**Sri Lanka Institute of Information**

**Technology**

Programming Applications and Frameworks (IT3030)

**Final Assignment Report**

GROUP ID: Y3S1\_WD\_47



**Edu\_Journey Educational Social Platform**

|  |  |
| --- | --- |
| **Student ID** | **Student Name** |
| IT22063564 | RAJAWARDHANA M B G T N |
| IT22547774 | L M N R ARIYARATHNA |
| IT22088482 | WIJESIRI A M |
| IT22143440 | SIRIWARDENA D R S |

Date: 05. 05. 2025

**Table of Contents**

[1. Introduction 3](#_Toc197286818)

[2. Functional Requirements & Non-Functional Requirements 4](#_Toc197286819)

[3. Overall Architecture Diagram 5](#_Toc197286820)

[4. REST API Architecture diagram 6](#_Toc197286821)

[5. Frontend Architecture Diagram 6](#_Toc197286822)

[6. System Functions 7](#_Toc197286823)

[7. GitHub 8](#_Toc197286824)

[8. References 8](#_Toc197286825)

# Introduction

### Edu Journey is an educational social platform that bridges academic and career development by enabling students to share resources, collaborate on learning experiences, and explore professional opportunities. Designed for public use, the system allows users to sign up via Gmail, Facebook, or dedicated accounts, with secure password reset functionality. The platform revolves around four interconnected modules: Job Management, mirroring LinkedIn’s hiring workflows, where users create, edit, or delete job posts (with image uploads), apply for roles, and enforce authentication to ensure only verified users manage or interact with listings; Course Management, offering a structured catalog for browsing, enrolling, and tracking academic progress (e.g., syllabus completion, assignment submissions), complete with search filters and withdrawal options; Notification Management, delivering real-time updates via WebSocket or SSE for post interactions (likes, comments), job application statuses, or course deadlines, with customizable read/unread toggles and bulk deletion; and Financial Management, integrating encrypted payment gateways for course fees, subscription plans, or job application charges, ensuring transaction transparency and secure storage. All data is centralized in a single database, while client requests pass through an API gateway to Spring Boot microservices (jobs, courses, notifications, payments), ensuring modular scalability. Built on React for dynamic frontend interactions and fortified with JWT authentication, role-based access, and end-to-end encryption, Edu Journey fosters a secure, collaborative ecosystem where students seamlessly transition from academic growth to career readiness.

# Functional Requirements & Non-Functional Requirements

1. Job Management - RAJWARDHANA M B G T N (IT22063564)

**Create Job Post**

Endpoint: POST /api/jobs

Request Fields: Title, description, optional image, creator ID.

Authentication: JWT token (authenticated users only).

Response: 201 Created with job ID and details.

**View Job Posts**

Endpoint: GET /api/jobs

Filters: Sorting (filterBy=recent), keyword search (search=software).

Response: 200 OK with job list and metadata (e.g., applications count).

**Apply for a Job**

Endpoint: POST /api/jobs/{jobId}/applications

Request Fields: Applicant ID, resume (file), cover letter (text).

Authentication: JWT token (applicant must be logged in).

Response: 201 Created with application ID.

**Edit/Delete Job Post**

Endpoints:

PUT /api/jobs/{jobId} (update job details).

DELETE /api/jobs/{jobId} (delete job).

Authentication: Only job owner (validated via JWT).

Response: 200 OK (success) or 403 Forbidden (unauthorized).

1. Course Management – L M N R ARIYARATHNA (IT22547774)

**Browse Courses**

Endpoint: GET /api/courses

Filters: Category (category=programming), instructor (instructor=John\_Doe).

Response: 200 OK with course details (name, description, duration, enrollment status).

**Enroll in a Course**

Endpoint: POST /api/courses/{courseId}/enroll

Request Fields: User ID, optional payment method (e.g., Stripe).

Authentication: JWT token (student must be logged in).

Response: 201 Created with enrollment confirmation.

**Track Course Progress**

Endpoint: PATCH /api/courses/{courseId}/progress

Request Fields: User ID, completed sections, assignments done.

Authentication: Student or instructor (JWT).

Response: 200 OK with updated progress percentage.

**Withdraw from a Course**

Endpoint: DELETE /api/courses/{courseId}/enroll

Request Fields: User ID.

Authentication: Student or admin (JWT).

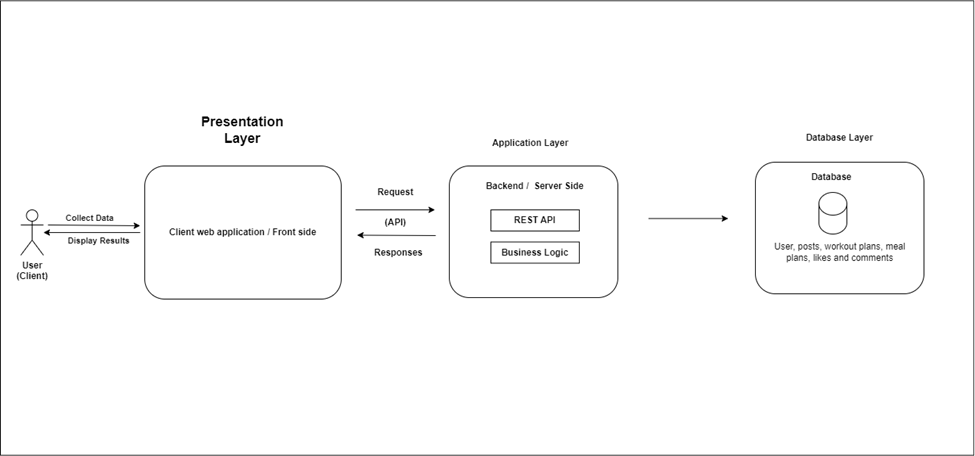
Response: 200 OK (withdrawal confirmed).

Non-Functional Requirements

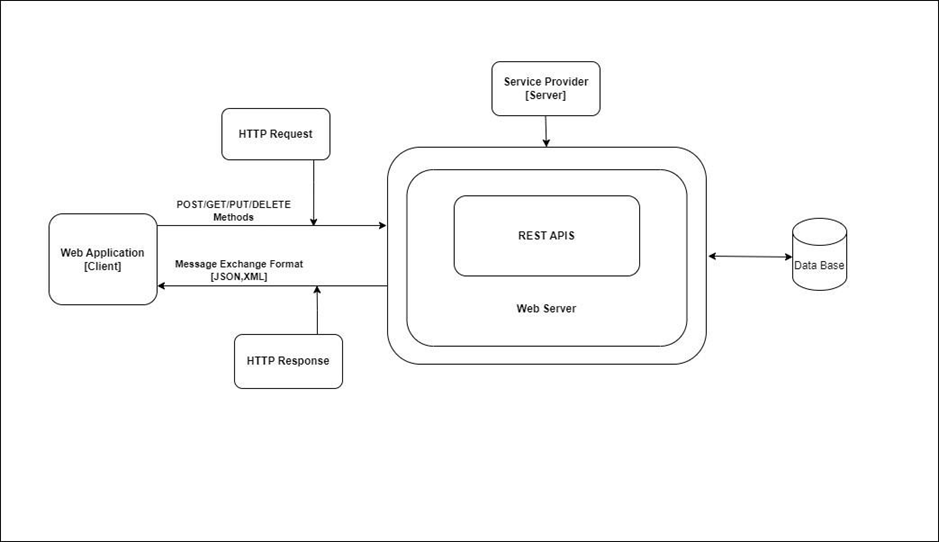
Non-Functional Requirements the REST API

* Performance – Response time, Throughput, Utilization, Static Volumetric, User interface, Conformity
* Security requirements
* Reliability
* Scalability Non-Functional Requirements the client web application
* Availability
* Maintainability
* Usability
* Accuracy
* Stability
* Accessibility

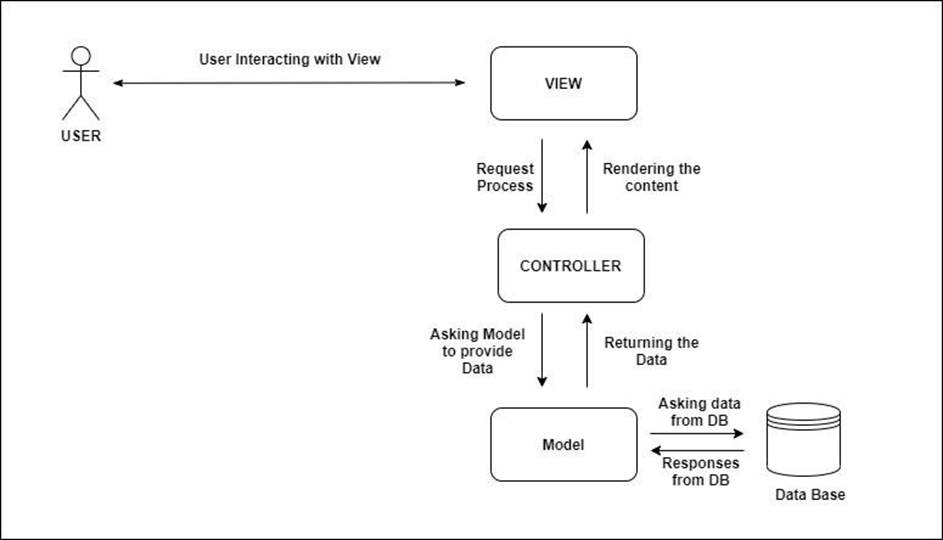
# Overall Architecture Diagram



# REST API Architecture diagram



# Frontend Architecture Diagram



# System Functions

**01. Notification Management-IT22088482(Wijesiri A.M)**

**Frontend**

The frontend of the notification management system presents a clear, seamless interface where users can:

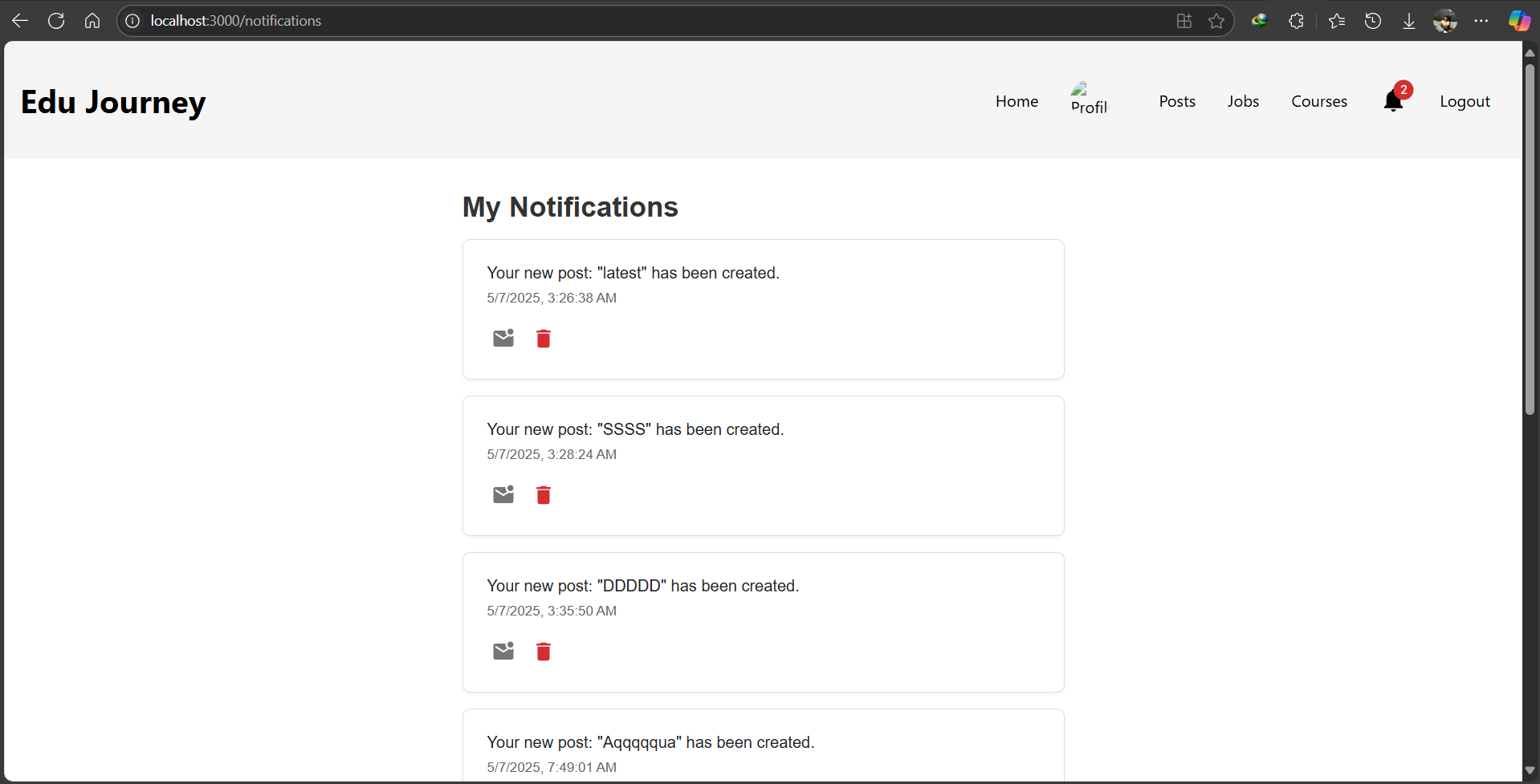
* View the list of their latest notifications.
* Retrieves all notification entries from the backend database
* See clearly formatted messages like: "Your new post: 'title' has been created."
* Identify the timestamp for each notification
* Perform quick actions with icons:

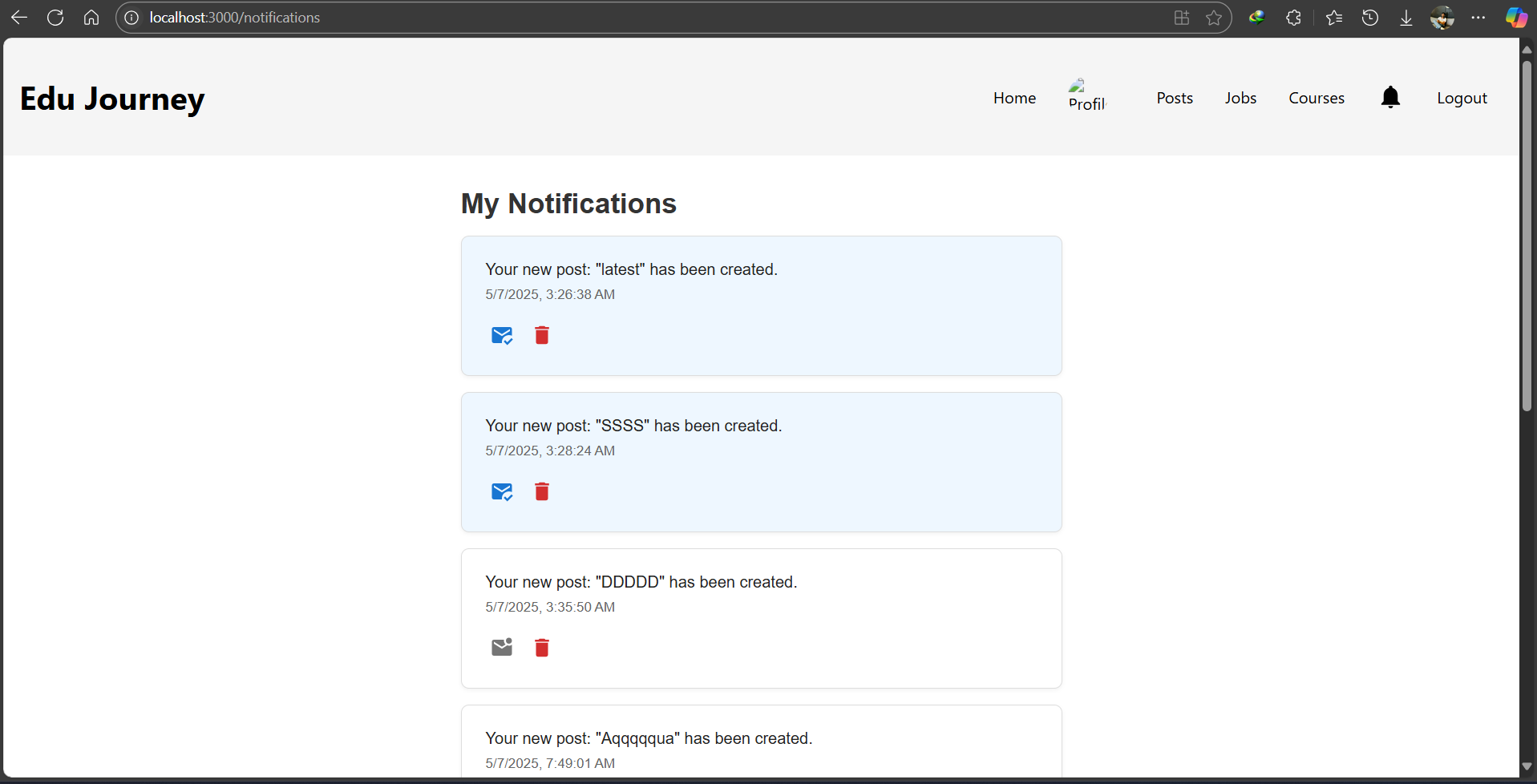
Envelope icon (possibly for marking as read or unread or opening details).

Trash icon to delete/remove the notification.

**Backend**

* Returns all notifications for the logged-in user using Principal.
* Returns the number of notifications where read == false.
* Returns a single notification only if it belongs to the user.
* Updating the read status and deletes the notification if it belongs to the current User.





**02. Post Management-IT22143440(Siriwardena D.R.S)**

**Frontend**

* Posts are displayed in a card layout and it displays in reverse chronically order.
* Each Post contains Post content such as Texts, images, or educational docs.
* Attachments with download links such as lecture slides and pdf.
* Edit icon and trash icons for the owner's posts.
* Button labeled “+ New Post” triggers a form for users to:

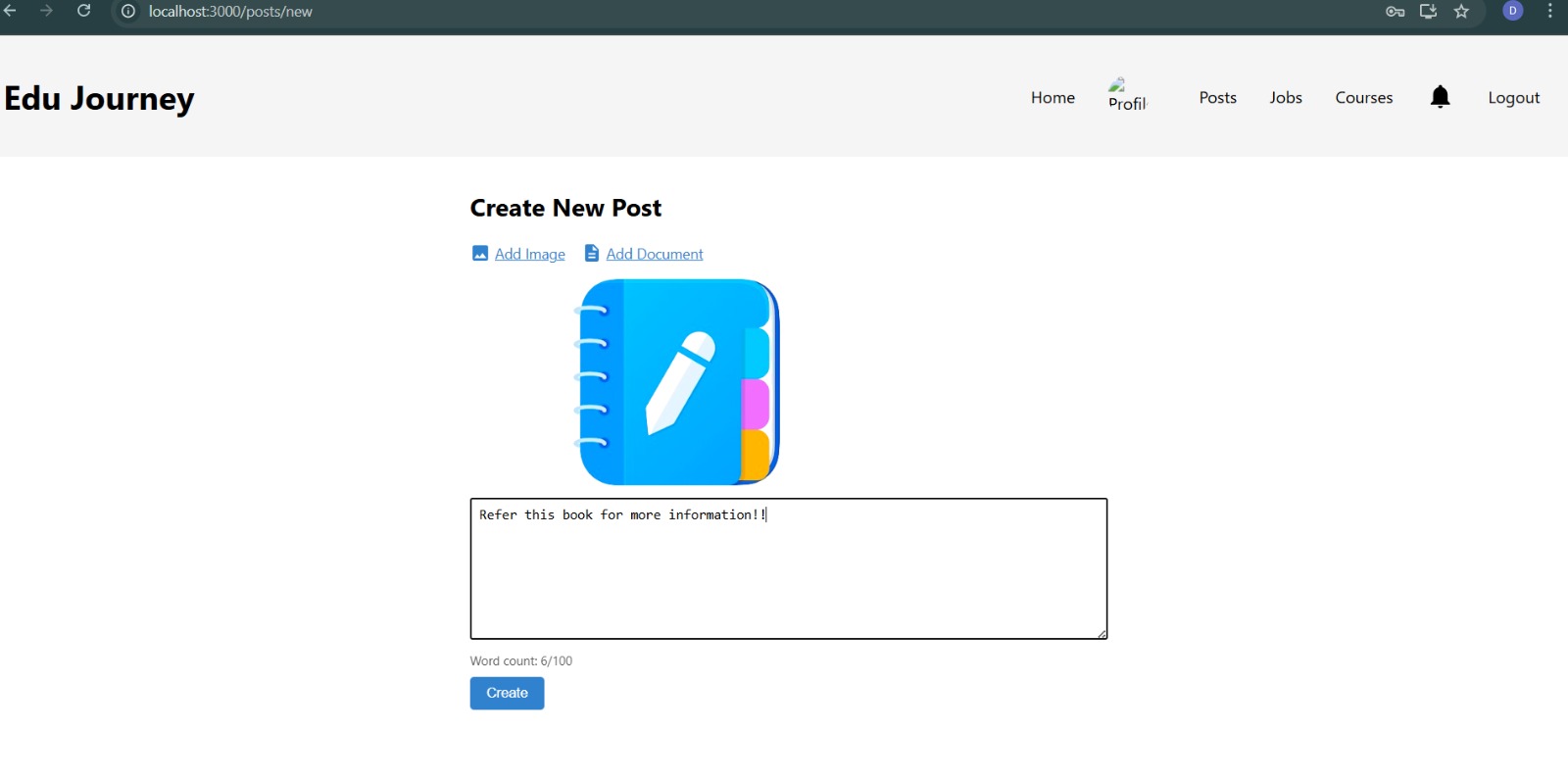
Enter text.

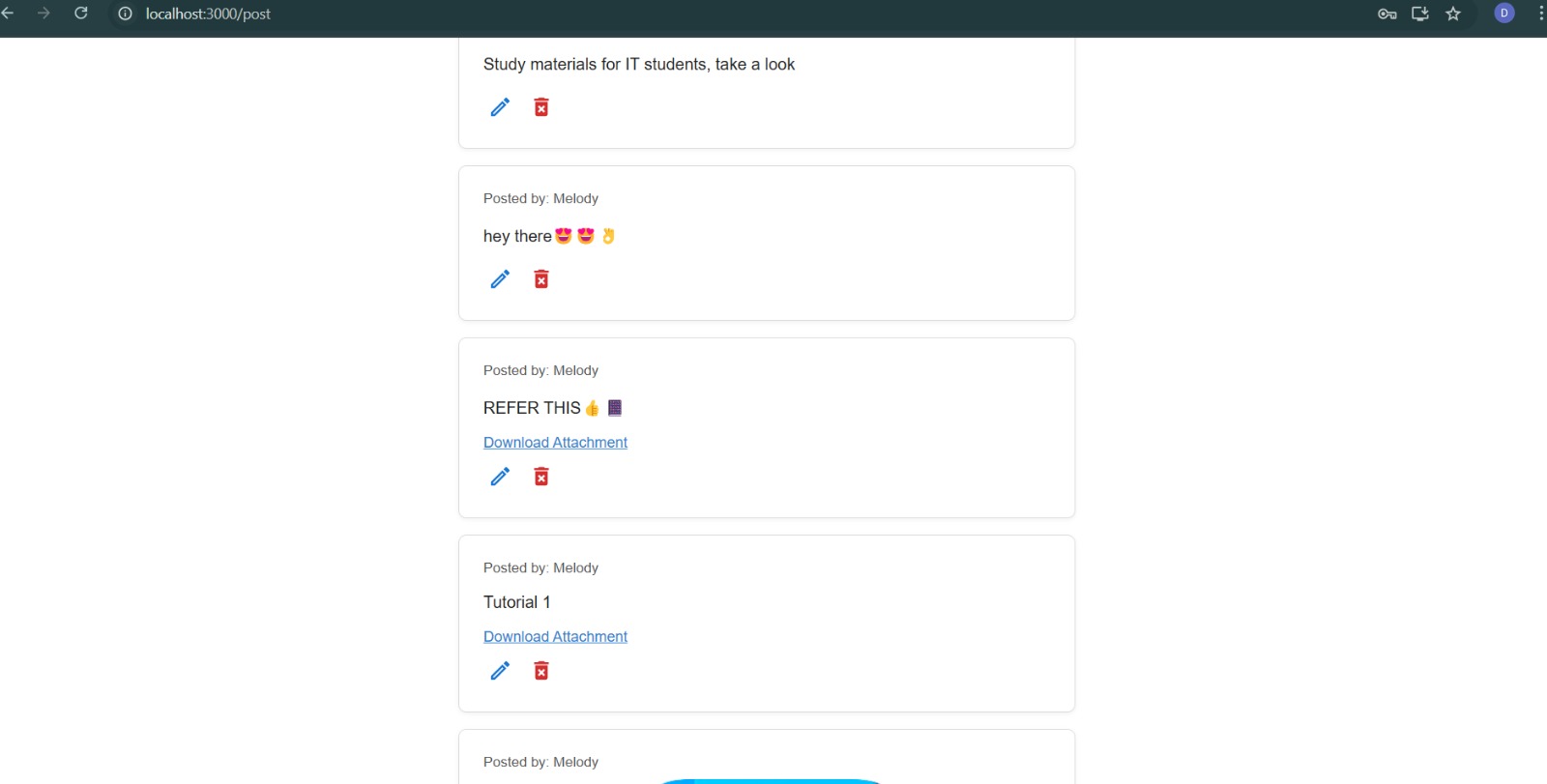
Upload attachments like (documents, images).

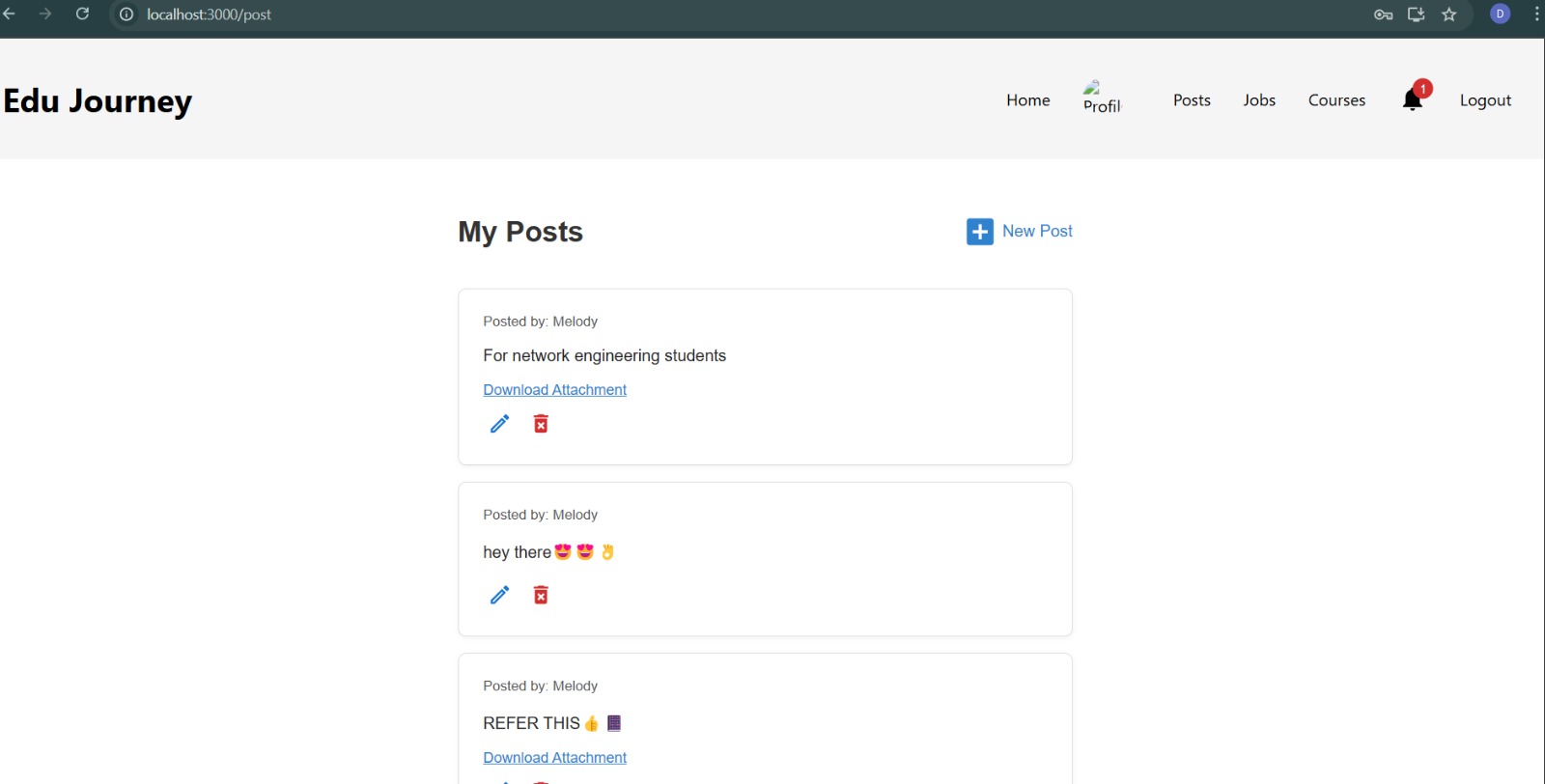
* Users can update existing posts they created using the pencil icon.
* Useful for correcting or improving shared content.
* Infinite scroll for better performance

**Backend**

* Automatically saves the post to the database.
* Fetches and returns a list of all posts from the database.
* Fetches a specific post using its ID.
* Finds the post by ID, updates the description (and other fields if added), and saves the changes.
* Deletes the specified post from the database if found.

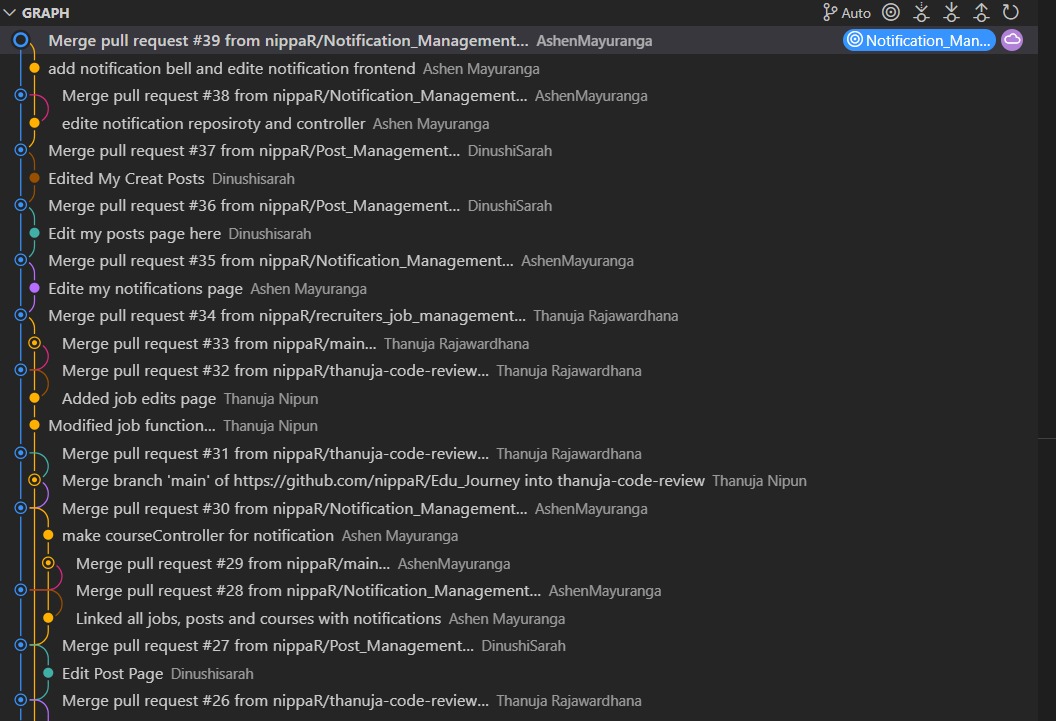


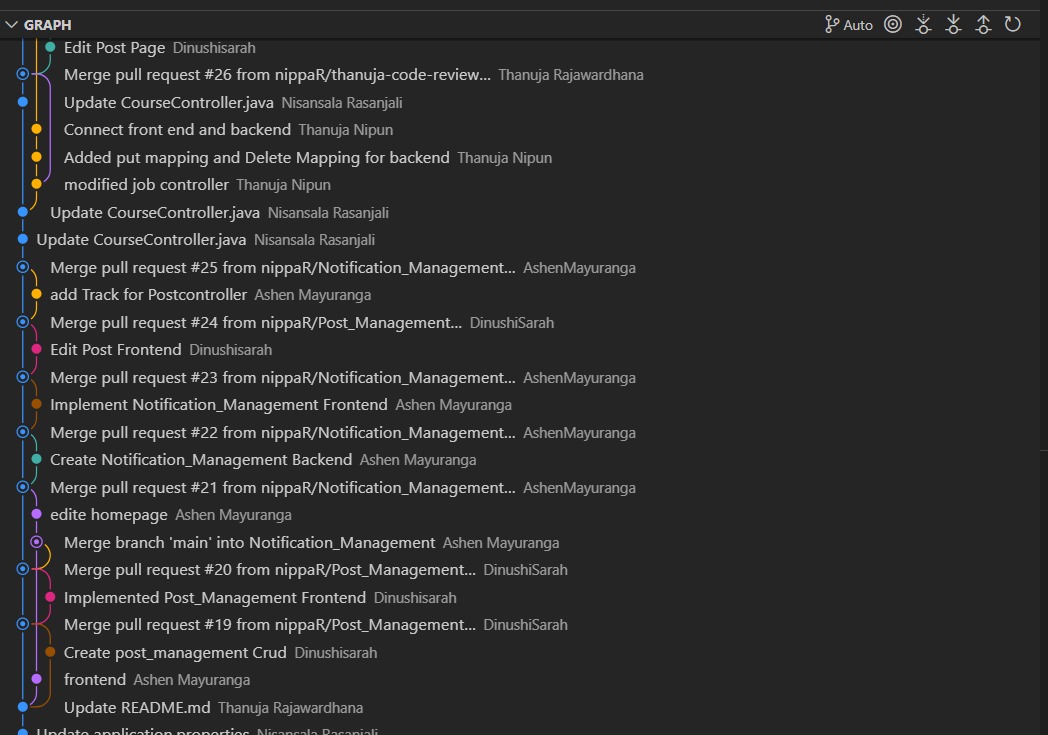




# GitHub

* **GitHub Repo link** -: <https://github.com/nippaR/Edu_Journey>
* **GitHub Commit Tree**





# References