Software Requirement Specification For One Credit Course Exemption

NAME	OVIYASHREE T
ROLL NO	7376221EC254
SEAT NO	107
PROJECT ID	17
PROBLEM STATEMENT	ONE CREDIT COURSE EXEMPTION

1. Introduction

1.1. Purpose:

The purpose of this document is to present a detailed description of the One credit course exemption portal. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate, and how the system will react to external stimuli.

1.2. Scope of Project:

This software system will serve as a portal for the One Credit Course exemption, enabling students to submit their completed one-credit course. One course can be exempted if three one-credit courses are completed successfully. From an administrative perspective, this system will provide a comprehensive analytical dashboard for credits earned through completing the course following the exemption of the course if required credits earned.

Administrators have the ability to approve or reject the request. The system will calculate whether the required credits for exemption of course are earned through one credit. If three one credit courses are completed then request will be approved by an administrator. Once the request is approved, students can exempt from the course of their choice.

2. System Overview:

2.1. Users:

1. Students:

They have the ability to submit applications for course exemption approval, upload relevant course completion documents, monitor the status of their application, and review their interaction history.

2. Admins:

Review submitted applications for course exemption, approve or reject applications (with remarks), and access analytical dashboards for oversight.

2.2. Features:

1. Login and registration:

Students can register for an account or login with their existing account.

2. Application Submission:

Students can input relevant details regarding their application for course exemption including course title, name, cluster, and any necessary attachments. Upon completion, the application is submitted to the admin interface for review and further processing

3. Application Status:

Students can view the current status of their application and also see the history logs in the option Activity logger.

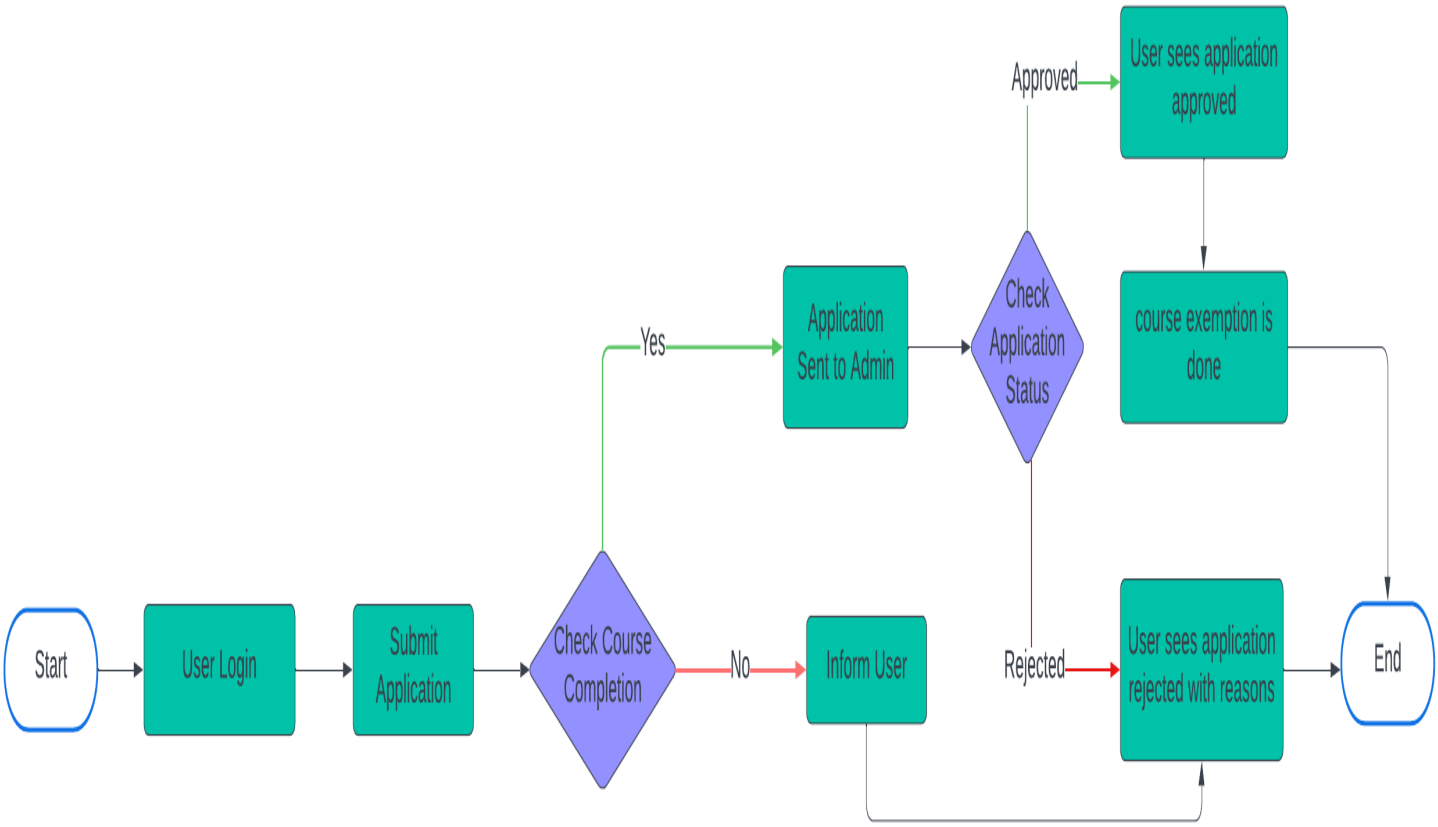
4. Admin Access:

Admin can view all submitted applications in a category of either software or hardware, view application details, approve or reject the application with suitable remarks.

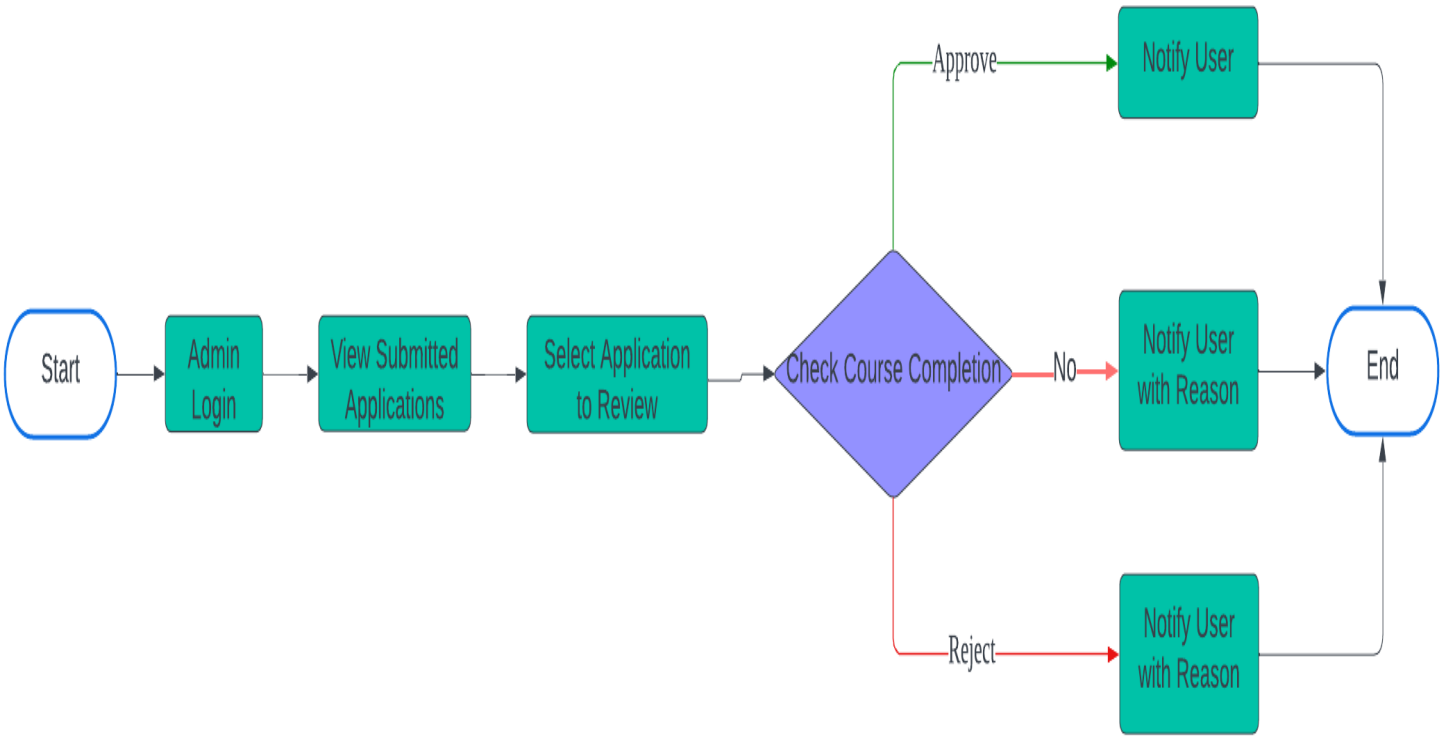
5. Admin’s Analytical Dashboard:

Admin can view the number of applications by category, appointments request and also see the latest log of applications.

User flowchart



Admin flowchart



3.1 Functional Requirements:

- **User Management:**
  - Students can register and login.
  - Admins have access control with an analytical dashboard and dedicated features.
- **Course exemption Application:**
  - Students can submit applications with appropriate details
  - Application form contains:
    - Name
    - Roll number
    - Department
    - Number of one-credit courses completed
    - Number of credits earned
    - Course to be exempted
- **Application Status:**
  - Students can view the current status of their application.
  - Students can also see the logs of their applications.
  - If approved, the system must notify the user of approval.
  - If the application is rejected then the remarks are shown.
- **Admin Dashboard:**
  - Admins can view a list of all submitted applications.
  - Admins can view the details of each application.
  - Admins can approve or reject applications with suitable remarks.
- **Analytics Dashboard:**
  - Admin can view the number of applications by its category at any time.
  - Number of appointments is requested based on the category.

3.2. Non-Functional Requirements:

- **Performance:**

The system must respond to user actions within 2 seconds to ensure efficient usability and must handle a concurrent user load of at least 100 users without significant performance degradation.

- **Security:**

User data must be encrypted during transmission and storage, and access to sensitive functionalities should be restricted to authorized admin users through secure authentication mechanisms.

- **Usability:**

The user interface should be intuitive and user-friendly, with clear and concise error messages provided to guide users in case of input errors or system failures.

- **Reliability:**

The system should be available 24/7 with minimal downtime and should have a backup and recovery mechanism in place to prevent data loss in case of system failures or crashes. Data integrity must be maintained during the submission, review, and status check process.

- **Scalability:**

The system should be designed to accommodate an increasing number of users and data volume over time, and it should be scalable to support additional features and functionalities as per future requirements. The system should be able to handle increased load during peak times, such as registration periods.

- **Maintainability:**

The system should be designed for easy maintenance and updates. Documentation should be provided for system functionalities and processes.

- **Compliance:**

The system should comply with relevant educational and data protection regulations.

4. Backend:

1. Student Entity

Name	String
Email	String
Password	Hash Code
Roll No	String

2. Course Exemption Form

Roll No	String
Details	Array of Objects

Category	String(drop down)
Title(course)	String
No of one Credits Attended	Number
No of Credits Earned	Number
pdfPath (proof for course completion)	String(using muller)
Status	String
CreatedAt	Date

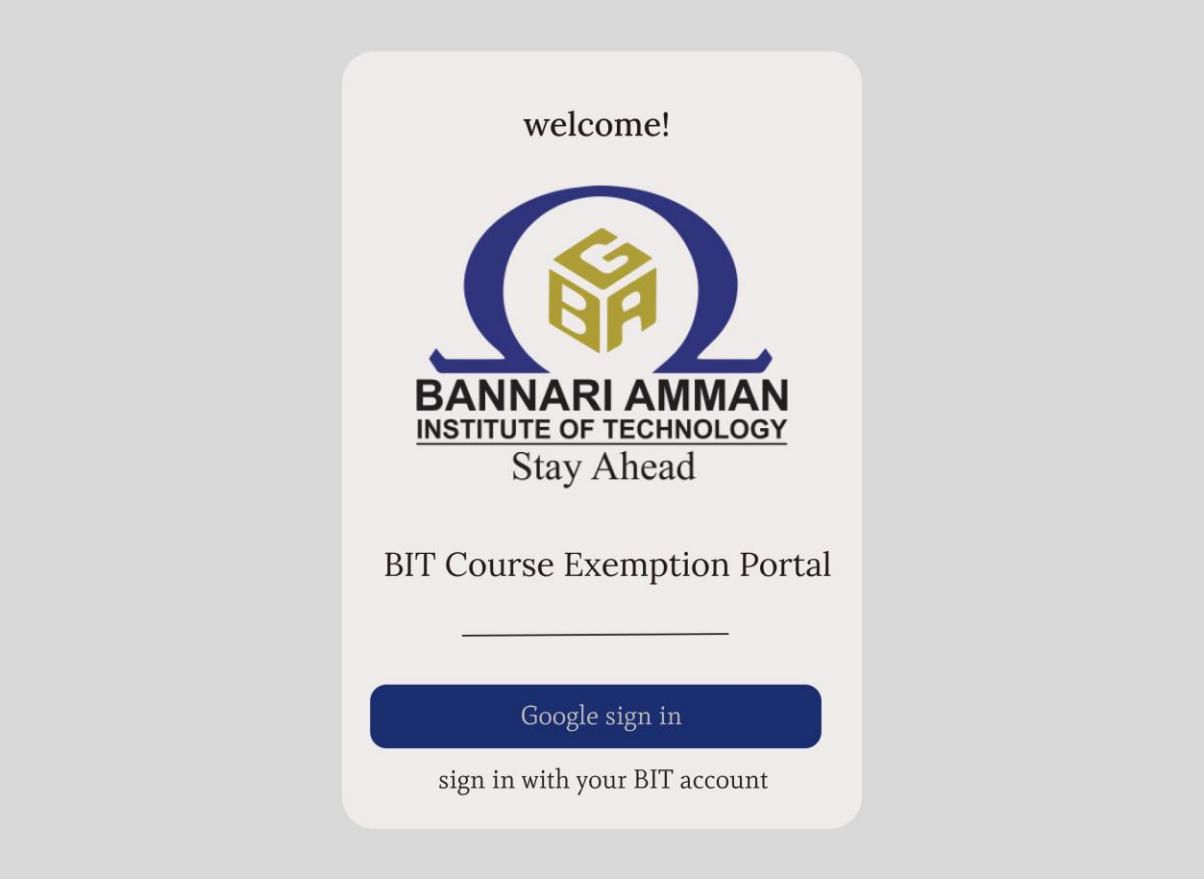
5. Stack:

MERN stack is used.

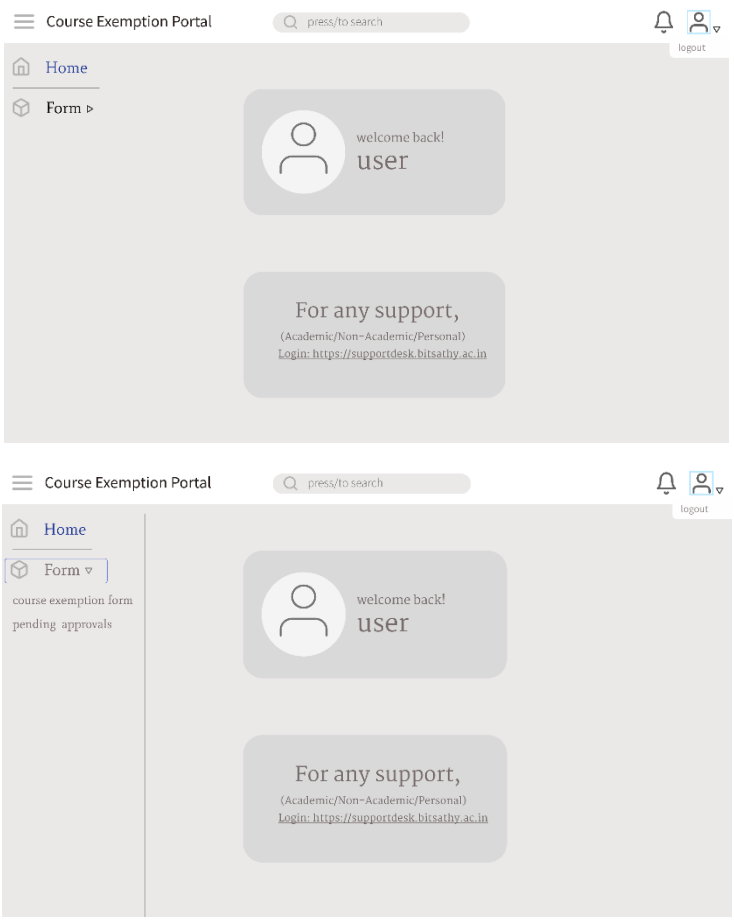
Front End	React (JS Library for building user interfaces)
Back End	Node.js with Express.js
Database	MongoDB(NOSQL Database)
API	OpenAPI

6. Prototype of the project:

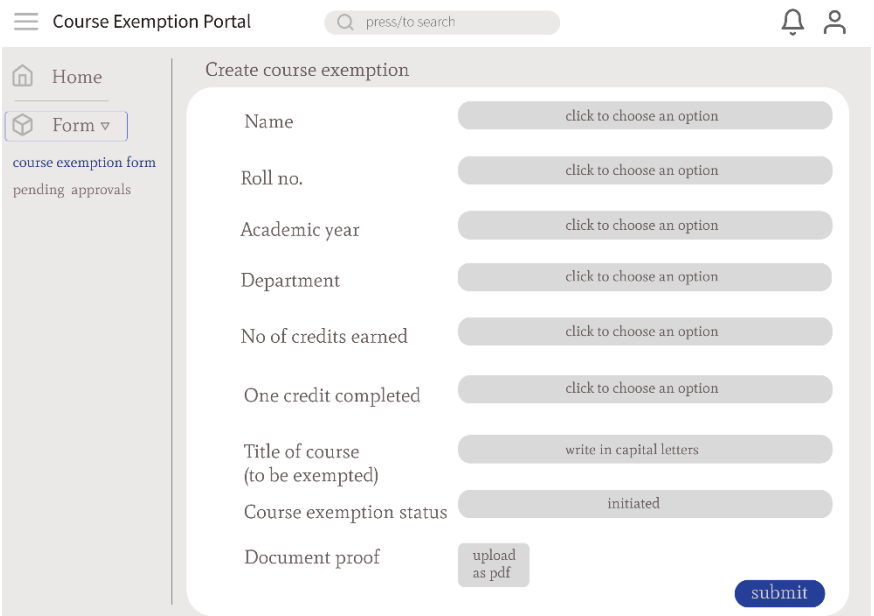
1. Login form:



2. Home page:

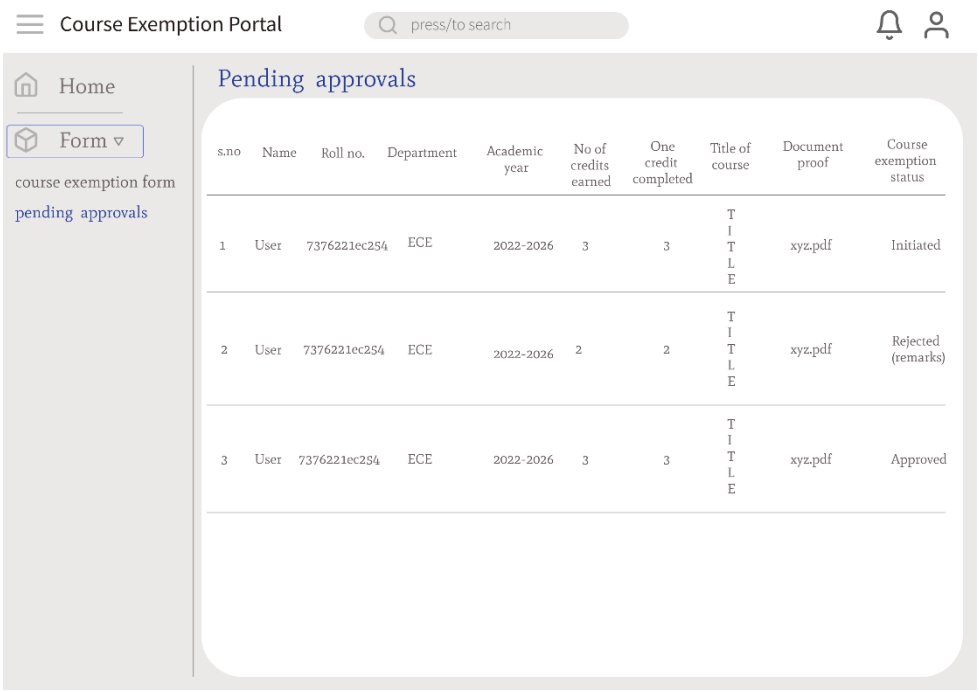


3. Course exemption form



4. Approval form

4.1 User page



4.2 Admin page