

# **1 INTRODUCTION**

## **1.1 Introduction**

Scholarship management system of our college needs to track which students have applied for scholarships. To address this need, we propose the development of a scholarship management website that enables students to register and submit their scholarship details, specifically their OTR (One-Time Registration) number issued by the National Scholarship Portal (NSP). This OTR number is a unique identifier assigned to students after they apply for a scholarship. Our system will allow students to upload their OTR card, input their OTR number, along with additional personal details such as father's name, mother's name, and date of birth. The system will then extract text from the uploaded OTR card and validate whether the provided OTR number matches the extracted one. If the numbers match, the student will receive a success email; otherwise, an alert will be issued for incorrect submissions.

## **1.2 Objectives**

- 1.** Design and implementation of a web-based platform for scholarship form submission and verification.
- 2.** Facilitate the college scholarship department to check the status of Student's scholarship application.
- 3.** Auto-generate and send confirmation messages to the user's email upon successful verification and submission.

## 2 SYSTEM REQUIREMENTS

Before installing and running this project, you need to ensure that your system is equipped with the hardware and software requirements. This section offers the minimum and recommended specifications needed to install, construct, and deploy the application successfully.

### 2.1 Hardware Requirements:

These include devices such as the processor (CPU), memory (RAM), storage (HDD/SSD), motherboard, graphics card (GPU), and input/output devices (keyboard, mouse, monitor, etc.). Hardware provides the foundation for running software and applications.

1. **Processor:** Intel Core i3 (minimum) / i5 or higher recommended
2. **RAM:** 8GB (minimum), 16GB recommended for smooth performance
3. **Storage:** At least 20GB of free disk space

### 2.2 Software Requirements:

This section outlines the hardware and software requirements needed to install, develop, and run the project efficiently.

1. **Operating System:** Windows 10/11
2. **Code Editor:** VS Code.
3. **Database:** MongoDB.

### 2.3 Development Environment

The deployment environment defines the necessary tools, frameworks, and configurations required to successfully host and run the application.

### **2.3.1 Frontend:**

1. React.js(with vite)
2. JavaScript
3. Node.js (v16 or higher)

### **2.3.2 Backend:**

1. Node.js (Express.js)
2. MongoDB (Database)
3. Postman (for testing)

## **2.4 Dependencies & Package**

This section lists the essential dependencies and packages required for the project, including libraries and frameworks needed for development and execution.

### **2.4.1 Frontend:**

- @reduxjs/toolkit (For state management in React apps)
- @tailwindcss/vite (For integrating Tailwind CSS with Vite.
- @tinymce/tinymce-react (For using TinyMCE, a rich text editor in React)
- axios (For making HTTP requests)
- lucide-react (For using Lucide icon set in React.
- react (Core library for building UI components.
- react-dom (For rendering React components in the DOM)
- react-hook-form (For handling form validation and state.
- react-icons (For adding icons from multiple icon libraries.
- react-redux (For connecting Redux with React components)
- react-router-dom (For handling routing in React apps)
- tailwindcss (For utility-first CSS styling.
- tesseract.js (For performing OCR—extracting text from images)

### 2.4.2 Backend:

- axios (For making HTTP requests)
- bcrypt.js (For hashing passwords)
- cookie-parser (For parsing cookies in requests)
- cors (For enabling Cross-Origin Resource Sharing)
- dotenv (For loading environment variables)
- express (For building the backend server)
- gridfs-stream (For handling large file uploads in MongoDB)
- helmet (For securing Express apps by setting various HTTP headers)
- jsonwebtoken (For authentication using JWT)
- mongoose (For interacting with MongoDB)
- multer (For handling file uploads)
- nodemon (For automatically restarting the server during development)

### 3 SOFTWARE REQUIREMENTS ANALYSIS

Software Requirement Analysis is the process of identifying and defining the necessary software components to develop an efficient and scalable system. This chapter outlines the problem, proposed solution, and the required software specifications to ensure seamless functionality and usability of the system.

#### **Problem Definition**

In the existing system, students generate OTR Card from NSP portal and submit their OTR (Online Transaction Reference) number on their personal account on the government PMS (Post- Matric Scholarship) site. However, the college scholarship management system faces difficulties in identifying students who have not submitted their OTR details.

#### **Challenges in the Existing System:**

1. No centralized way to track OTR submissions.
2. Manual verification leads to inefficiencies.
3. Increased chances of missing student records.

#### **3.1 Proposed Solution**

To address these challenges, we propose a Student OTR Submission and Verification System that allows students to upload their OTR number along with an image for verification. The system will then store the OTR number with student details, ensuring transparency and efficiency in scholarship tracking. To address Problems in existing system here are key features of proposed system:

1. **Student OTR Upload:** Students can submit their OTR number and upload an image for making sure that filled OTR number is matched with OTR card image.
2. **Verification Process:** The system verifies the OTR number before storing it in the database.

3. **Student Record Management:** Stores student details along with their OTR numbers.
4. **Admin Dashboard:** The college management can track students who have submitted or missed their OTR submissions.
5. **Notification System:** Sends email to students after submitted their OTR.

Above proposed solution can be achieved by implementing following features or module in web application.

#### **1 Student Module:**

1. **Login:** Students create an account after log in.
2. **OTR Submission:** Students enter their OTR number and upload an image.
3. **View Submission Status:** Students can check whether their OTR is verified.

#### **2 Teacher Module:**

1. **Verify OTR Submissions:** Admins check the validity of OTR numbers.
2. **Manage Student Records:** View and manage student details with OTR status.
3. **Generate Reports:** Track students who have submitted or missed their OTR.

## 4 SOFTWARE DESIGN

To achieve a well-structured and efficient solution, different design models are used, including flowcharts, UML diagrams, Data Flow Diagrams (DFDs), sequence diagrams, use case diagrams, and database designs. These diagrams illustrate the working of the system, module interactions, and data processing, serving as a blueprint for the actual implementation. The following sections will provide detailed insights into the system's design, covering its logical structure, functional components, and deployment architecture.

Detailed design of proposed work is given below:

appropriate design diagrams to give various solution of our proposed work is given below

1. Flowchart
2. UML diagram
3. DFDs
4. Sequence Diagrams

**1. Flowchart:** A flowchart is a graphical representation of a process that illustrates the sequence of steps using different symbols. Below is a flowchart description for the OTR (One-Time Registration) Verification Process along with an explanation of the symbols used Symbols used in Flowchart:

- Oval (Terminator) Represents Start and End points.
- Rectangle (Process) Represents a task or process.
- Diamond (Decision) Represents a decision point with Yes/No paths.
- Arrows (Flowline) Shows the direction of process flow.

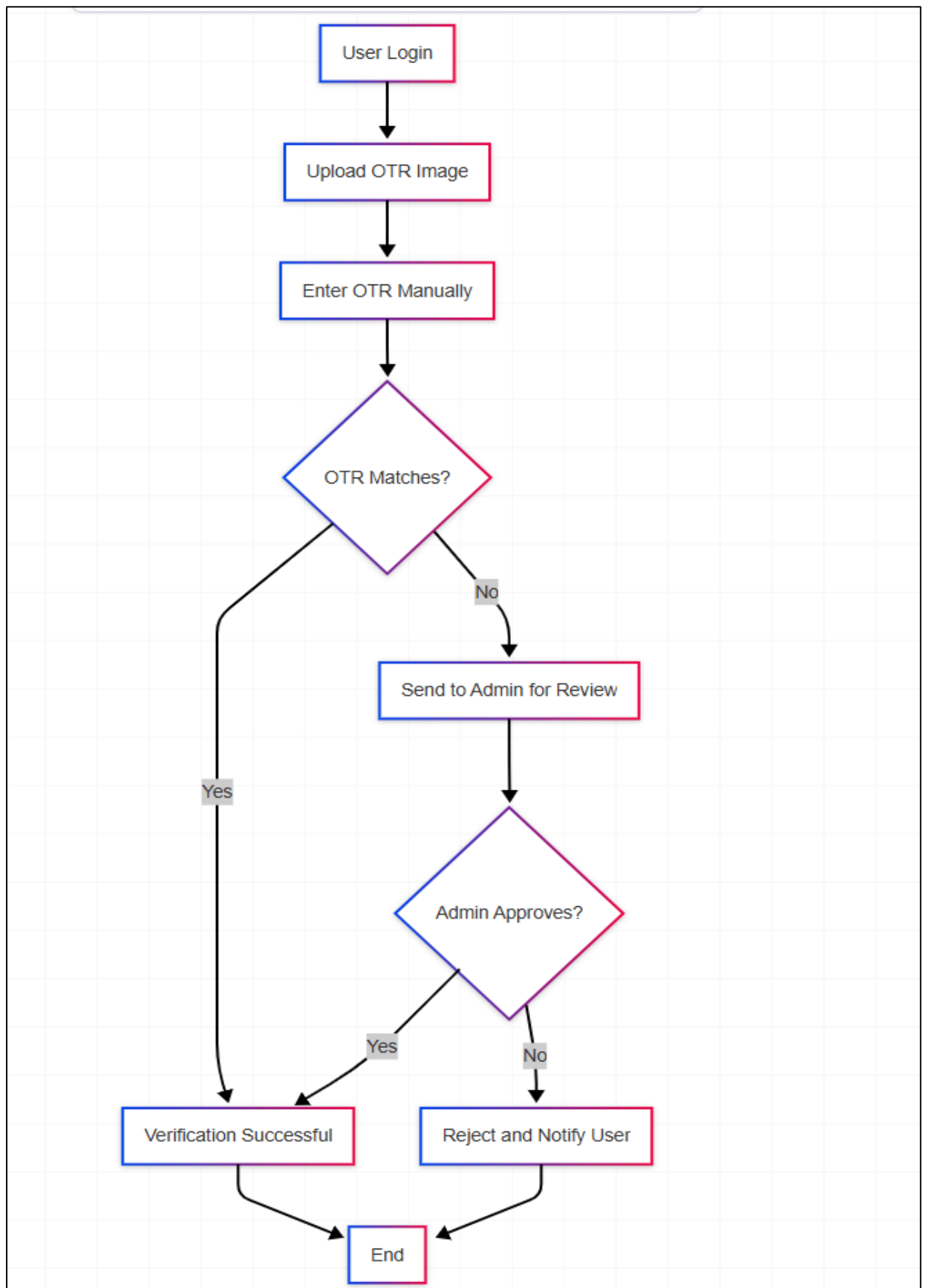


Figure 4.1 Flowchart of purposed work



## 2. UML Diagram

This UML Component Diagram represents a system designed for handling the upload, verification, and approval or rejection of Official Transcript Record (OTR) images. The system consists of three primary components: User, System, and Admin.

- 2.1.1. The **User Component** represents an individual interacting with the system. It provides two main functions: login(), which allows the user to authenticate, and upload OTR Image(), which enables the user to upload an OTR image for processing.
- 2.1.2. The System Component is responsible for processing and verifying the uploaded OTR image. It includes the store OTR() function, which saves the uploaded OTR image, and the verify OTR() function, which ensures the validity of the OTR details before sending them for review.
- 2.1.3. The Admin Component handles the review process and determines the final status of the OTR. It consists of the approve OTR() function, which approves the OTR if it meets the required criteria, and the reject OTR() function, which rejects it if it is deemed invalid.

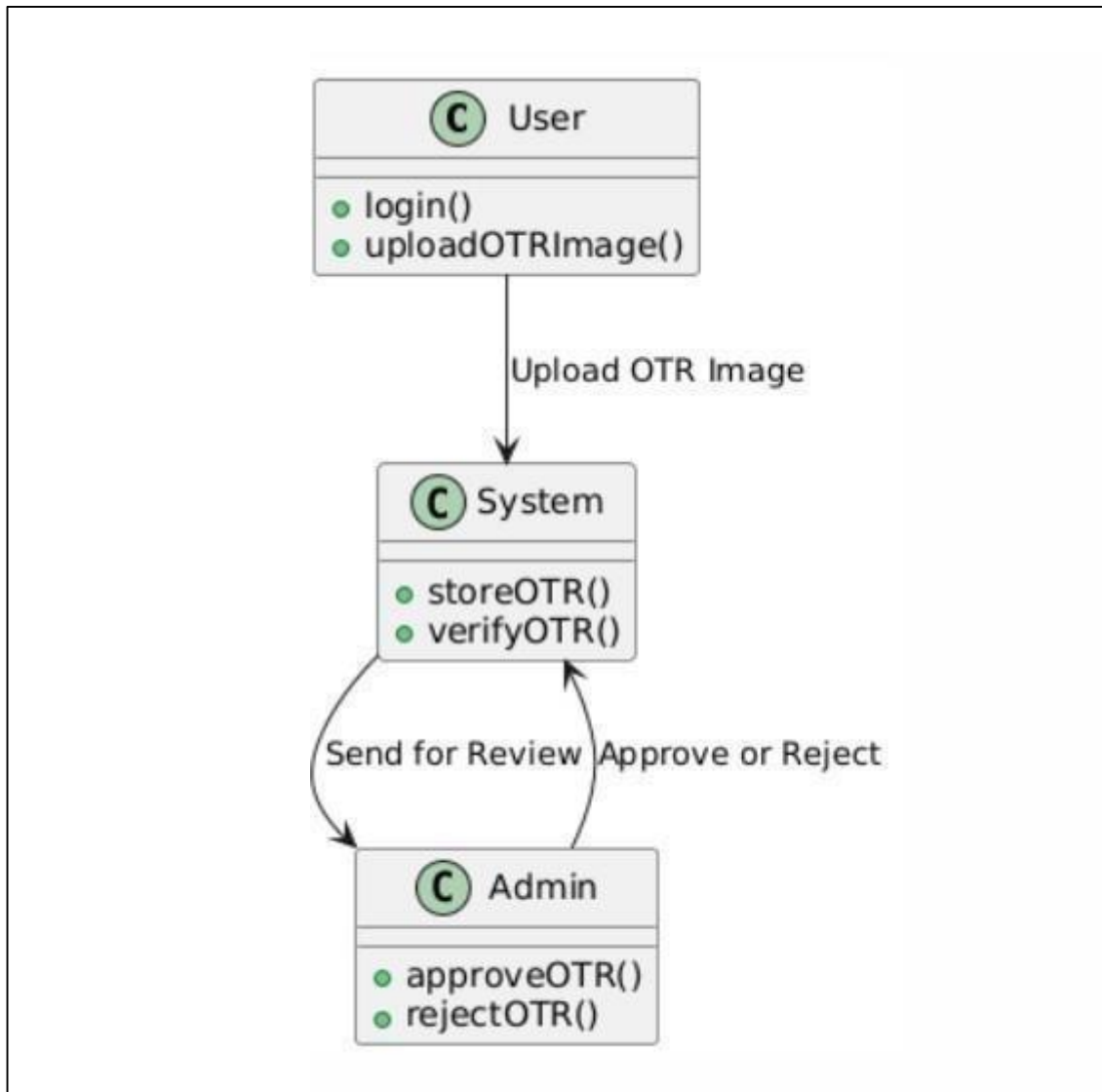


Figure 4.2 UML Diagram

### 3. DFD's

#### 3.1 DFD Level 0 (Context Diagram)

This is a high-level overview of the entire system, showing only the main entities and data flow.

**Components:**

**External Entities:**

User: Uploads OTR image and receives verification status.

Admin: Reviews and approves/rejects OTR.

**Processes:**

OTR Verification System: Central process handling uploads, verification, and admin approval.

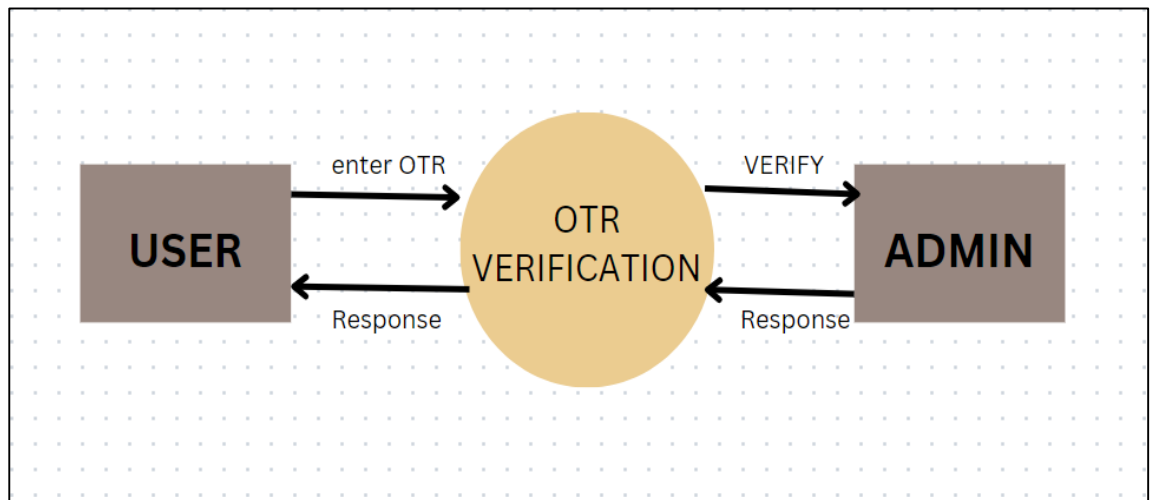
**Data Flow:**

User → OTR Verification System: Uploads OTR image.

OTR Verification System → Admin: Sends OTR for review.

Admin → OTR Verification System: Approves or rejects.

OTR Verification System → User: Sends verification status.



*Figure 4.3 DFD level-0*

### 3.2. DFD Level 1

This expands the main process into multiple subprocesses.

**Processes in Detail:**

**Login & Upload OTR (User)**

The user logs in and uploads an OTR image.

**Store & Verify OTR (System)**

The system stores the OTR and verifies its authenticity.

**Review OTR (Admin)**

The verified OTR is sent to the admin for review.

### Approve or Reject (Admin)

- The admin either approves or rejects the OTR.
- Send Notification (System)
- Sends the verification result back to the use

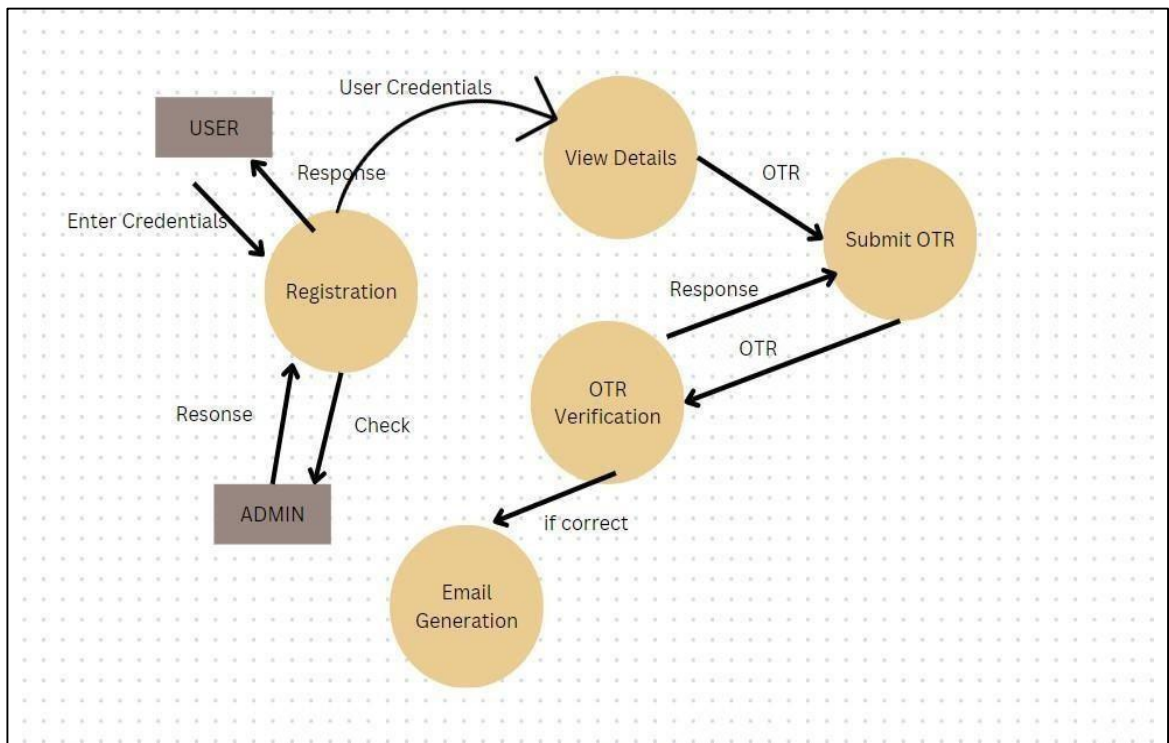


Figure 4.4 Level-1 DFD

## 4. Sequence Diagram

A Sequence Diagram represents the interaction between different components in a system over time. It shows how messages are exchanged between objects (actors and system components).

### Actors & Components:

- User → Logs in and uploads an OTR image.
- System → Processes and verifies the uploaded OTR.
- Admin → Reviews and approves/rejects the OTR.

### Flow of Interaction:

- User logs in → Sends login credentials to the System.

- System verifies login → Authenticates the user.
- User uploads OTR image → System receives and stores it.
- System verifies OTR → Checks the validity of the image.
- System sends OTR to Admin → If verification is required.
- Admin reviews OTR → Decides whether to approve or reject.
- Admin sends response → Approves or rejects OTR.
- System updates OTR status → Sends approval/rejection notification to the user.

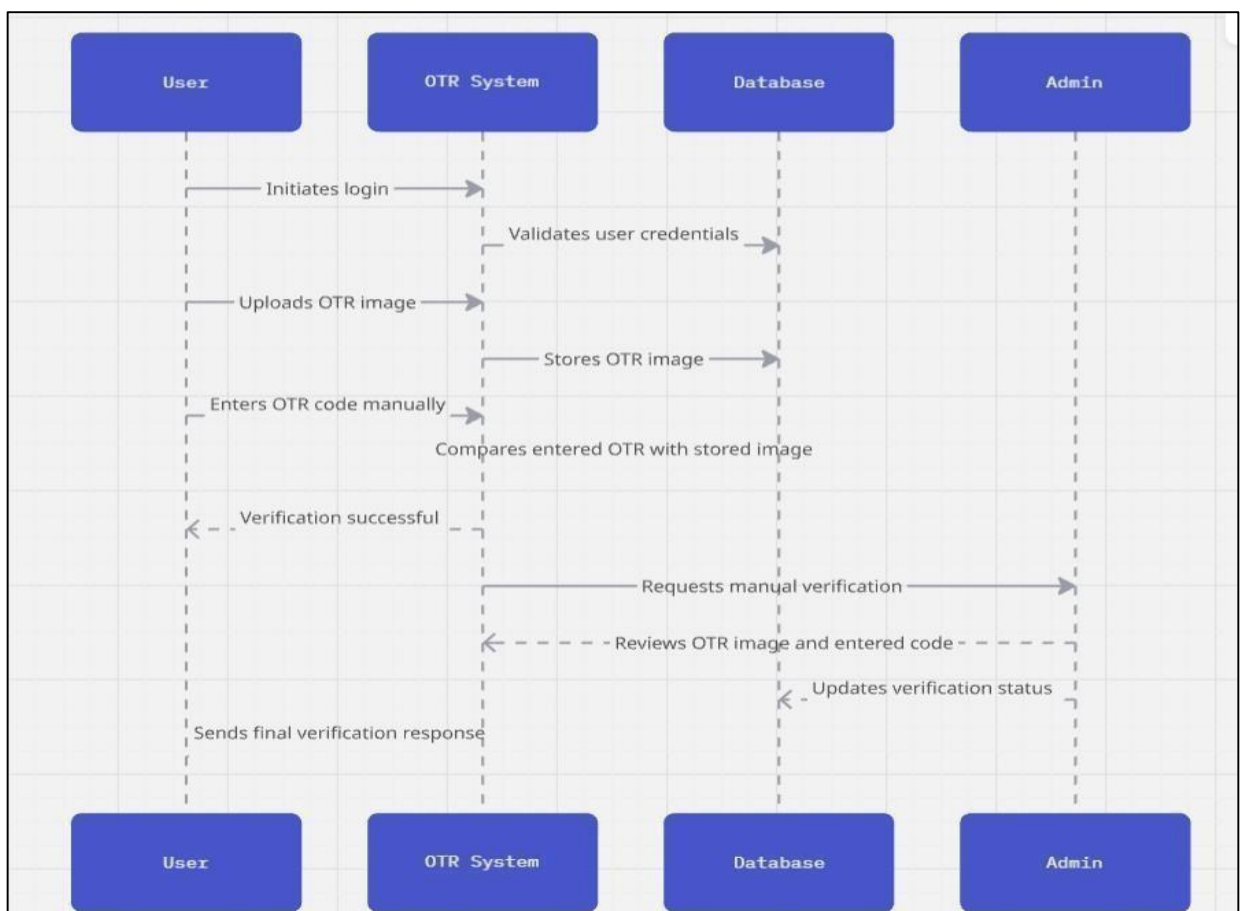


Figure 4.5 sequence diagram

## 1. Use Case Diagram

A Use Case Diagram represents the interactions between different actors (users) and the system. It visually depicts what the system does without detailing how it does it.

Actors:

### User Actions:

1. Logs into the system.
2. Uploads OTR image.
3. Receives verification status.
4. System Actions:
5. Verifies the OTR image.
6. Sends OTR for admin review (if required).

### Admin Actions:

1. Reviews OTR.
2. Approves or rejects the OTR.

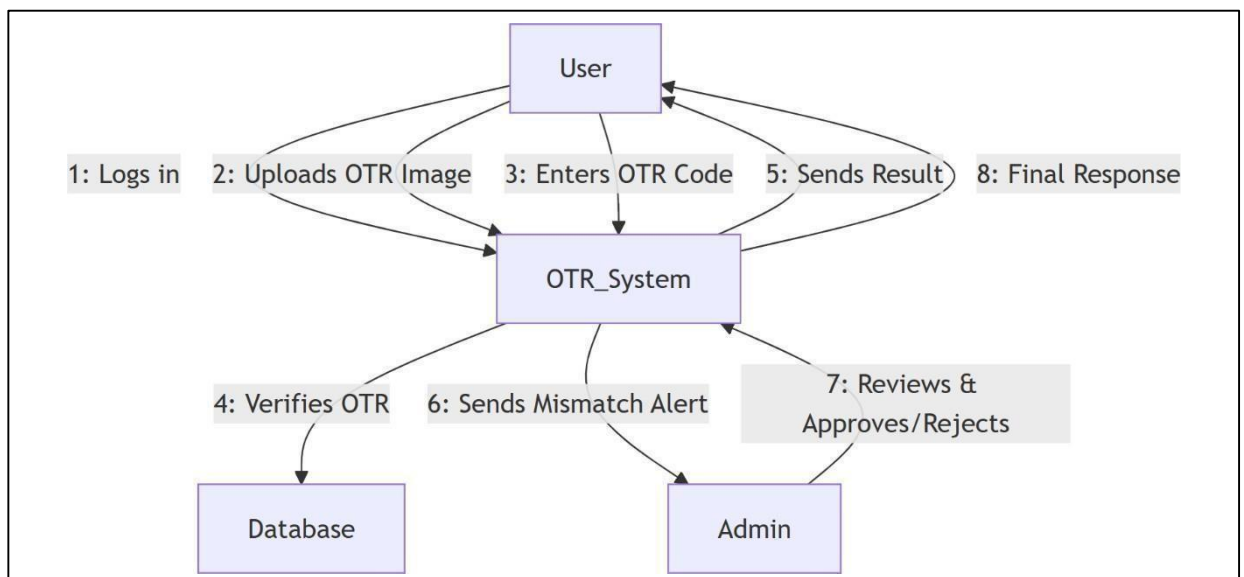


Figure 4.6 Use Case Diagram

## 5 TESTING

Testing is a fundamental step in ensuring that the proposed system functions correctly, without logical or syntactic errors. The Online OTR Verification System undergoes multiple stages of testing to verify that all features work seamlessly and as expected. To evaluate the functionality of various components, Postman has been used to test API endpoints, while MongoDB Atlas ensures secure data storage and retrieval. The following testing procedures were conducted to validate the system's functionality:

### 1. Login Testing

Login functionality is a critical component of the system, ensuring that only authorized users can access the platform. The login API was tested using Postman, where student credentials were submitted to verify authentication. The API successfully accepted POST requests containing valid student login data and returned a JWT token for session management. Incorrect credentials were tested to ensure that unauthorized access was properly restricted. The test confirmed that the login mechanism works as expected, securing user sessions with authentication tokens.

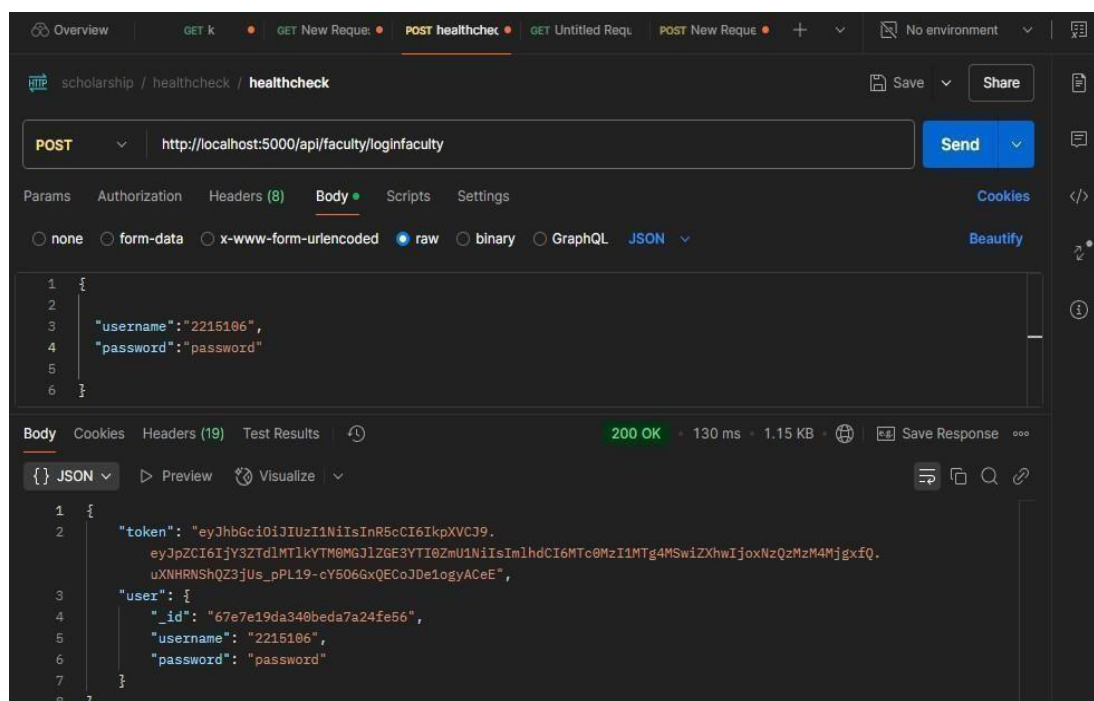


Figure 5.1 Login testing

## 2. Student Details Testing

After successful login, students should be able to access their personal and academic details. Using Postman, the student details API was tested by sending a GET request with an authentication token. The system successfully retrieved and displayed the correct student details stored in the database. The API correctly handled authorized and unauthorized access, ensuring that only logged-in students could view their details.

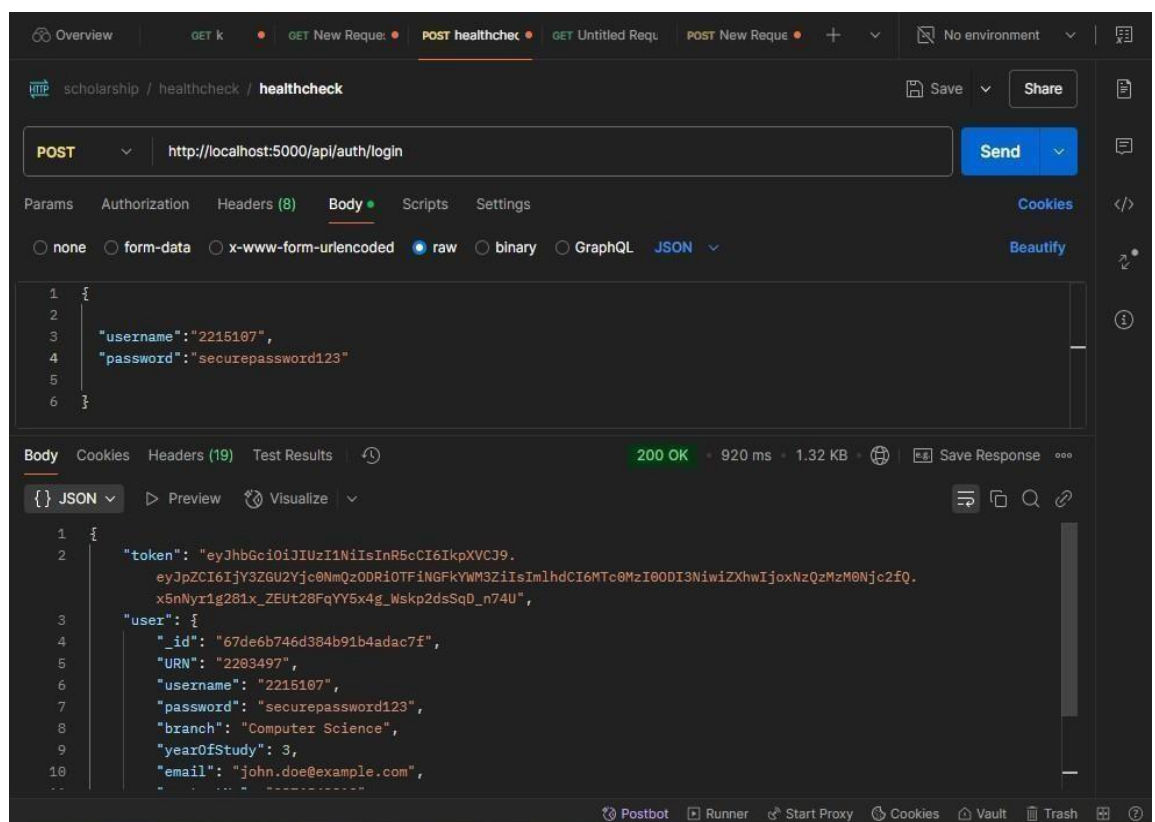


Figure 5.2 student details viewing after login

## 3. OTR Verification Testing

OTR verification is a key feature that ensures the authenticity of student submissions. The testing involved uploading an OTR document and validating its content against the records stored in the system. The API processed the uploaded image and extracted relevant details using OCR (Optical Character Recognition). The extracted data was then cross-checked with the existing records in MongoDB Atlas. Test cases were performed with both valid and invalid OTR documents, verifying



that the system. correctly identified and flagged discrepancies. The results confirmed that the OTR verification feature accurately detects mismatched data and prevents incorrect records from being stored.

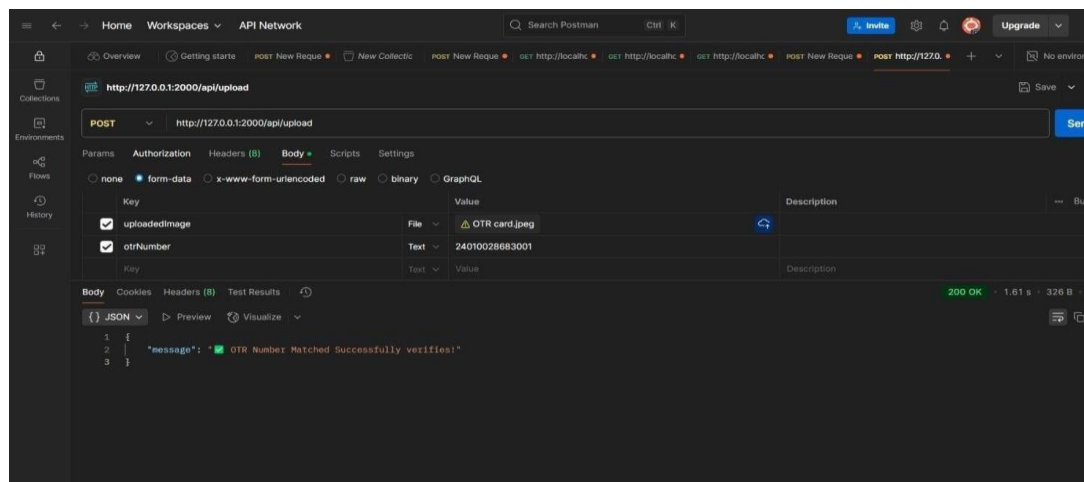


Figure 5.3 OTR Verification Testing

#### 4. Local Storage Data Storing Testing

To ensure secure data handling, MongoDB Atlas is used as the database for storing student records and OTR submissions. The system was tested for correct data storage and retrieval processes. After a student logs in, their session details and profile information are securely stored in MongoDB. The admin role-based access was also tested to verify that only authorized administrators could access the database. The test confirmed that sensitive student data remains protected and is accessible only to authorized personnel.

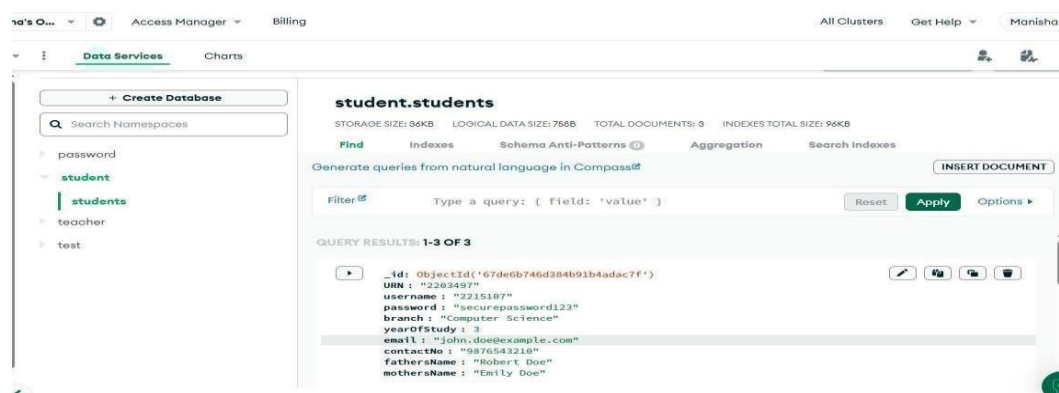


Figure 5.4 user details in database

## 5. Storing extracting data from the OTR image to the Mongo DB Atlas.

This test was conducted to verify whether the system correctly extracts data from the uploaded OTR images and stores it in the database. The OCR module was tested using multiple OTR images of different formats (JPEG, PNG, and PDF). The system successfully extracted key details such as student name, roll number, marks, and course details and stored them in MongoDB Atlas. The database retrieval function was also tested to ensure that extracted data could be accessed for verification purposes. The results confirmed that the data extraction and storage mechanism works efficiently without errors.

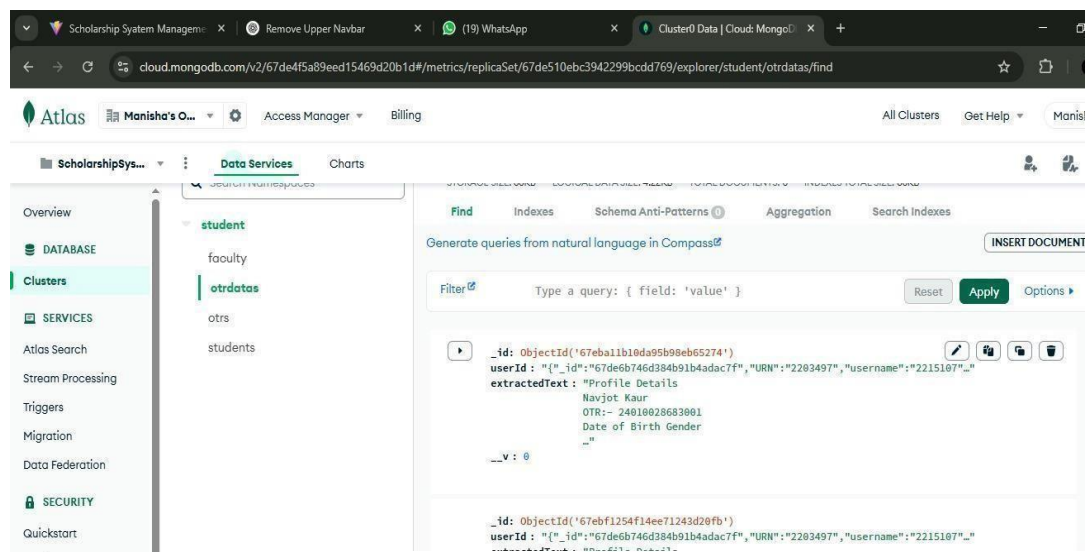


Figure 5.5 Storing extracting data form image

## **6 PERFORMANCE OF THE PROJECT DEVELOPED**

### **Project Overview**

The Online OTR Verification System is a web-based application developed to simplify and automate the submission and verification of OTR (Online Transcript Record) for students applying for scholarships. The system enhances efficiency by providing a secure platform for students to submit their OTR and for teachers to verify the records. It ensures data accuracy, prevents fraudulent submissions, and facilitates smooth communication between students and the scholarship management authority.

The project has successfully implemented several key features that improve the overall functionality and reliability of the system. Below is a breakdown of the major components developed in the project.

### **Key Accomplishments**

#### **1. Landing Page**

The landing page serves as the first point of interaction for users. It provides an option for users to select their role from a dropdown menu. If a student selects their role, they are directed to the student dashboard; otherwise, they are redirected to the teacher dashboard. The landing page also includes contact details for assistance and an announcement section displaying scholarship-related updates.

#### **2. User Authentication System**

A secure authentication system has been implemented using JWT-based authentication and token verification. This ensures that only authorized users can log in. The system supports user registration, login, and session management while enforcing role-based access control for students and teachers.

### **3. Student Dashboard**

Upon successful login, students are directed to their personalized dashboard. The dashboard provides an overview of the student's activities, including the ability to submit their OTR and track the status of their submission.

### **4. View Details**

Students can view their personal and academic details, ensuring that their profile information is accurate and up to date. This feature allows them to verify the correctness of their records before submission.

### **5. Submit OTR**

Students can submit their OTR card by uploading an image. The uploaded document is securely stored in the database for further processing. This feature ensures that all records are maintained digitally, reducing the risk of data loss.

### **6. OTR Verification**

The system verifies the uploaded OTR by cross-checking it with the records stored in the database. If any discrepancies are detected, the system notifies the student to resubmit the correct information. This automated verification process helps maintain the accuracy and integrity of the student database.

### **7. Teacher Dashboard**

Teachers have access to their own dashboard, where they can filter student records based on branch and batch. They can also monitor the status of OTR submissions and ensure that all students have completed the verification process before their applications are forwarded for scholarship approval.

#### **8. Filtering Student Data**

Teachers can filter student records based on specific criteria such as branch and batch. This feature makes it easier to manage large amounts of student data and focus on verifying records efficiently.

## **9. Create Announcements**

Teachers can create and publish announcements related to scholarships. They can provide information on eligibility criteria, application deadlines, and required documents. Additionally, teachers can upload important documents and guidelines to assist students in the application process.

## **Performance Summary & Impact**

The Online OTR Verification System significantly improves the efficiency of the OTR submission and verification process. It eliminates manual errors by automating validation checks and ensures data security through JWT authentication. The system is designed with a user-friendly interface that allows students to submit their OTR with ease while enabling teachers to manage student records efficiently. By implementing this system, the verification process for scholarship applications has become more streamlined, transparent, and reliable.

## 7 OUTPUT SCREENS

### a) Home Page (First Objective):



Figure 7.1 Header of home page



Figure 7.2 Main Body Content of Home Page

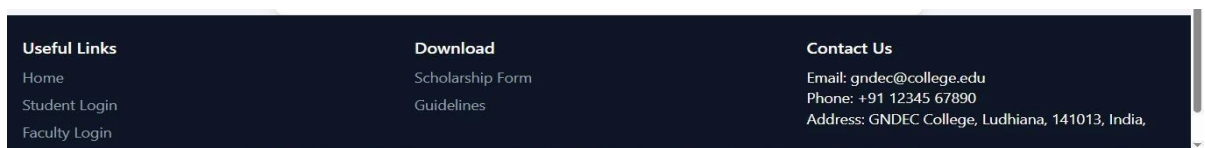



Figure 7.3 Footer of home page

b) When user select student



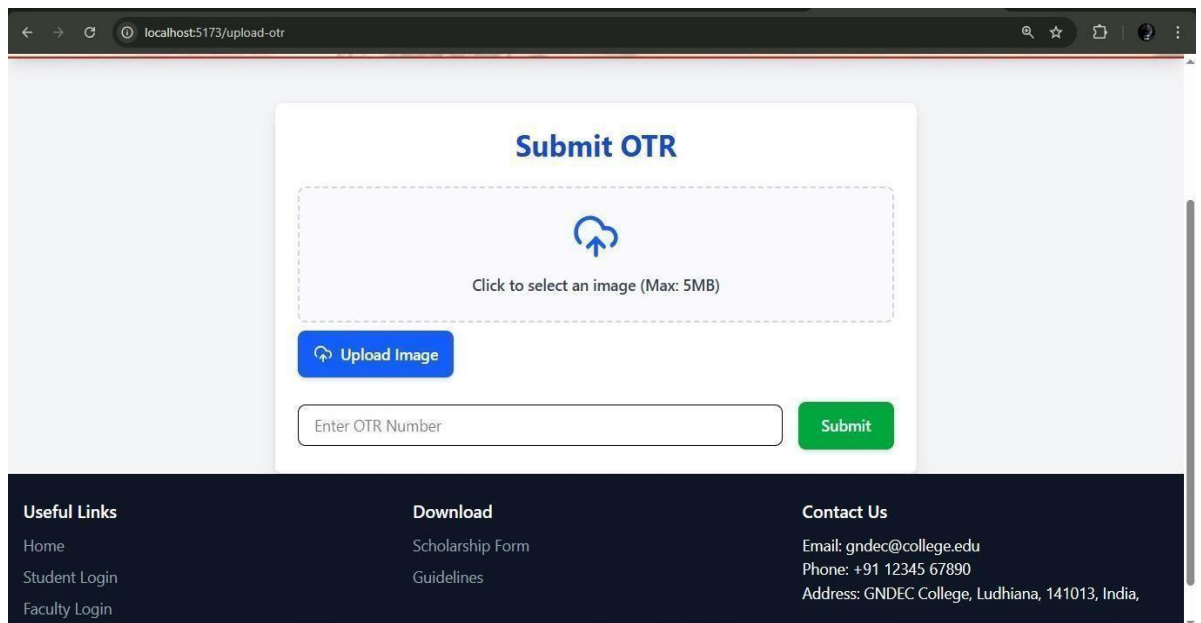
The image shows a 'Student Login' form centered on a light gray background. At the top of the form is a circular logo featuring a book and a torch. Below the logo, the text 'Student Login' is displayed in bold. The form contains two input fields: the first is labeled 'Username (Numbers Only)' with a placeholder 'Enter your student ID'; the second is labeled 'Password' with a placeholder 'Enter your password' and a toggle icon (an eye) to its right. A blue 'Login' button is positioned at the bottom of the form.

Figure 7.4 Student login tab



The image displays a student dashboard for Guru Nanak Dev Engineering College, Ludhiana. The header features the college's name in Gurmukhi and English, along with its logo and accreditation details: 'An Autonomous College w/s 2(f) and 12(B) of UGC Act 1956', 'AICTE Approved, Punjab Govt. Aided Status, ISO 9001 : 2008 Certified', 'Affiliated to I.K. Gujral Punjab Technical University, Jalandhar', 'IEI Accredited UG Programmes', and 'Institution Accredited by NAAC (A Grade) & TCS'. The main navigation bar is green and includes links for 'Home', 'Control Panel', 'Info', and a user identifier '2215107'. The 'Control Panel' dropdown menu is open, showing options: 'View Profile Details', 'Upload Documents', and 'Submit OTR Card'. The 'Info' dropdown menu is also open, showing 'Vision-Mission' and 'FAQ'. The central area of the dashboard displays a welcome message: 'Welcome, 2215107!'.

Figure 7.5 Student dashboard



**Submit OTR**

Click to select an image (Max: 5MB)

Upload Image

Enter OTR Number

Submit

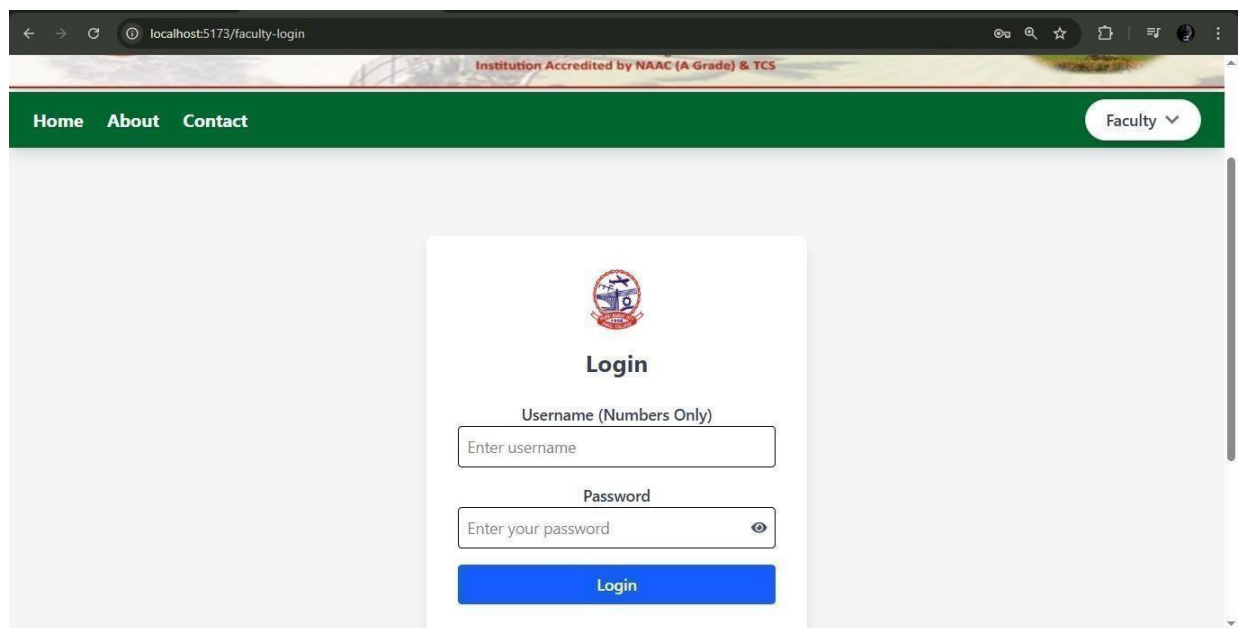
**Useful Links**  
 Home  
 Student Login  
 Faculty Login

**Download**  
 Scholarship Form  
 Guidelines

**Contact Us**  
 Email: gndec@college.edu  
 Phone: +91 12345 67890  
 Address: GNDEC College, Ludhiana, 141013, India,

Figure 7.6 Submit OTR form

c) When user select Faculty



localhost:5173/faculty-login

Institution Accredited by NAAC (A Grade) & TCS

Home About Contact

Faculty

**Login**

Username (Numbers Only)  
 Enter username

Password  
 Enter your password

Login

Figure 7.7 Faculty Login tab



localhost:5173/facultyProfile

Select Course(s):

- Mechanical Engineering
- Computer Science & Engineerin
- Civil Engineering
- Electronics & Communication Er

Hold **Ctrl** (Cmd on Mac) to select multiple.

Enter Batch:

e.g., 2023-2027

Example: 2023-2027

Submit

Figure 7.8 Faculty dashboard

localhost:5173/facultyProfile

Bachelor of Computer ALppicac

Master of Computer Applicatin

Hold **Ctrl** (Cmd on Mac) to select multiple.

Enter Batch:

2022-2024

Example: 2023-2027

Submit

Name	Course	Batch	OTR Submission	Contect No.
Navjot Kaur	Computer Science & Engineering	2022-2024	submitted	9876543210

Figure 7.9 Student OTR submitted status

## 8 REFERENCES

1. Convert Images to Text in Node.js with Tesseract.js by Medium.

<https://medium.com/@abhishekchamoli007/convert-images-to-text-in-node-js-with-tesseract-js-a-step-by-step-guide-b4fa5f5ee809>

2. Image To Text Conversion With React And Tesseract.js Article by Smashing magazine.

<https://www.smashingmagazine.com/2021/06/image-text-conversion-react-tesseract-js-ocr/>

3. International Journal of Research Publication and Reviews:

<https://ijrpr.com/uploads/V4ISSUE4/IJRPR11943.pdf>