growth.”

### **4.4 Summary of Relationships**

✅ **User (users)**

* Belongs to a level
* Can have multiple quiz results

✅ **Level (levels)**

* Has many courses

✅ **Course (courses)**

* Belongs to a level
* Has many lessons
* Has many course materials

✅ **Lesson (lessons)**

* Belongs to a course
* Has many materials
* Has many quizzes

✅ **Quiz (quizzes)**

* Belongs to a lesson
* Has many questions
* Has many quiz results

✅ **Question (questions)**

* Belongs to a quiz
* Has many question options

## **🛠️ 5. Implementation Details**

The **Implementation Details** section provides a closer look at how the platform was built technically. It highlights key Laravel features, Blade templates, and example code snippets that demonstrate how different components of the system work together.

### **5.1 Key Routes and Controllers**

✅ **Example Route:** In the web.php file:

php

CopyEdit

Route::get('/dashboard', [DashboardController::class, 'index'])->middleware(['auth', 'verified'])->name('dashboard');

This route:

* Points to the index() method in DashboardController.
* Is protected by the auth and verified middleware to ensure only logged-in, verified users can access the dashboard.
* Is named dashboard for easy referencing in Blade templates.

✅ **Example Controller Method:** Here’s a simplified example from the DashboardController:

php

CopyEdit

public function index(): View

$courses = Course::where('level\_id', Auth::user()->level\_id)->get();

return view('dashboard', compact('courses'));

}

This function:

* Retrieves all courses that match the logged-in student’s level.
* Passes them to the dashboard.blade.php view for display.

### **5.2 Blade Templates and Layouts**

The platform uses Laravel’s **Blade templating engine** to build modular and reusable views.  
 ✅ **Example:**

blade

CopyEdit

<x-app-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl">Dashboard</h2>

</x-slot>

<div class="py-6">

<!-- Dashboard Content -->

</div>

</x-app-layout>

Here:

* x-app-layout provides a consistent layout for student pages.
* The header slot shows the page title.
* The main content is placed within the layout’s slot.

### **5.3 User Authentication and Middleware**

✅ **Laravel Authentication:** The system uses Laravel’s built-in **authentication features** (e.g., Laravel Breeze or Fortify). This provides:

* Secure login and registration flows.
* Password hashing and reset features.
* Email verification.

✅ **Middleware Layers:** Example:

* auth – Ensures that only logged-in users can access certain routes.
* verified – Ensures users have verified their email addresses before accessing sensitive parts of the platform.

### **5.4 Role-Based Access Control (RBAC)**

✅ Admins and students have different privileges:

* **Admins:** Full access to manage courses, levels, and student data.
* **Students:** Limited to viewing their own dashboard, taking quizzes, and tracking progress.

RBAC ensures that only authorized users can access or modify sensitive data.

### **5.5 Use of Components and Layouts**

✅ The system uses Blade components for modularity and maintainability:

* **Admin Layout:** x-admin-layout
* **App Layout:** x-app-layout
* **Guest Layout:** x-guest-layout

✅ Example usage in Blade:

blade

CopyEdit

<x-admin-layout>

<!-- Admin page content -->

</x-admin-layout>

## **✅ 6. Testing and Results Analysis**

Testing is a crucial part of the project to ensure the platform’s functionality, usability, and security. This section describes how testing was carried out and summarizes the key outcomes.

### **6.1 Testing Environment**

The project was tested locally using the following stack:

* **Laravel Framework:** v10
* **PHP Version:** 8.1
* **Database:** MySQL
* **Testing Tools:** Postman for API testing, manual form tests in the browser
* **Browsers:** Chrome and Firefox for cross-browser compatibility

✅ **Local Deployment:** The platform was run on **localhost** using Laravel’s built-in server (php artisan serve) to simulate a real-world environment.

### **6.2 User Flow Testing**

✅ **Student Registration:**

* Form tested with valid and invalid inputs (e.g., missing fields, incorrect email format).
* Birth date validation tested with different ages (below 3 years or over 100 years).
* System correctly suggests the appropriate level based on age.

✅ **Student Login & Dashboard:**

* Verified secure login and session management.
* Checked that students see the correct level’s courses and materials.

✅ **Admin Login & Dashboard:**

* Verified admin-specific views, including course and student management.
* Checked RBAC (Role-Based Access Control) to ensure restricted access for admins only.

✅ **Content Management:**

* Tested adding, editing, and deleting courses, lessons, and materials from the admin panel.
* Verified that updates are immediately reflected in the student dashboard.

### **6.3 Quiz & Results Testing**

✅ Students were able to:

* View quizzes linked to lessons.
* Submit answers, which were stored and linked to their user account.
* See quiz results displayed in their profile dashboard.

### **6.4 Screenshots of Testing Results**

*(Add screenshots here if available, e.g., student dashboard, quiz submission, admin course management, etc.)*

Example:

**Figure 2: Student Dashboard Screenshot** “This screenshot shows the student’s dashboard, including progress tracking and recommended content.”

### **6.5 User Feedback**

The platform was reviewed by a small group of teachers and supervisors. Key feedback included:

* 👍 Easy registration and login processes
* 👍 Clean, user-friendly interface
* 🔧 Suggested improvements for content categorization and navigation

## **⚙️ 7. Challenges and Solutions**

During the development of the educational website, several challenges were encountered. This section describes these issues and how they were resolved to ensure the platform is secure, scalable, and user-friendly.

### **7.1 Role-Based Access Control (RBAC)**

✅ **Challenge:** Ensuring that students and administrators have access only to the features and data relevant to their roles.

✅ **Solution:**

* Used **Laravel middleware** to restrict access to sensitive routes.
* Created separate dashboards for admins and students (x-admin-layout, x-app-layout).
* Implemented role-checks within controllers to ensure admins have full control over content management, while students only see their own progress and resources.

### **7.2 Bulk Data Import (Excel Upload)**

✅ **Challenge:** Allowing administrators to quickly import a large number of students without manual entry.

✅ **Solution:**

* Integrated **Excel file upload** functionality using packages like Maatwebsite/Laravel-Excel.
* Validated and processed Excel data to create new student records linked to their respective levels.

### **7.3 Dynamic Content Management**

✅ **Challenge:** Ensuring that courses, lessons, and materials can be updated dynamically without affecting existing student progress.

✅ **Solution:**

* Used **Eloquent ORM** relationships (hasMany, belongsTo) for efficient data management.
* Built dynamic admin dashboards to allow real-time updates and immediate reflection of new content in the student dashboards.

### **7.4 Age and Level Assignment**

✅ **Challenge:** Automatically assigning students to the correct level based on their age during registration.

✅ **Solution:**

* In the RegisteredUserController, calculated the age using the **Carbon** library.
* Queried the levels table to find the level matching the student’s age range (min\_age and max\_age).
* Automatically assigned the student to the relevant level.

### **7.5 Frontend Responsiveness**

✅ **Challenge:** Ensuring that the platform is visually consistent and works smoothly across devices (desktop, tablet, mobile).

✅ **Solution:**

* Used **Tailwind CSS** for a responsive design that adapts to different screen sizes.
* Tested views in multiple browsers and devices to ensure consistency.

## **🚀 8. Future Improvements and Enhancements**

While the platform already offers a solid foundation for digital learning, there are several future enhancements that could further improve its functionality, usability, and reach. These potential improvements will ensure the platform remains adaptable and continues to meet evolving educational needs.

### **8.1 Adding Online Quizzes and Assessments**

✅ **Description:** Currently, the platform includes basic quizzes linked to lessons. Future updates could integrate more sophisticated online assessments, such as:

* Timed exams.
* Automated grading.
* Progress analytics for students and teachers.

✅ **Benefit:** Enhances interactivity and provides more robust progress tracking.

### **8.2 Implementing Discussion Forums**

✅ **Description:** Integrating forums or discussion boards where students can ask questions, share ideas, and collaborate with peers.

✅ **Benefit:** Encourages peer-to-peer learning and engagement.

### **8.3 Developing a Mobile App**

✅ **Description:** Creating a dedicated mobile app version of the platform for iOS and Android devices.

✅ **Benefit:** Improves accessibility and provides a smoother experience on smartphones and tablets.

### **8.4 Advanced Analytics and Reporting**

✅ **Description:** Adding analytics tools to provide detailed insights into:

* Student progress and performance.
* Popular courses and materials.
* Areas needing improvement.

✅ **Benefit:** Supports data-driven decision making for both students and educators.

### **8.5 AI-Powered Recommendations**

✅ **Description:** Using artificial intelligence to analyze student performance and suggest personalized learning paths.

✅ **Benefit:** Adapts the learning experience to the student’s unique strengths and areas for improvement.

### **8.6 Multilingual Support**

✅ **Description:** Expanding the platform’s reach by adding support for multiple languages.

✅ **Benefit:** Makes the platform accessible to a wider audience globally.

## **📝 9. Conclusion**

The **Educational Website for Students** project represents a significant step towards modernizing education by harnessing the power of digital learning tools. Throughout the project, careful planning, technical expertise, and a clear vision have been employed to create a platform that meets the needs of both students and educators.

✅ **Key Outcomes:**

* A responsive, user-friendly platform tailored to different educational levels.
* A robust system for managing courses, lessons, and student progress.
* Secure authentication and role-based access control to protect user data and maintain educational integrity.
* A scalable foundation that supports future growth and feature enhancements.

✅ **Impact:** By bridging the gap between traditional classroom environments and modern e-learning solutions, this project empowers students to learn at their own pace and educators to manage and improve their content delivery more effectively.

✅ **Future Potential:** With planned enhancements such as advanced quizzes, AI-powered recommendations, and mobile app support, this educational platform is well-positioned to become a valuable resource in the evolving world of digital education.

## **📚 10. References**

Below is a list of key references and resources used throughout the development of this project:

1️⃣ **Laravel Documentation**

* URL:<https://laravel.com/docs>
* Primary source for framework setup, authentication, middleware, and Eloquent ORM usage.

2️⃣ **Laravel Breeze/Fortify**

* URL:<https://laravel.com/docs/10.x/starter-kits>
* Starter kit and authentication best practices.

3️⃣ **Maatwebsite Laravel Excel**

* URL: https://docs.laravel-excel.com
* Used for bulk data import functionality (Excel files).

4️⃣ **Tailwind CSS**

* URL:<https://tailwindcss.com/docs>
* Used for responsive design and modern styling.

5️⃣ **Stack Overflow**

* URL:<https://stackoverflow.com>
* Used to resolve various coding challenges and implementation questions.

6️⃣ **e-Learning Global Reports**

* Example: “60% of students prefer online learning to traditional methods.” (e-learning global report)
* Used to support the introduction and highlight the importance of digital learning.

✅ *Note:* Additional references, such as blog posts or tutorials, can be included here if they were used during development.

## **📂 11. Appendix**

The appendix provides supporting materials, additional resources, and code snippets that were used or referenced during the project. It serves as a helpful resource for future developers or reviewers.

### **11.1 Key Seeders**

#### **🔹 LevelSeeder**

Responsible for creating default educational levels.

php

CopyEdit

Level::create([

'name' => 'Beginner',

'description' => 'For students aged 6 to 8.',

'min\_age' => 6,

'max\_age' => 8,

'order' => 1,

]);

#### **🔹 AdminSeeder**

Creates a default administrator for initial system access.

php

CopyEdit

Admin::updateOrCreate(

['email' => 'admin@gmail.com'],

[

'name' => 'Super Admin',

'password' => Hash::make('password123'),

]

);

#### **🔹 CourseSeeder**

Adds sample courses for each level.

php

CopyEdit

Course::create([

'title' => 'Basic English',

'description' => 'Tailored for the Beginner level.',

'level\_id' => $level->id,

]);

### **11.2 Example Controller**

#### **🔹 RegisteredUserController (Registration Handling)**

Key logic for validating registration, calculating age, and assigning levels:

php

CopyEdit

$age = Carbon::parse($request->birth\_date)->age;

$level = Level::where('min\_age', '<=', $age)

->where('max\_age', '>=', $age)

->first();

$user = User::create([

'name' => $request->name,

'email' => $request->email,

'birth\_date' => $request->birth\_date,

'level\_id' => $level?->id,

'password' => Hash::make($request->password),

]);

### **11.3 Example Layout**

#### **🔹 Admin Layout (layouts.admin.app)**

blade

CopyEdit

<x-admin-layout>

<x-slot name="header">

<h2 class="font-semibold text-xl">Admin Dashboard</h2>

</x-slot>

<main>

{{ $slot }}

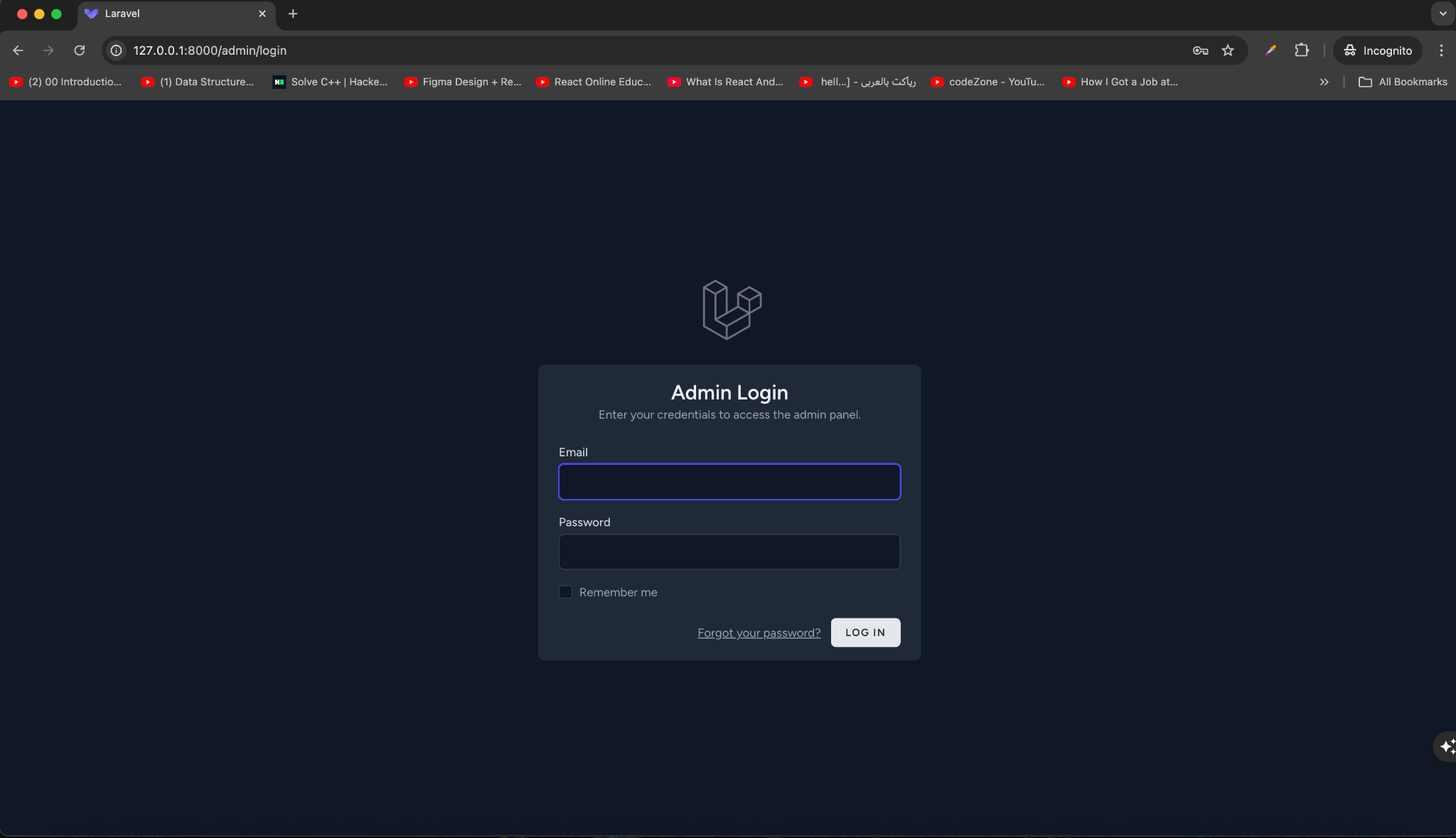
</main>

</x-admin-layout>

### **11.4 Screenshots**

#### **Admin Login Page**

This screenshot shows the secure **Admin Login Page**, where administrators enter their email and password to access the system. The design is simple and clean, ensuring easy access for administrators.

****

#### **Admin Dashboard Home Page**

This image presents the **Admin Dashboard Home Page**, which serves as the control center for managing courses, levels, student data, and more. The dashboard offers intuitive navigation and essential tools for content and user management.

****

#### **👤 Admin Profile Page**

The **Admin Profile Page** provides administrators with an intuitive and secure interface to manage their personal account details within the educational platform. This page is crucial for ensuring the accuracy of administrative information and maintaining system security.

Key sections and features on this page include:

##### **🟩 Profile Information**

This section allows administrators to:

* **Name:** View and update the administrator’s name (e.g., “Super Admin”).
* **Email:** View and update the email address used for administrative login and communication.  
   ✅ A **Save** button is provided to save any changes made to these fields.

##### **🟩 Update Password**

For enhanced security, administrators can update their password through a dedicated section:

* **Current Password:** Required to confirm the administrator’s identity.
* **New Password:** Field to set a secure new password.
* **Confirm Password:** Ensures that the new password is entered correctly.  
   ✅ A **Save** button is included to confirm and apply the new password.

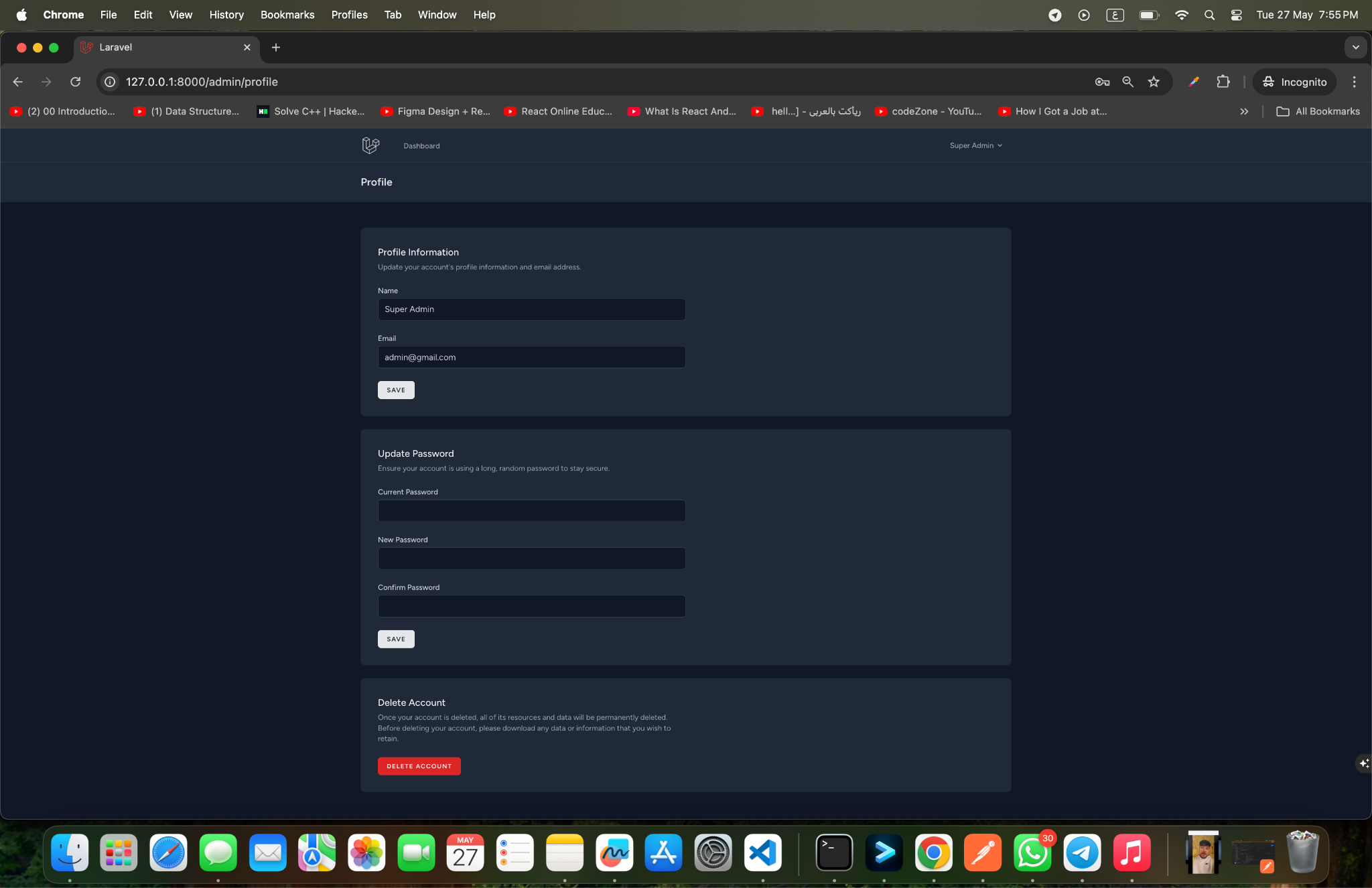
##### **🟩 Delete Account**

Administrators also have the ability to **permanently delete their account** if needed. This section includes:

* A clear warning indicating that **all resources and data will be permanently deleted** upon deletion.
* A recommendation to download any data or information before proceeding.
* A **Delete Account** button to execute the account deletion process.

✅ **Example: Admin Profile Page Overview**

| **Section** | **Description** |
| --- | --- |
| Profile Information | View and update name and email for the administrator. |
| Update Password | Change the current password to enhance security. |
| Delete Account | Option to permanently delete the administrator’s account and associated data. |



#### **👤 Student Creation Page**

This page enables administrators to add new students to the platform quickly and efficiently. The form is designed to be intuitive and easy to use, ensuring that essential student details can be entered and stored accurately.

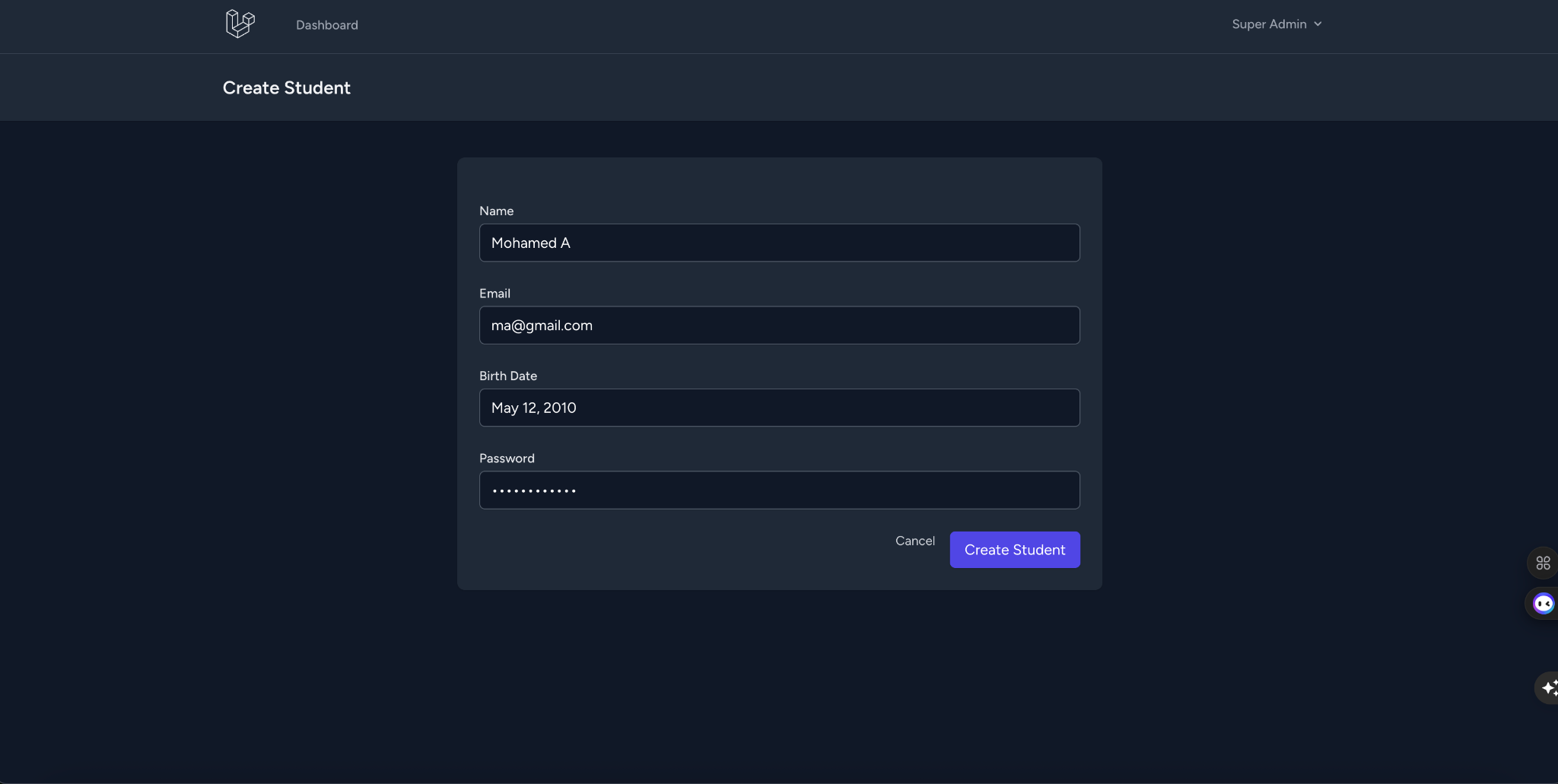
The **Student Creation Form** includes the following fields:

* **Name:** The full name of the student.
* **Email:** A unique email address used as the student’s identifier in the system.
* **Birth Date:** The date of birth to help automatically suggest the appropriate educational level.
* **Password:** A secure password for student login and access.
* **Cancel:** A button to clear the form and cancel the creation process if needed.
* **Create Student:** A button to submit the form and create the new student profile.

This form ensures data consistency and allows for quick onboarding of new students, making it easier for administrators to manage and maintain student records.

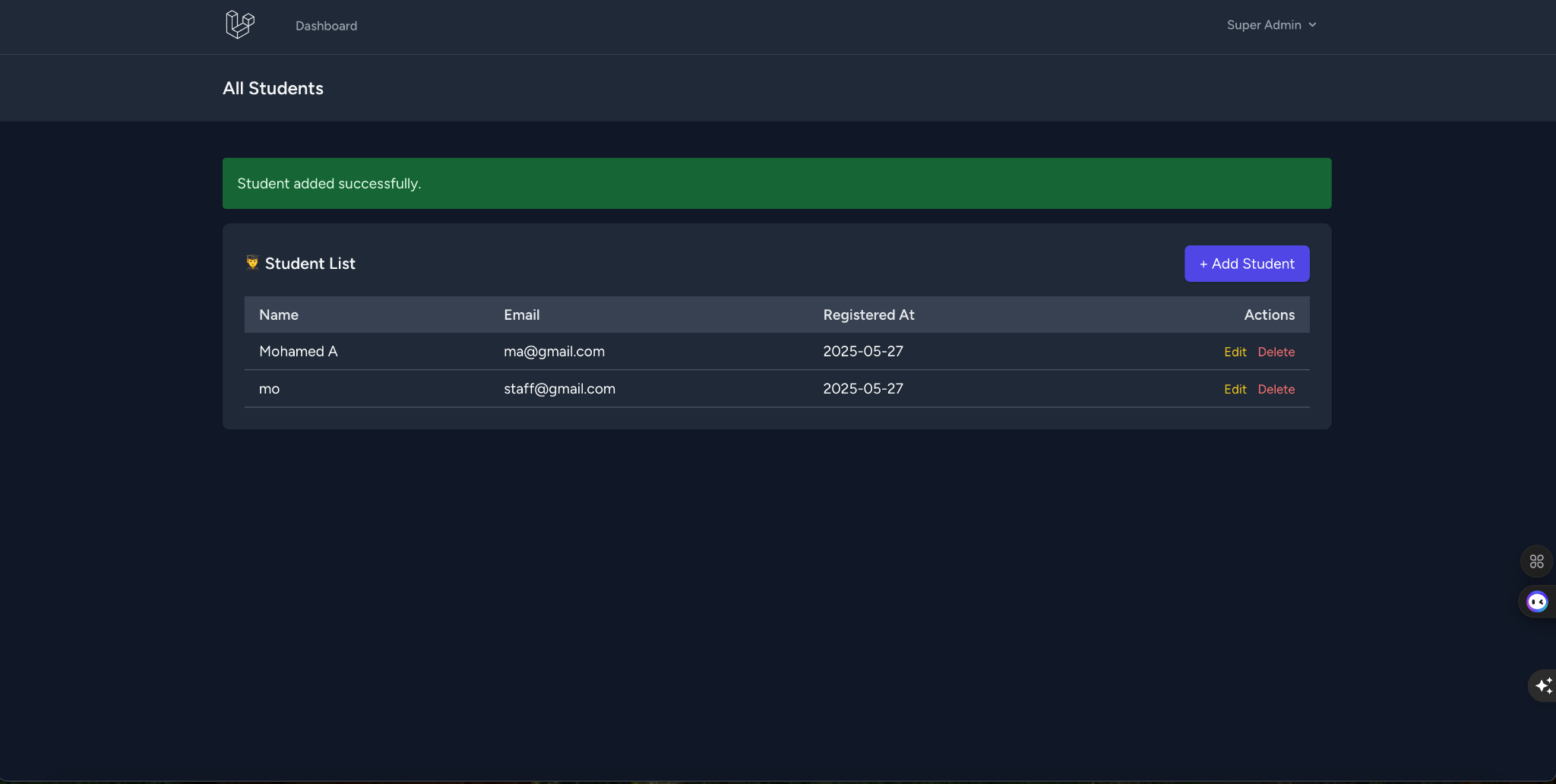
✅ **Example: Student Creation Form Entry**

| **Field** | **Example Entry** |
| --- | --- |
| Name | Ahmed Ali |
| Email | ahmed.ali@example.com |
| Birth Date | 2010-06-15 |
| Password | \*\*\*\*\*\*\*\* (hidden) |



#### **Student Listing Page**

This image showcases the **Student Listing Page**, a table view that lists all registered students along with options to edit or delete student profiles. This interface ensures effective data management and accessibility.

****

#### **Level Creation Page**

**This page allows administrators to create new educational levels for the platform. Each level represents a specific educational stage (such as Primary, Preparatory, Secondary, etc.) and helps to structure the learning content accordingly. Administrators can define the name, description, age restrictions, and display order to ensure proper categorization and easy navigation.**

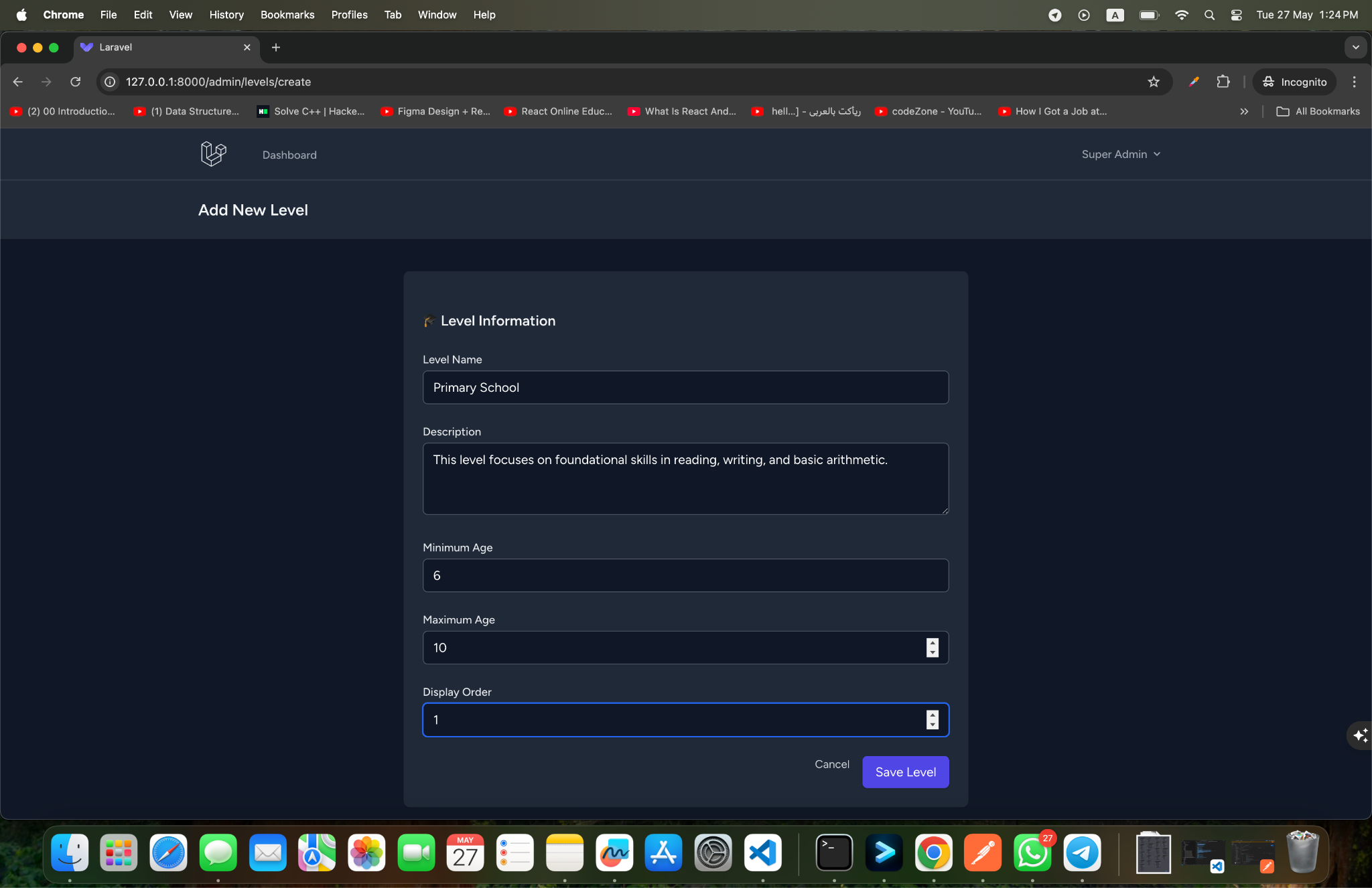
**The Level Creation Form includes the following fields:**

* **Level Name: The name of the educational stage (e.g., “Primary School” or “Secondary Education”).**
* **Description: A brief overview of what this level covers or its purpose.**
* **Minimum Age: The minimum recommended age for this educational level (e.g., 6 for primary).**
* **Maximum Age: The maximum recommended age (e.g., 10 for primary).**
* **Display Order: A numerical value to determine the order in which levels are shown in the interface.**

**The form is designed to be straightforward and user-friendly, enabling quick and efficient data entry.**

**✅ Example: Level Creation Form Entry**

| **Field** | **Example Entry** |
| --- | --- |
| **Level Name** | **Primary School** |
| **Description** | **This level focuses on foundational skills in reading, writing, and basic arithmetic.** |
| **Minimum Age** | **6** |
| **Maximum Age** | **10** |
| **Display Order** | **1** |

****

#### **✏️ Edit Level Page**

#### **The Edit Level Page provides administrators with a convenient interface to update and manage the details of an existing educational level. This ensures that the platform’s learning structure remains accurate, relevant, and aligned with evolving educational standards.**

#### **The form in this page is pre-filled with the current data of the level, allowing administrators to:**

#### **Level Name: Update the name of the educational stage if changes are required.**

#### **Description: Modify the overview or purpose of the level.**

#### **Minimum Age: Adjust the minimum recommended age for this level.**

#### **Maximum Age: Update the maximum age recommendation.**

#### **Display Order: Change the order in which levels are displayed in the interface.**

#### **At the bottom of the page, two main actions are available:**

#### **Cancel: Cancel the editing process and return to the previous page without saving changes.**

#### **Update Level: Save the updated level information to the database.**

#### **This page ensures that administrators have full control over the platform’s educational framework, maintaining its effectiveness and alignment with student needs.**

#### 

#### **✅ Example: Editing a Level**

#### **Before editing:**

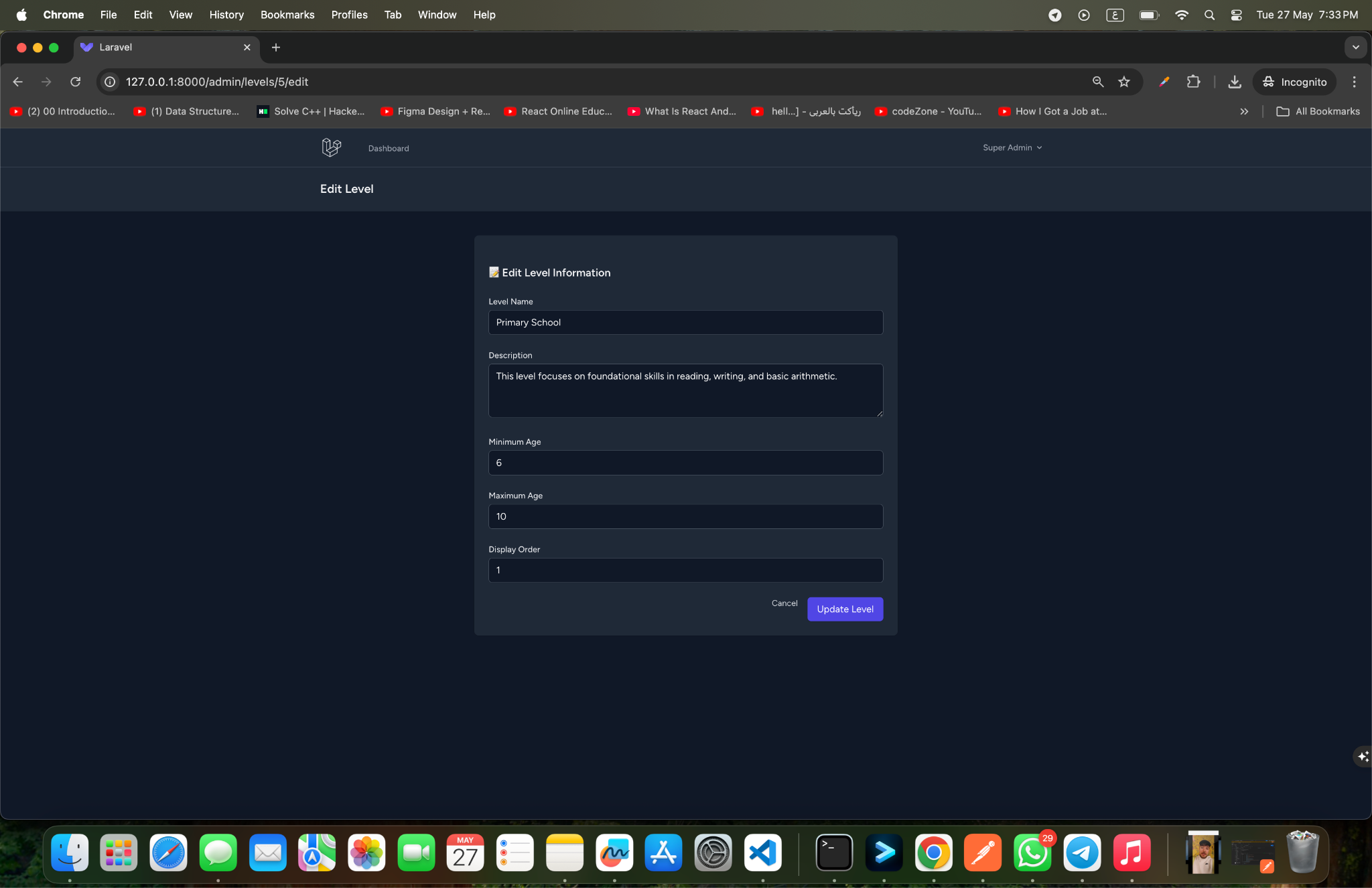
#### **Level Name: *Primary School***

#### **Description: *Basic foundational education for children aged 6-10.***

#### **After editing:**

#### **Level Name: *Primary Education***

#### **Description: *Updated to include a broader curriculum and updated teaching methods.***



#### 

#### 

#### **📄 Individual Level Page**

#### **The Individual Level Page provides a detailed overview of a single educational level within the platform. This page helps administrators review all essential details and track how the level integrates with other educational content.**

#### **Key features displayed on this page include:**

#### **Level Name: Clearly shows the name of the level (e.g., “Primary School” or “Secondary Education”).**

#### **Description: A comprehensive summary of what this level covers and its importance within the platform.**

#### **Minimum Age: The minimum recommended age for students within this level.**

#### **Maximum Age: The maximum age for this educational stage.**

#### **Display Order: The position of this level in the listing of available levels.**

#### **Additional insights may be displayed, such as:**

#### **Number of Courses Linked: The total number of courses assigned to this level.**

#### **Associated Content: Direct links to the courses or materials that belong to this level.**

#### **This page provides administrators with a clear snapshot of each level’s purpose and its role in the broader educational structure.**

#### 

#### **✅ Example: Individual Level Page Details**

| **Field** | **Example Entry** |
| --- | --- |
| **Level Name** | **Primary School** |
| **Description** | **Focuses on fundamental skills and foundational knowledge for children aged 6-10.** |
| **Minimum Age** | **6** |
| **Maximum Age** | **10** |
| **Display Order** | **1** |
| **Number of Courses** | **5 courses linked** |

#### 

#### 

#### 

#### **🗂️ Level Listing Page**

**This page presents an organized and comprehensive list of all educational levels within the platform. It allows administrators to manage and monitor the different levels, ensuring that the educational content is properly structured and up-to-date.**

**The Level Listing Table includes the following information:**

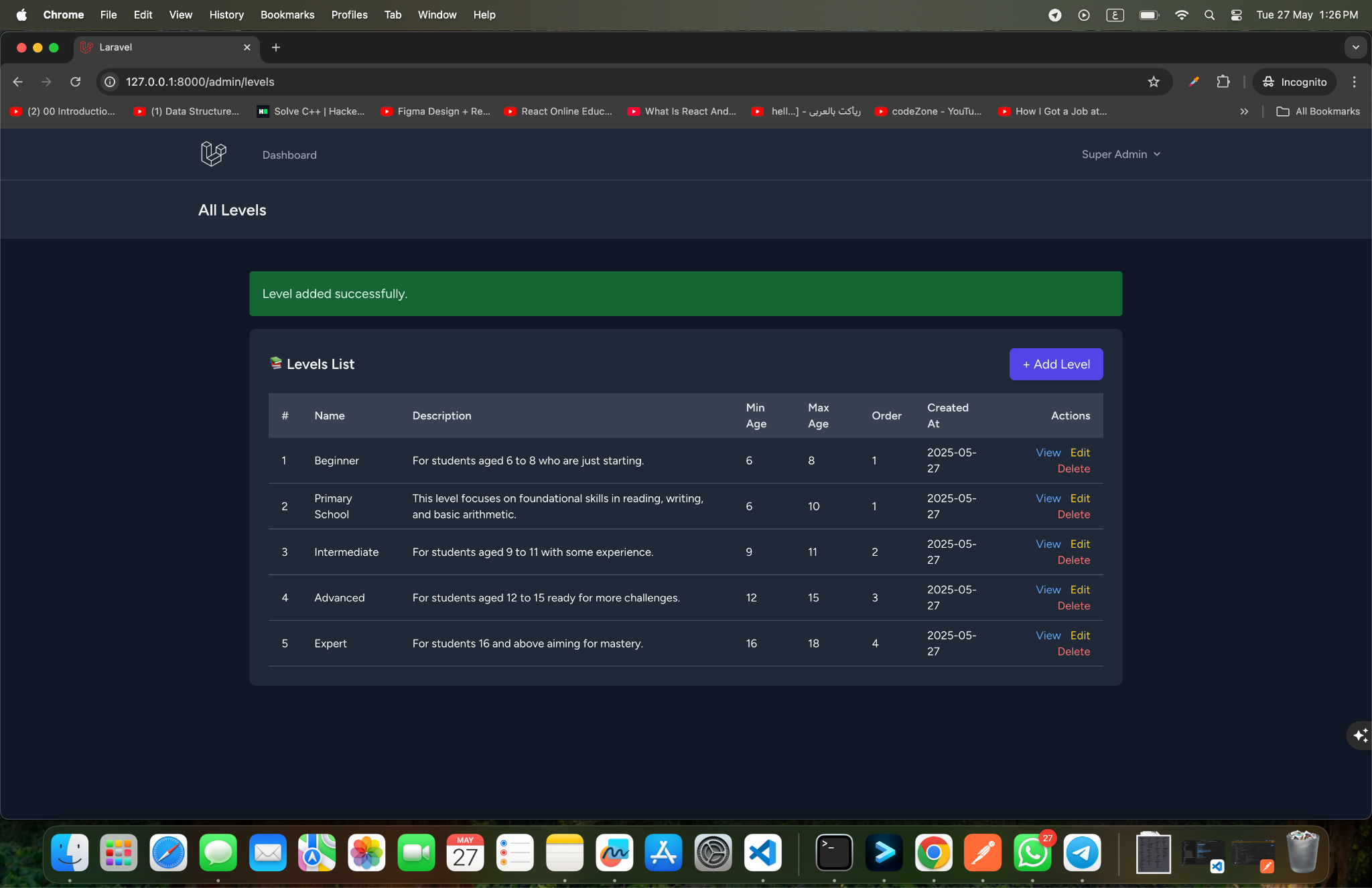
* **Level Name: The name of the educational stage (e.g., “Primary School,” “Secondary Education”).**
* **Description: A brief overview of what this level covers.**
* **Minimum Age: The minimum recommended age for this level.**
* **Maximum Age: The maximum recommended age for this level.**
* **Display Order: A numerical value indicating the order in which levels appear.**
* **Actions: Buttons or links to edit or delete the level as needed.**

**This table view provides an intuitive and efficient way for administrators to:**

* **View and compare all available levels.**
* **Quickly identify and access levels that require updates or adjustments.**
* **Maintain the accuracy and relevance of the educational structure.**

**✅ Example Table (Level Listing)**

| **Level Name** | **Description** | **Min Age** | **Max Age** | **Order** | **Actions** |
| --- | --- | --- | --- | --- | --- |
| **Primary School** | **Focuses on foundational skills.** | **6** | **10** | **1** | **Edit/Delete** |
| **Preparatory** | **Prepares students for advanced topics.** | **11** | **14** | **2** | **Edit/Delete** |
| **Secondary** | **Specializes in subject-focused learning.** | **15** | **18** | **3** | **Edit/Delete** |

****

#### **🎓 Course Creation Page**

**This page allows administrators to create and manage new courses within the platform. Each course is linked to a specific educational level and includes various materials to enhance the learning experience. The Course Creation Form is organized into intuitive sections for efficient data entry and management.**

**The form is divided into two main sections:**

##### **📘 Course Information**

**This section gathers basic details about the course:**

* **Course Title: The name of the course (e.g., “Introduction to Algebra”).**
* **Level: The educational level this course belongs to. Administrators can select the appropriate level from a dropdown list (e.g., Primary, Secondary).**
* **Description: A brief overview of the course, explaining its objectives and scope.**

##### **📂 Course Materials**

**This section allows administrators to enrich the course with supporting materials and resources:**

* **Upload Images: Allows the upload of images (e.g., course cover image or visual aids).**
* **Upload Videos: Allows the upload of video resources that complement the course content.**
* **Useful Links: Field for adding external resources or reference materials, entered as comma-separated URLs (e.g., “**[**https://example.com**](https://example.com)**, https://youtube.com...”).**
* **Summary Text: A short textual summary or highlight for the course to help students quickly understand the course focus.**

##### **✅ Form Actions**

* **Cancel: A button to discard the entered data and return to the previous page.**
* **Save Course: A button to submit the form and create the new course profile.**

**✅ Example: Course Creation Form Entry**

| **Field** | **Example Entry** |
| --- | --- |
| **Course Title** | **Introduction to Algebra** |
| **Level** | **Primary School** |
| **Description** | **This course covers foundational algebra concepts, including variables, equations, and basic problem solving.** |
| **Upload Images** | **algebra\_intro.jpg** |
| **Upload Videos** | **algebra\_lesson1.mp4** |
| **Useful Links** | [**https://example.com**](https://example.com)**,** [**https://khanacademy.org**](https://khanacademy.org) |
| **Summary Text** | **A comprehensive introduction to algebra for primary students.** |

#### **🗂️ Course Listing Page**

**This page displays a comprehensive list of all courses within the platform. The table view includes essential course details and provides administrators with quick access to manage each course.**

**The Course Listing Table includes:**

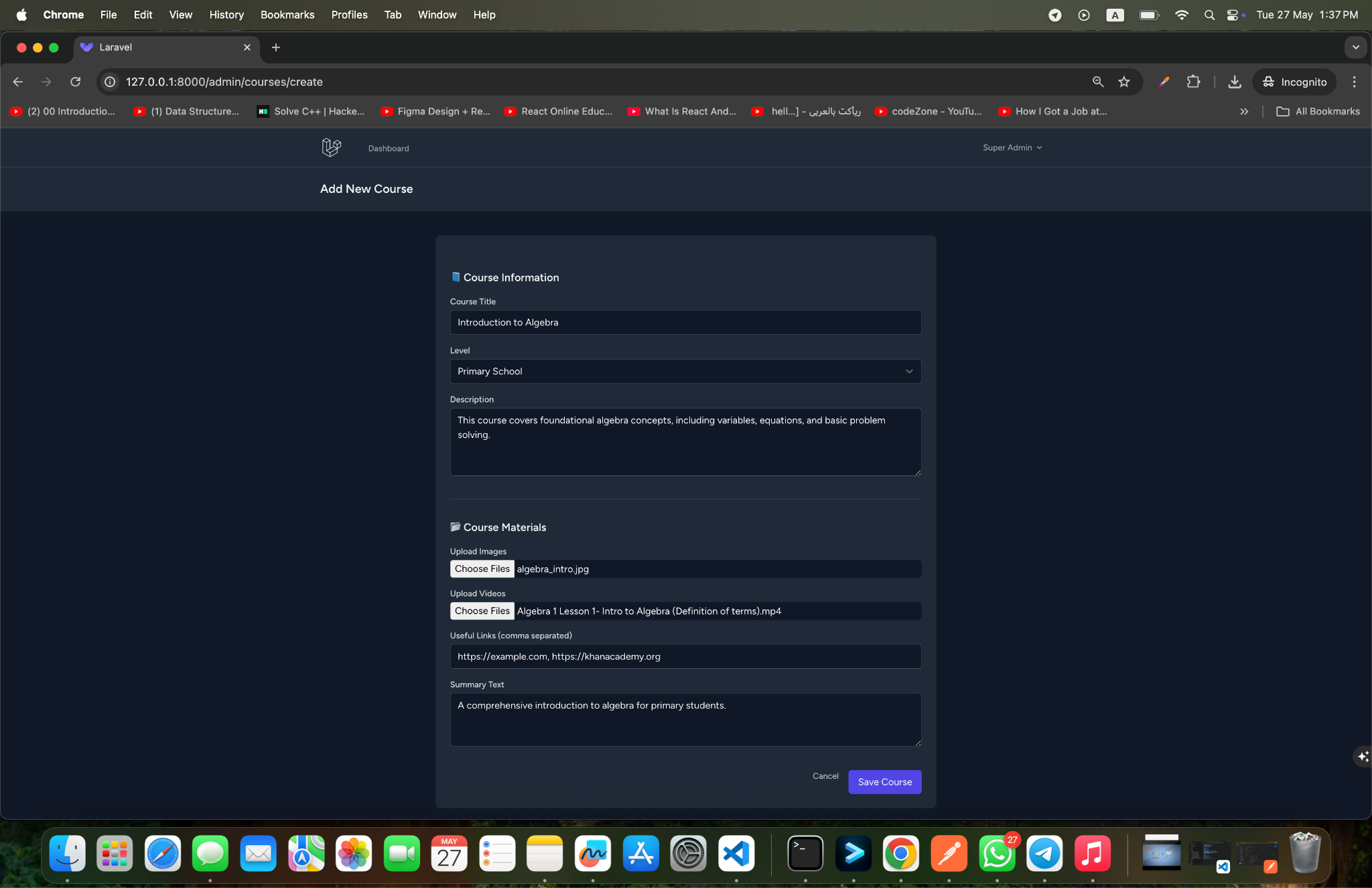
* **Title: The name of the course.**
* **Description: Brief course summary.**
* **Level: The educational level the course belongs to.**
* **Image: Visual preview of the course (if uploaded).**
* **PDF: Link to any associated PDF resources.**
* **Actions: Buttons or links to edit or delete the course.**

**This page allows administrators to efficiently:**

* **View all existing courses.**
* **Edit or update course information as needed.**
* **Maintain the quality and accuracy of the available courses.**

**✅ Example Table (Course Listing)**

| **Title** | **Description** | **Level** | **Image** | **PDF** | **Actions** |
| --- | --- | --- | --- | --- | --- |
| **Introduction to Mathematics** | **Basic concepts in algebra and geometry.** | **Primary School** | **Available** | **math\_intro.pdf** | **Edit / Delete** |
| **English Language Basics** | **Foundational English skills.** | **Primary School** | **Available** | **english\_basics.pdf** | **Edit / Delete** |

****

#### **🗂️ Course Listing Page**

**This page displays an organized and comprehensive list of all courses available on the educational platform. It allows administrators to manage and monitor courses efficiently, ensuring that educational content remains well-structured and up-to-date.**

**The Course Listing Table includes the following columns:**

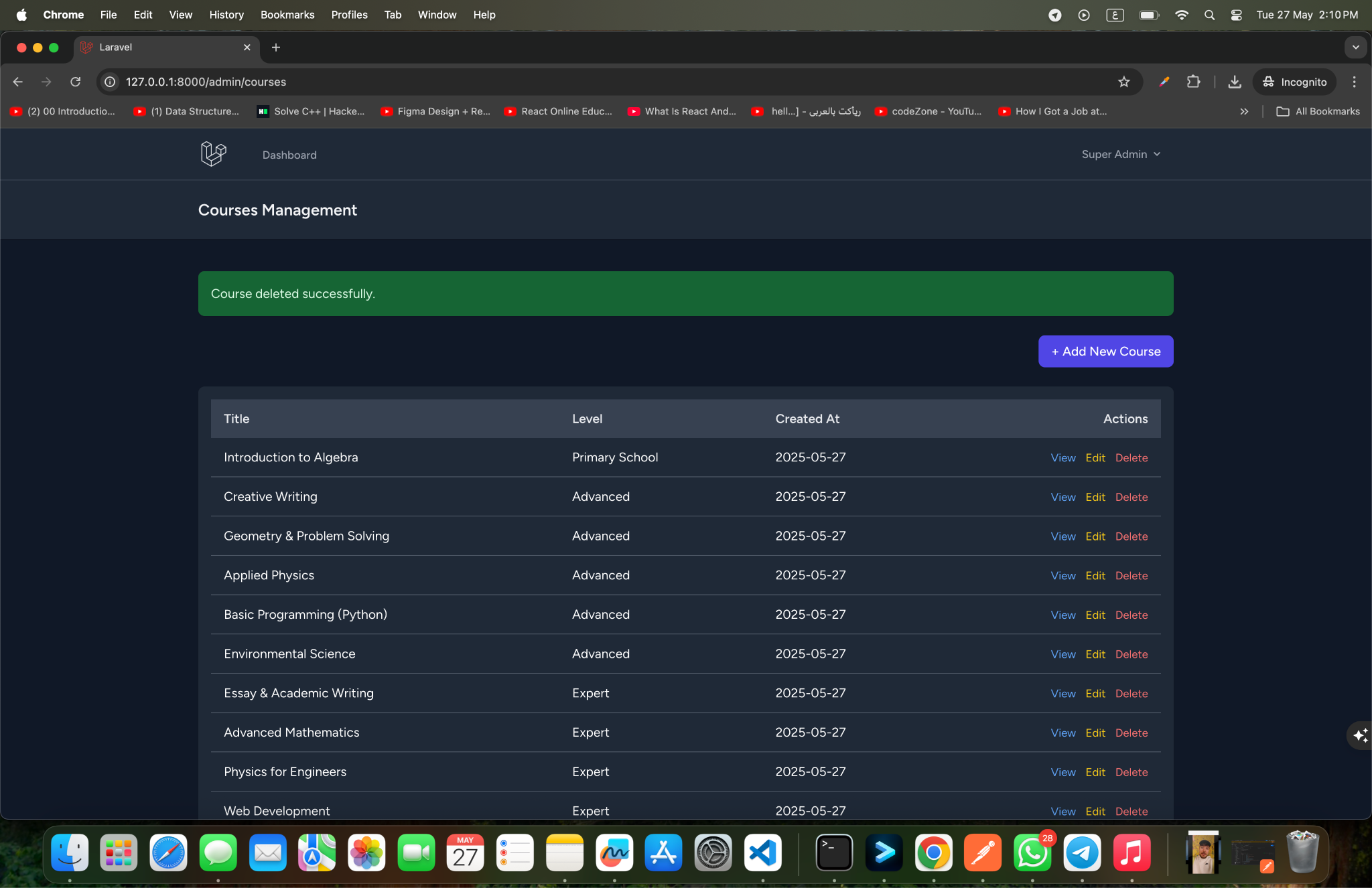
* **Course Title: The name of the course (e.g., “Introduction to Algebra”).**
* **Level: The educational level this course belongs to (e.g., Primary, Secondary).**
* **Description: A brief summary of what the course covers.**
* **Images: A preview of uploaded images associated with the course (if any).**
* **Videos: Links or references to video resources (if any).**
* **Useful Links: External resources or supporting materials linked to the course.**
* **Summary Text: A short highlight or overview of the course.**
* **Actions: Buttons or links to edit or delete the course.**

**This page provides a clear overview of the available courses, enabling administrators to:**

* **Quickly access and manage course details.**
* **Ensure that each course is properly categorized and enriched with relevant materials.**
* **Keep the platform’s educational content accurate, engaging, and up-to-date.**

**✅ Example Table (Course Listing)**

| **Course Title** | **Level** | **Description** | **Images** | **Videos** | **Useful Links** | **Summary** | **Actions** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Introduction to Algebra** | **Primary School** | **Basic algebra concepts and problem-solving techniques.** | **Available** | **Available** | [**https://example.com**](https://example.com) | **Foundational course in algebra.** | **Edit / Delete** |
| **English Language Basics** | **Primary School** | **Foundational skills in reading and writing English.** | **Available** | **Available** | [**https://english-learning.org**](https://english-learning.org) | **Building core English skills.** | **Edit / Delete** |

****

#### **✏️ Edit Course Page**

**The Edit Course Page allows administrators to update the information of an existing course. This functionality ensures that course content can be maintained, expanded, or corrected as needed over time. The interface mirrors the course creation form but is pre-filled with the course’s current data for ease of modification.**

**Administrators can adjust the following fields:**

* **Course Title: Update the course name if it changes or needs refinement.**
* **Level: Modify the associated educational level if the course is reassigned.**
* **Description: Change the course’s overview to better reflect updated content or objectives.**
* **Upload Images: Replace or add new image resources (e.g., updated cover image).**
* **Upload Videos: Update video files linked to the course.**
* **Useful Links: Edit or add external resources (comma-separated).**
* **Summary Text: Revise the brief course summary as needed.**

**At the bottom of the form, two main actions are provided:**

* **Cancel: Discard any changes and return to the previous page.**
* **Update Course: Save the updated course information to the system.**

**This page ensures flexibility and control over course content, allowing the platform to evolve and stay aligned with current educational standards and needs.**

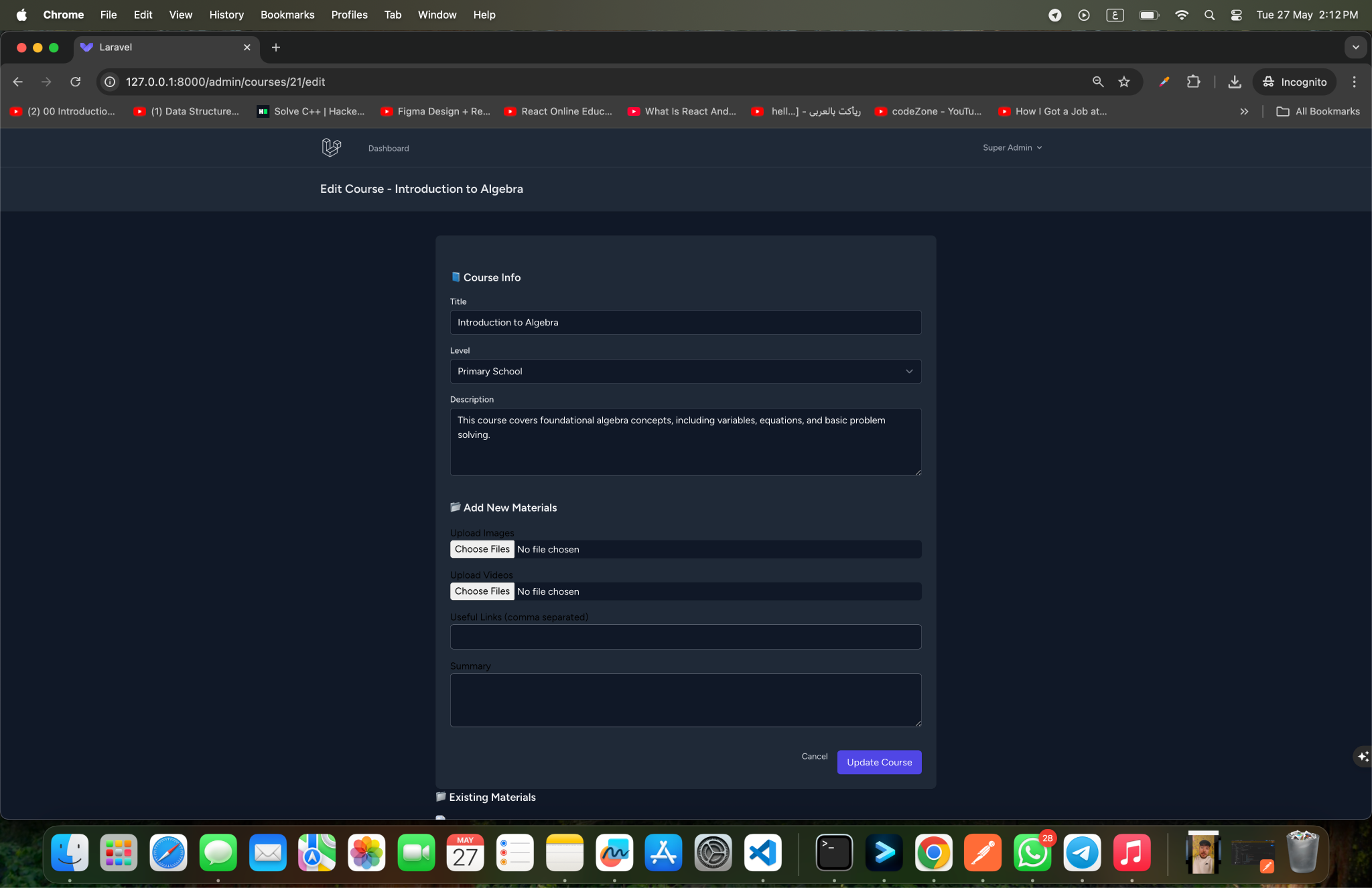
**✅ Example: Editing a Course**

**Before editing:**

* **Course Title: *Introduction to Algebra***
* **Description: *Basic algebra concepts***

**After editing:**

* **Course Title: *Algebra 101***
* **Description: *Expanded course with additional exercises and problem-solving strategies***

****

### 

### **Figure 3: Student Dashboard and Management Screenshots**

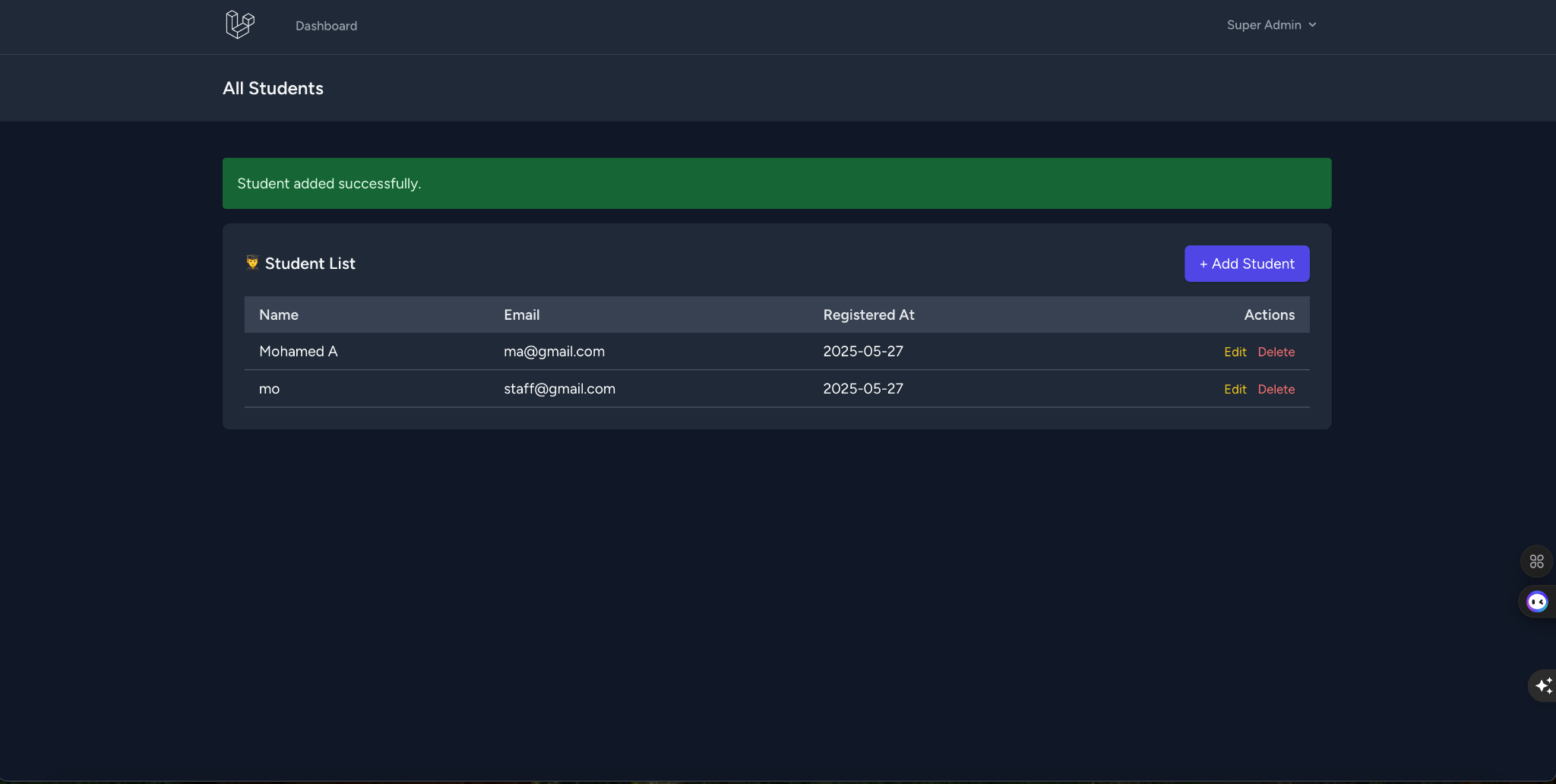
### 

### 

| **Description** | **Details** |
| --- | --- |
| **Overview** | **This set of screenshots demonstrates the student management features of the educational website platform.** |
| **Key Screens Highlighted** | **- Student Creation Page: This interface allows administrators to create new student profiles by entering details such as name, age, and level.- Student Listing Page: A table displaying all registered students, their details, and options to edit or delete their profiles.- Individual Student Page: A detailed view of a single student’s profile, including progress tracking and assigned courses.** |
| **Purpose** | **To showcase the administrative control and flexibility in managing student data within the platform.** |
| **Features Demonstrated** | **- User-friendly data entry for new students.- Quick access to student information and management controls.- Clean, organized interface to enhance usability.** |

### **Screenshots :**

### 



### 

#### **🔒 Form Validation in Student Registration**

#### **To ensure that the data entered by students during registration is valid and secure, the platform implements robust form validation using Laravel’s validation features. The system checks each field based on the following rules:**

#### **Name: Must be provided, consist of text, and not exceed 255 characters.**

#### **Email: Required, properly formatted, lowercase, and unique within the users table.**

#### **Password: Required, must match the confirmed password field, and follow the default secure password rules in Laravel (including minimum length and complexity).**

#### **Birth Date: Required, must be a valid date, and the system enforces a reasonable age range by ensuring:**

#### **Minimum Age: 3 years old (birth date must be before or equal to today minus 3 years).**

#### **Maximum Age: 100 years old (birth date must be after or equal to today minus 100 years).**

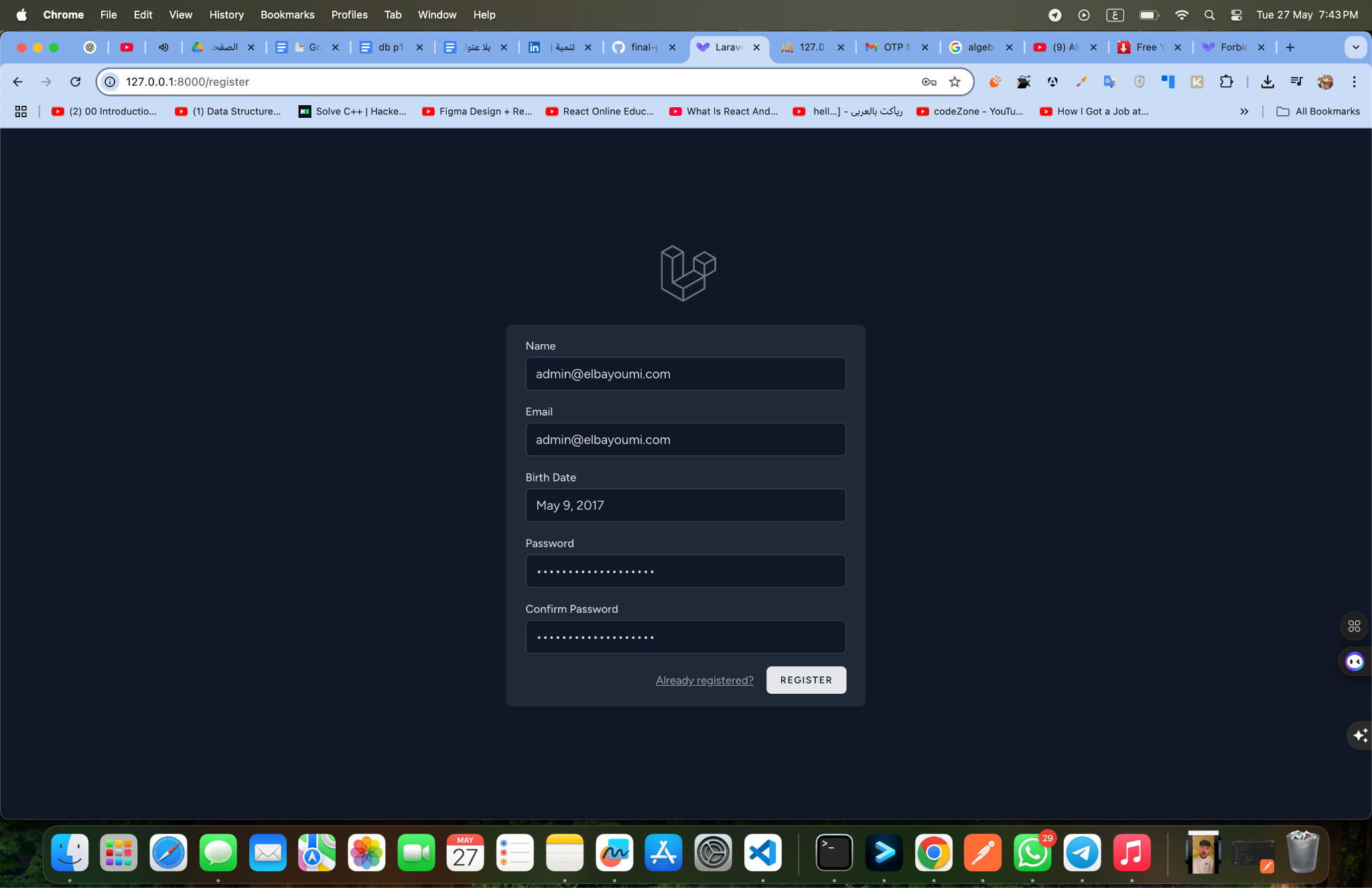
#### 

#### **✅ Age Calculation & Level Assignment After validating the birth date, the system calculates the student’s age using the Carbon library. It then determines the appropriate educational level by comparing the age to the level’s minimum and maximum age ranges.**

#### **✅ Example of Validation Logic**

| **Field** | **Validation Rules** |
| --- | --- |
| **Name** | **required, string, max:255** |
| **Email** | **required, string, lowercase, email format, unique in users table** |
| **Password** | **required, confirmed, secure password standards (Laravel’s default)** |
| **Birth Date** | **required, valid date, between 3 and 100 years old** |

#### **This validation ensures that only accurate and consistent data is stored in the database, improving the platform’s reliability and security.**



#### 

#### 

#### **👤 Student Login Page**

### **The Student Login Page provides a secure and intuitive interface for students to access the educational platform. This page requires students to enter their credentials to authenticate and gain access to their personalized dashboard.**

### **Key Elements:**

### **Email Address: The student’s unique email used for login.**

### **Password: A secure password field to ensure account protection.**

### **Login Button: A button to submit credentials and log in to the system.**

### **Forgot Password Link (if implemented): Allows students to reset their password if needed.**

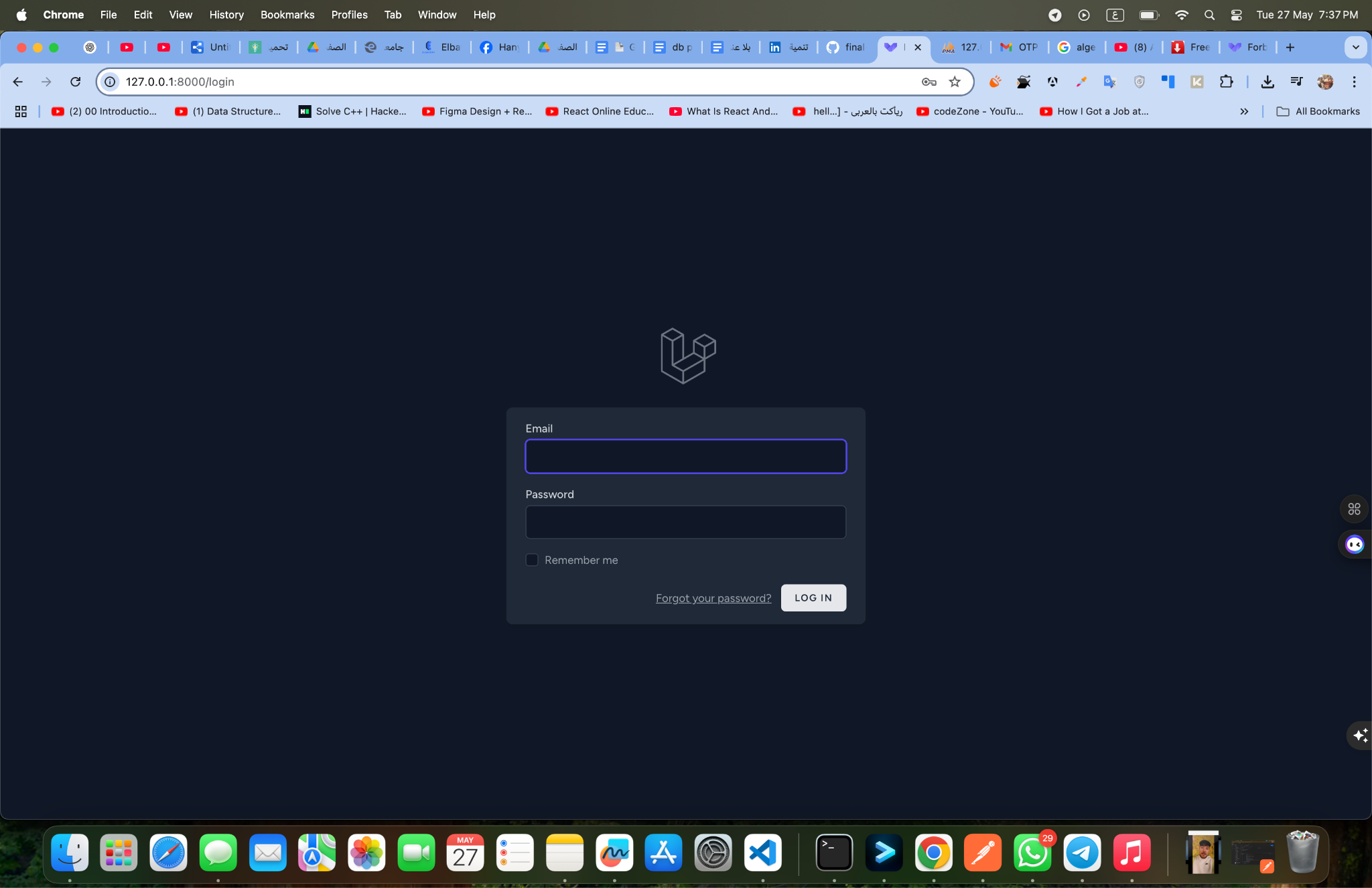
### **The login form is designed with simplicity and accessibility in mind, ensuring students of different technical backgrounds can easily log in.**

### 

### **✅ Example: Student Login Page Layout**

| **Field** | **Description** |
| --- | --- |
| **Email** | **Student email address (e.g., student@example.com).** |
| **Password** | **Password field for secure access.** |
| **Login Button** | **Button to authenticate and access the dashboard.** |

### 



### 

### 

### 

### 

#### **🏠 Student Dashboard Home Page**

### **The Student Dashboard Home Page serves as the central hub for students once they log in to the platform. It provides a clear and intuitive interface, allowing them to easily navigate and access their personalized educational content.**

### **Key features and elements displayed on this page:**

### **Personalized Greeting & Overview: The dashboard typically greets the student and displays their assigned educational level along with their age (e.g., “Courses for Level: Primary School (Age: 8)”), helping them stay oriented within the system.**

### **Profile Dropdown Menu: A profile section is available at the top right, providing quick access to the following actions:**

### **Profile: View or edit personal information.**

### **Age & Level: View the student’s age and educational level for easy reference.**

### **Logout: Securely sign out of the platform.**

### **Courses Display: The main area of the dashboard presents courses relevant to the student’s educational level. For example, students in the Primary School level will see a course card like:**

### **Course Title: Introduction to Algebra**

### **Short Description: A brief summary of the course’s goals and topics (e.g., “This course covers foundational algebra concepts, including variables, equations, and basic problem-solving.”).**

### **View Course Button: A call-to-action button to navigate to the full course content and lessons.**

### **The dashboard is designed with simplicity and clarity in mind, ensuring that students of all technical levels can easily navigate the platform and start learning right away.**

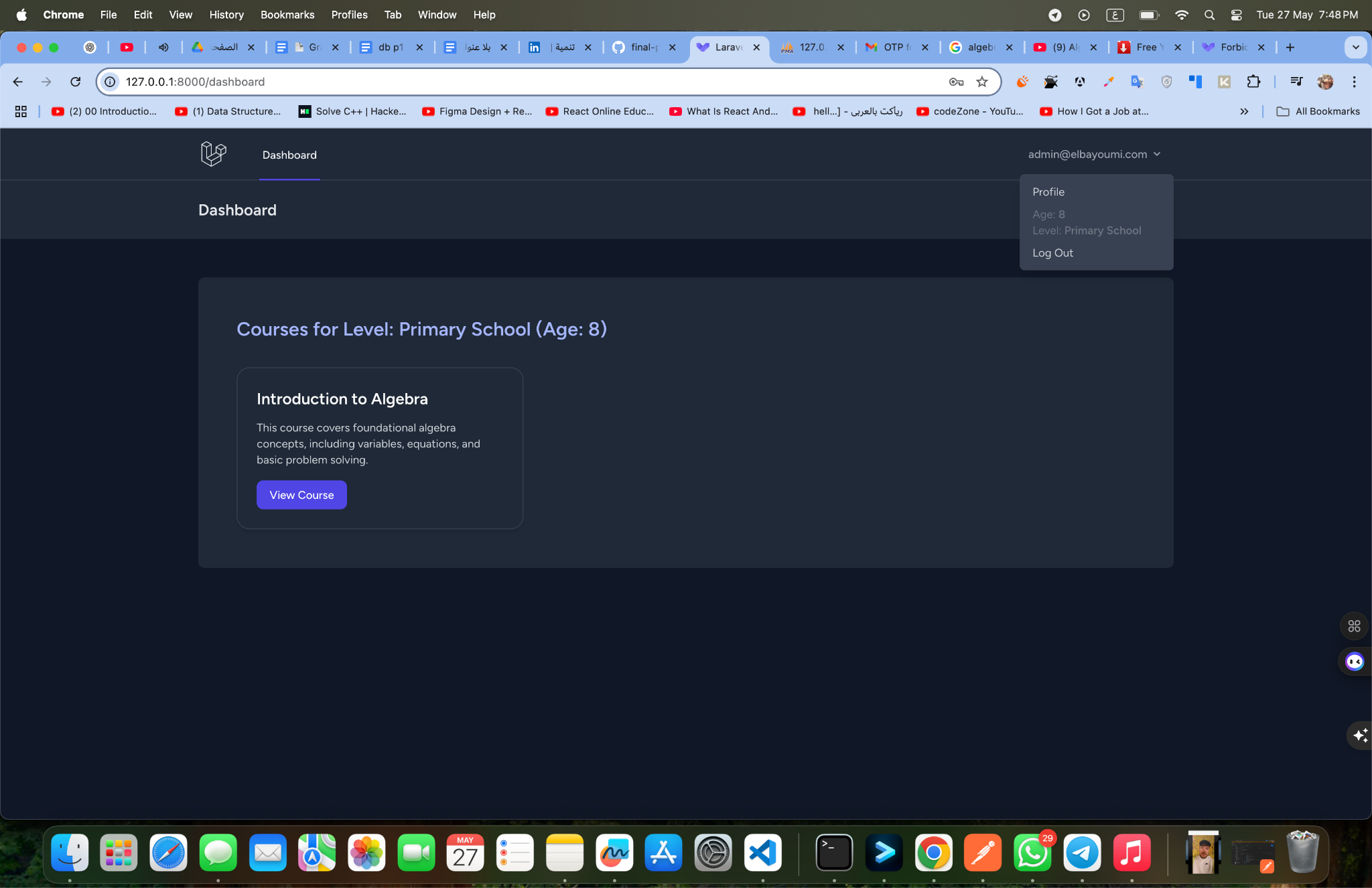
### 

### **✅ Example: Student Dashboard Overview**

| **Element** | **Description** |
| --- | --- |
| **Profile Menu** | **Shows email, age, level, and a logout option.** |
| **Level & Age Display** | **Indicates the student’s current educational level and their age.** |
| **Course Cards** | **Each course card contains a title, short description, and a link to view the full course.** |
| **Navigation Simplicity** | **User-friendly interface that prioritizes access to the most relevant content for the student.** |

### 

### 



#### **👤 Student Profile Page**

The **Student Profile Page** provides students with a user-friendly interface to view and manage their personal information. This page is designed to enhance user autonomy and ensure data accuracy while prioritizing security.

Key sections and features on this page include:

##### **🟩 Profile Information**

This section displays the student’s key details and allows them to update personal information as needed:

* **Name:** The student’s full name.
* **Email:** The email address used for login and communication.
* **Birth Date:** The student’s date of birth, which is used to calculate their age and suggest the appropriate educational level.
* **Age & Level:** Directly visible information about the student’s current age and assigned educational level (e.g., Primary School).

✅ A **Save** button is provided to confirm and save any changes made.

##### **🟩 Update Password**

To maintain account security, students can change their password by providing:

* **Current Password:** To confirm identity before making changes.
* **New Password:** The new password to be set.
* **Confirm Password:** To ensure the new password was entered correctly.

✅ A separate **Save** button is included for securely updating the password.

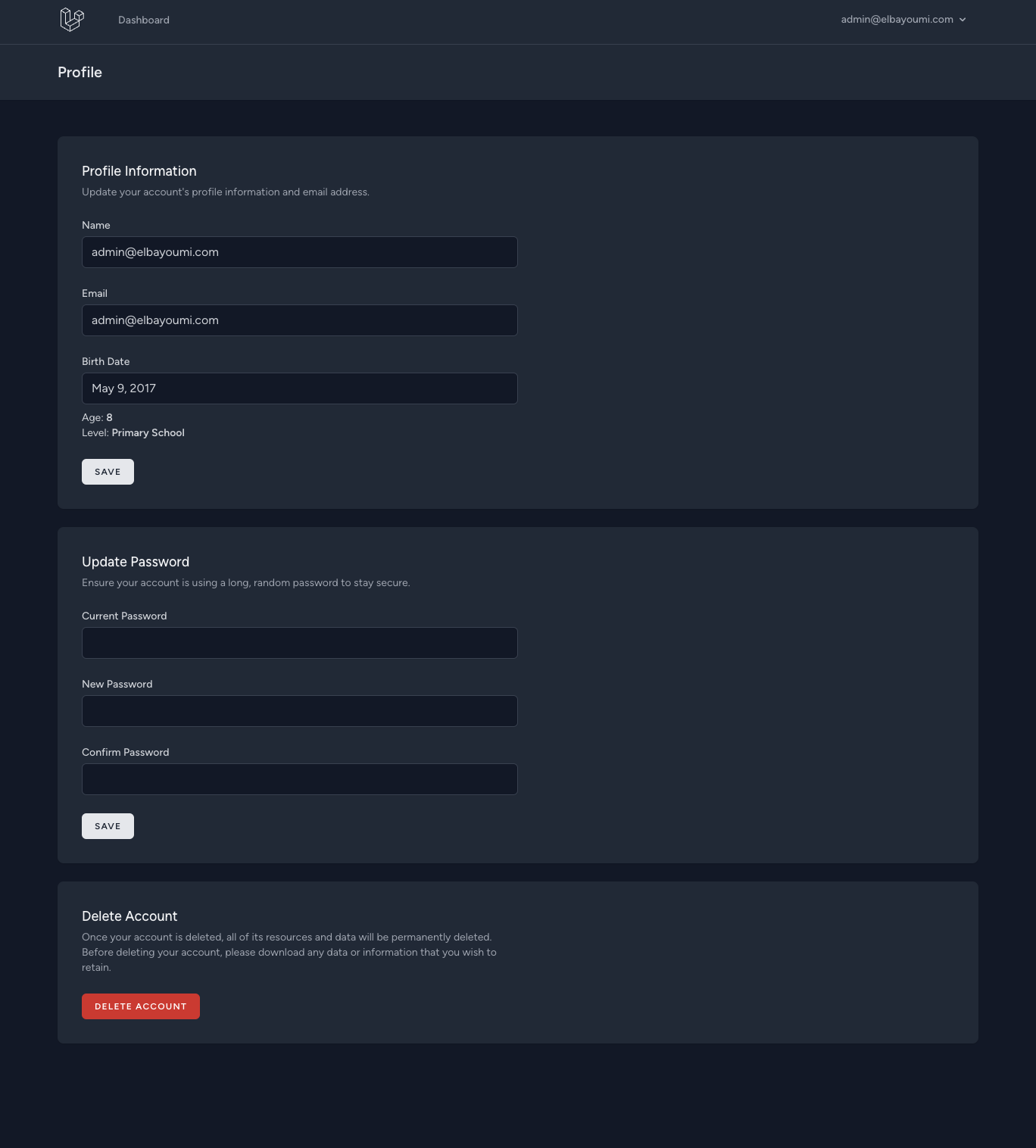
##### **🟩 Delete Account**

For privacy and control, students have the option to permanently delete their account. This section includes:

* A clear warning that **all resources and data will be permanently deleted**.
* A note to download any important data before proceeding.
* A **Delete Account** button to confirm and execute the deletion.

✅ **Example: Profile Page Overview**

| **Section** | **Description** |
| --- | --- |
| Profile Information | View and update name, email, birth date, and see age & level. |
| Update Password | Change current password securely. |
| Delete Account | Option to remove account and all associated data permanently. |



#### **📚 Course Content View Page**

The **Course Content View Page** provides a detailed presentation of the course materials and metadata for each course. It ensures that students and administrators alike can quickly understand the course’s purpose, scope, and associated resources.

Key features and elements on this page include:

* **Course Title:** The name of the course (e.g., “Introduction to Algebra”).
* **Course Description:** A brief summary of the course’s content and objectives (e.g., “This course covers foundational algebra concepts, including variables, equations, and basic problem solving.”).
* **Level:** Indicates the educational level the course is intended for (e.g., “Primary School”).
* **Created At:** Shows the date when the course was created, helping to track updates and freshness of content.
* **Back to Dashboard:** A convenient navigation link to return to the main dashboard or course listing.

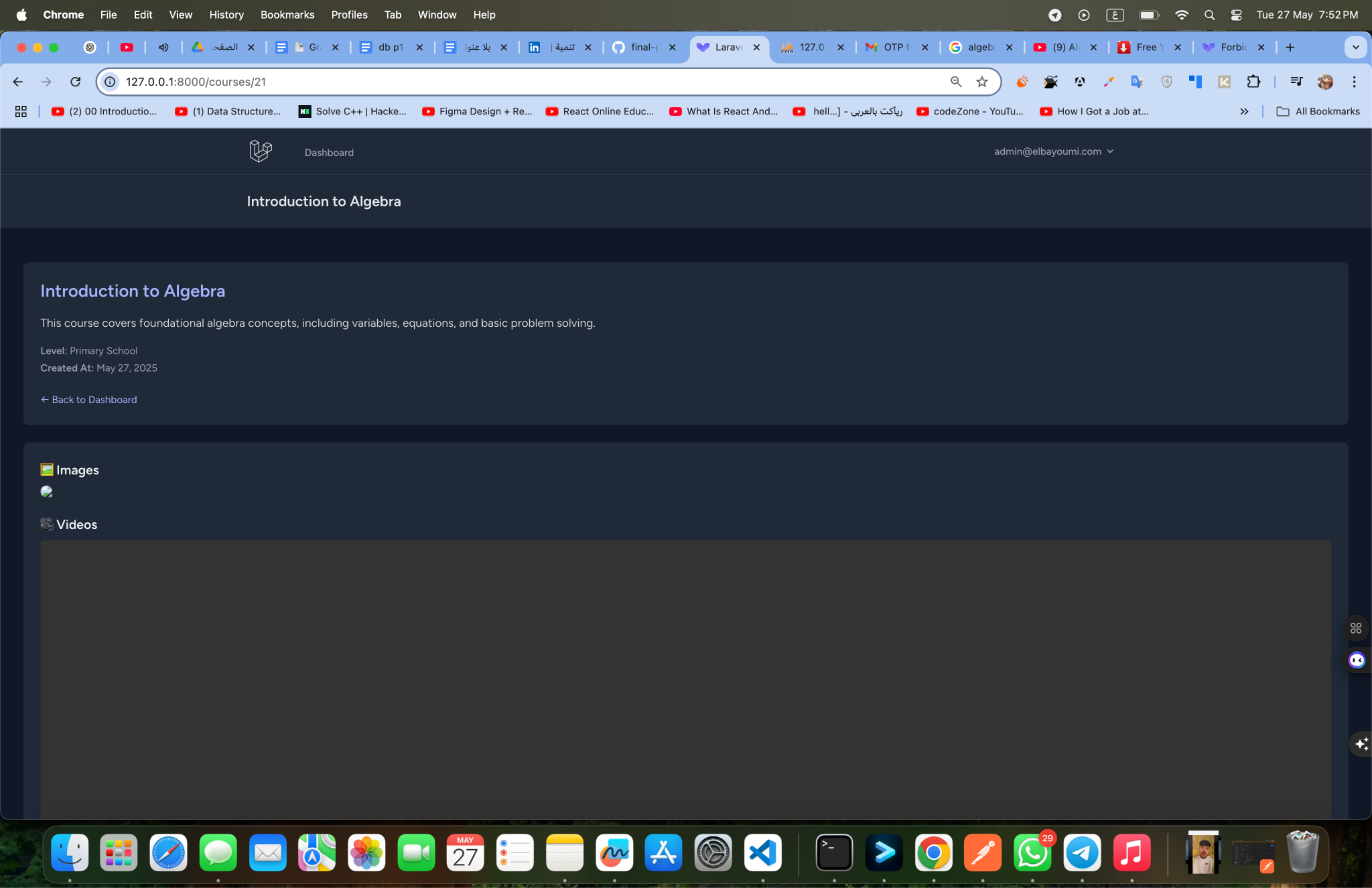
##### **📂 Course Materials**

The course includes a variety of rich educational resources, organized into different sections for clarity:

* **🖼️ Images:** Visual resources associated with the course to aid understanding.
* **🎥 Videos:** Video materials that provide interactive or visual explanations of key concepts.
* **🔗 Useful Links:** External resources and references (e.g., “https://example.com”, “https://khanacademy.org”) to expand learning opportunities beyond the platform.
* **📄 Summary:** A concise summary or highlight of the course’s focus (e.g., “A comprehensive introduction to algebra for primary students.”).

✅ **Example: Course Content View Page**

| **Field** | **Example Entry** |
| --- | --- |
| Course Title | Introduction to Algebra |
| Course Description | Foundational algebra concepts for primary students. |
| Level | Primary School |
| Created At | May 27, 2025 |
| Useful Links | <https://example.com>,<https://khanacademy.org> |
| Summary | A comprehensive introduction to algebra for primary students. |



#### **📄 Student Registration and Level Assignment Logic**

The **student registration process** in this project is handled by the RegisteredUserController class, which ensures a secure and accurate registration flow. Here’s how it works: