StudyJet

An online learning platform for Programming Courses

Case Description

Prepared By: Benjina Paudyal

IT Technician with a specialization in Programming

April 8, 2025

Advisor: Flemming Sørensen, TEC Ballerup

Case Description

Project Name: StudyJet

Project Purpose:

StudyJet is a comprehensive online learning platform designed specifically for programming education. The platform offers a seamless and interactive experience where users can register, log in, browse and purchase courses, learn through video content, save courses to their Wishlist, manage a shopping cart and make secure payments all through a user-friendly interface.

To demonstrate the course structure and learning flow, the platform initially features a selection of the creator's own YouTube videos as placeholders. These videos are meant solely for demonstration purposes and do not represent the final course materials.

User Roles and Account Creation:

1. **Student:**
   When users register on the homepage, they are automatically assigned the student role. They must confirm their email address before they can log in. After registration, they will receive a confirmation email with a link to verify their account. Once verified, they can log in, browse courses, manage their cart and Wishlist, purchase courses, and watch course videos after purchase.
   For added security, students can enable **Two-Factor Authentication (2FA)** through their account settings. They can also change their password at any time from the account settings.
   Authentication: JWT (JSON Web Tokens) will be used to ensure secure authentication. After logging in, students will receive a JWT that allows them to access protected routes and perform actions such as purchasing courses or viewing their Wishlist. The token will be included in the request headers for secure communication between the frontend and backend.
2. **Instructor:**
   Instructors cannot register themselves. Instead, anyone wishing to become an instructor can apply through the homepage by submitting their CV. The admin will

review the application and, if approved, manually add the instructor to the system. After being added, instructors will receive a confirmation email with default password and must set a new password during their first login. Once onboarded, instructors can create and edit courses, but all content must be approved by the admin before being published. Instructors can also view which students have purchased their courses.

Like students, instructors can enable **optional two-factor authentication (2FA)** for added security and can change their password via account settings.

Authentication: Like students, instructors will receive a JWT after login for secure access to their courses and other features.

3. **Admin:**

   A single admin account is automatically created during the initial database migration using the DB initializer. This admin has access to platform moderation tools, including the ability to view all registered users (students and instructors) and their basic information (such as username and email), see all the available courses on the platform along with their status (e.g. approved, rejected, pending), add instructors, and approve or reject submitted course content. Admins can also enable optional **two-factor authentication(2FA)** for added security and change their password through their account settings.

   Authentication: Admins will also authenticate using JWT, which grants them access to all moderation tools.

4. **Guest:** Guests can view available courses but must register to purchase or save them to their Wishlist. If a guest attempts to add a course to their Wishlist or cart, they will be redirected to the log in or registration page.

Payment Integration: Secure payment processing is handled through Stripe. For demonstration purposes, Stripe’s sandbox environment with test card data is used.

Default User Setup:

The system initializes with one admin, five instructors, five students, and a set of sample courses via the DB initializer. Additional users and data can be added later through the interface to showcase the platform’s full functionality.

Technology Stack:

- Backend: ASP.Net Core Web API (with ASP.NET Identity for user authentication and authorization, JWT token handling for session management).
- Frontend: Angular (TypeScript, HTML, and CSS)
- Database: SQL Database (for data storage and management)
- CSS Framework: Bootstrap CSS (for responsive, user-friendly design)
- Email Service: EmailJS (for sending CV applications through the homepage) and SMTP (for sending the confirmation link)

Additional Considerations:

- The project will use GitHub for version control, ensuring clear code management and version tracking.