“PORTAL FOR INSTITUTE PROJECT REPOSITORY”

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

Harshita Bhatt (224A1127)
Chinmayee More (224A1130)
Prajakta Palsamkar(224A1132)
Heena Patil (224A1134)

Supervisor

Dr.Aparna Bhonde

Department of Computer Engineering SIES Graduate School of Technology Nerul, Navi Mumbai - 400 706

University of Mumbai

(AY 2024-25)

CERTIFICATE

This is to certify that the Mini Project entitled “Portal for institute project repository” is a Bonafide Work of Harshita Bhatt (224A1127), Chinmayee More (224A1130) , Prajakta Palsamkar (224A1132), Heena Patil (224A1134), Submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “Bachelor of Engineering” in “Computer Engineering”.

(Dr.Aparna Bhonde)
Supervisor

(Dr. Aparna B)
Head of Department

(Dr. K Lakshmi Sudha)
Principal

Mini Project Approval

This Mini Project entitled “Portal for institute project repository” by Harshita Bhatt (224A1127) , Chinmayee More (224A1130) , Prajakta Palsamkar (224A1132), Heena Patil (224A1134), is approved for the degree of Bachelor of Engineering in Computer Engineering.

Examiners

1……

(Internal Examiner Name & Sign)

2……

(External Examiner name & Sign)

Date:

Place:

Contents

ABSTRACT

The portal for the institute project repository is a platform developed to maintain and manage academic and research projects within the institute. It serves as a comprehensive, centralized digital library that allows students, faculty, and researchers to store, organize, and access a wide range of project materials. These include documents, presentations, source code, project videos, and external resources such as YouTube links and Google Drive. By providing a single point of access for project data, the portal aims to streamline project management and foster collaboration across departments and research teams.

This portal is developed using Python with the Django framework for backend development and HTML and CSS for front-end design. SQLite is used as the database, ensuring robustness, security, and scalability. Django's flexibility and powerful ORM (Object-Relational Mapping) capabilities make it ideal for handling database interactions efficiently. HTML and CSS are used to create a responsive and user-friendly interface for both the front end and the user experience. SQLite provides a lightweight, yet robust database solution to store and retrieve project-related data.

A key feature of the portal is its ability to store multimedia content, including project videos, and link to external URLs such as YouTube. This allows users to access detailed demonstrations, tutorials, or supplementary content related to specific projects. The portal also includes advanced search functionalities, user role management, and secure access control, ensuring that users can easily find and contribute to projects while maintaining privacy and data integrity.

The system dynamically fetches data from the database to display project statistics, including the number of projects and graphical representations of project submissions, department-wise distribution, and other related metrics. Additionally, the portal includes a 'forgot password' feature, where users can verify their email before resetting the password. For security, CAPTCHA verification is added to the login process to prevent automated access. For registration, users must provide essential details such as the institute name, email ID (only gst.sies.edu.in emails are accepted), mobile number, gender, branch, PRN number, upload student ID, and set a password.

The main goal of this project is to enhance the academic and research experience by facilitating easy access to project resources, encouraging knowledge sharing, and promoting innovation. By simplifying project sharing and collaboration, the portal plays a crucial role in supporting the institute's academic community, fostering a collaborative environment, and enabling continuous learning and development across various research fields.

In conclusion, the institute project repository portal is a valuable tool that organizes and connects academic resources in a centralized, user-friendly manner. Through its efficient design and use of modern technologies, the portal is poised to significantly improve the way projects are managed, shared, and accessed, ultimately benefiting students, faculty, and researchers alike.

ACKNOWLEDGEMENT

We would like to express our thanks to the people who have helped us the most throughout our project. We are grateful to our guide Dr. Aparna Bhonde and coordinator (Prof. Suvarna Chaure) for great support for the project.

A special thanks goes to each other who worked together as a team in completing the project, where we all exchanged our own interesting ideas, thoughts and made it possible to complete our project with all accurate information. We also wish to thank our parents for their personal support and attention who inspired me to go my own way.

We would also like to extend our sincere gratitude to our Principal, Dr. K Lakshmi Sudha and our Head of the Department, Dr. Aparna Bannore for their continuous support and encouragement.

We also would like to thank our other faculty members for providing us with all the required resources and references for the project.

CHAPTER 1
INTRODUCTION

1.1 Overview

- The portal for the institute project repository is a platform designed to provide students, faculty, and researchers with a centralized location to store, manage, and share academic and research projects. This repository serves as a safe space for organizing various project materials, such as documents, codes, presentations, videos, and links to external resources like YouTube and Google Drive.

- The efficient management of these project resources is crucial for ensuring seamless access, collaboration, and knowledge sharing among users. A well-designed portal not only enhances the user experience but also encourages academic and research growth within the institute.

- In today's digital world, managing such vast amounts of data and resources can be a challenging task. With the increasing volume of academic projects, it becomes vital to have a system that simplifies and automates the management process. A portal that integrates project materials, multimedia, Google Drive links, and external resources like YouTube helps reduce the complexity of manual project management, ensuring efficiency and smooth operations.

- Given these requirements, the development of the project repository portal has become necessary. The system, built using Python with Django, HTML, and CSS, will significantly improve the management and access to academic projects, enabling better collaboration, ease of use, and seamless integration of various project-related resources.

1.2 Motivation

- The motivation behind developing the institute project repository portal arises from the increasing need for an efficient and user-friendly platform to manage academic and research projects. As educational institutions continue to produce a growing amount of academic work, the demand for a system that can efficiently store, organize, and share project materials becomes more essential than ever.

- The main objective of the project is to create a centralized, online portal that serves as a repository for academic projects. This portal will enable users to upload new projects, update existing project details, share project files such as documents, codes, presentations, and videos, and access external resources like YouTube links and Google Drive files. Additionally, the portal will provide easy access to project reports, ensuring seamless interaction and management of academic materials.

- The project also offers an opportunity to apply theoretical knowledge gained in the classroom to real-world situations, enhancing practical skills in web development, data management, and system design. This experience deepens the understanding of project management processes and helps develop problem-solving abilities, while also fostering collaboration among students, faculty, and researchers within the institute.

- Ultimately, the goal of this project is to contribute to the evolution of academic project management systems, making them more accessible, organized, and efficient for users. By streamlining the process of storing, sharing, and editing project data, the portal aims to improve collaboration, enhance research efforts, and support the academic growth of the institute.

1.3 Problem Statement & Objectives

1.3.1 Problem Statement

The Institute Project Repository Portal is designed to effectively manage academic and research projects within the institute. The portal has two primary roles: Student and Professor role.

- **Student Role:**

  The student must first register to access the portal. During registration, the following details will be collected: Institute Name, Student Name, Email ID (Only gst.sies.edu.in emails are accepted), Mobile No, Gender, Branch, PRN No, Upload Student ID, and Password.
  Once registered, students can log in to the system and perform the following tasks:

  - Upload and manage project materials, including documents, code, presentations, videos, and links to external resources such as YouTube and Google Drive.
  - View their project details (e.g., project title, description, materials uploaded).
  - A user can edit their project and the edits are saved in database appropriately.
  - User can view their profile (i.e. their details submitted during registration).

  For faculty members, dummy user IDs and passwords will be created by the administrator, allowing them to log in without needing to go through the registration process. Faculty will have access to specific functionalities for overseeing projects but will not have the ability to register themselves.

- **Faculty Role and Responsibilities**

  Faculty members (teachers or professors) must register with the portal using the same set of required information as other users, including their name, faculty ID, email ID, and any other necessary details.

  Upon successful registration and login, they gain access to the following functionalities:

  - View Student Projects : Faculty members can access all projects submitted to the portal. They can click on the project titles to view detailed project information.

- Evaluate Student Projects : Faculty can evaluate and assign marks only for the projects where they have been designated as mentors.
- Manage Profile: Faculty members can view and manage their profile information.

1.3.2 Objectives

- Crafting an Intuitive Interface : Develop an easy-to-use platform with clean and accessible designs, ensuring both students and faculty can navigate smoothly and accomplish their tasks effortlessly using HTML and CSS.

- Simplifying Project Management : Streamline essential functions like project uploads, edits, and views for students. Faculty members can review and evaluate projects submitted by their mentees, keeping the process simple and efficient.

- Strengthening Security: Put user safety first by embedding strong measures like authentication, encryption, and secure access controls, ensuring data remains protected at all times.

- Optimizing Data Management : Build a reliable database system powered by SQLite that safely stores and organizes all user information, project details, and transactional records in a structured and efficient manner.

- Bringing Real-Time Insights : Offer real-time updates to faculty through dynamic features like login dashboards that display the total number of projects, registered students, evaluated submissions, and even a visual pie chart showcasing the distribution of submitted projects—all powered by dynamic data fetching.

- Planning for Growth : Design the portal to handle increasing users and projects while being flexible enough to incorporate new features in the future—keeping scalability at the heart of the system.

1.3.3 Additional Features

- Dynamic Data Fetching : The system dynamically retrieves data from the SQLite database to showcase project statistics, including the total number of projects, graphical representations of project submissions, department-wise distributions, and a pie chart generated from database information.

- Forgot Password Functionality : Users can reset their password through the "Forgot Password" feature, which requires email verification to ensure account security.

- CAPTCHA Verification for Login : CAPTCHA verification is integrated into the login process to prevent unauthorized automated access and enhance security measures.

- Email Validation for Registration : During registration, users must submit a valid email address (e.g., gst.sies.edu.in), which is verified before account creation to maintain the integrity of the platform.

1.3.4 Organization of the Report

- **Chapter 1**: Introduction – This chapter introduces the topic, explaining the motivation behind developing the Institute Project Repository Portal. It outlines the problem statement and the key objectives of the project.
- **Chapter 2**: Existing System – This chapter discusses the current methods used to manage academic projects and highlights their limitations. It then introduces the proposed system, emphasizing how it overcomes the drawbacks of existing models.
- **Chapter 3**: Proposed System – This chapter provides an in-depth look at the proposed system, including its architectural framework, system design, algorithms, and process design needed to build the portal.
- **Chapter 4**: Design & Methodology – This chapter focuses on the design and methodology of the project. It includes UI design for each module, an explanation of the development methodology, and details on algorithm implementation.
- **Chapter 5**: Practical Implementation – This chapter demonstrates the practical implementation of the portal's features, such as input validation, user authentication, dynamic data fetching, CAPTCHA verification, forgot password, and project upload mechanisms. It includes relevant code snippets and explanations of the functionalities.

CHAPTER 2

SURVEY OF EXISTING SYSTEM

2.1 Survey of Existing System

The Yukti Portal is a versatile platform designed to manage academic and innovation-related projects. Below are its key features:

- **Login System:**
  - The portal includes a secure login system with username and password authentication.
  - CAPTCHA verification is integrated to prevent unauthorized automated access.
  - Users can register if they do not already have an account, and a "Forgot Password" feature allows them to reset their password via email verification.
  - Email verification is mandatory during registration to ensure the validity of user accounts.

- **Interactive Project Submission and Management:**
  - The portal allows users to interact dynamically by submitting their projects, which are categorized into three types: Idea, Prototype and Starter .
  - This structured system helps streamline project submissions and ensures that users can effectively manage their entries.

- **Incubator Role:**

  While the specific functionalities for incubators are unclear, the portal likely includes tools for mentoring, evaluating, or supporting innovative projects.
- **Admin Role:**
  - Administrators oversee the portal's operations, including managing user accounts, approving or rejecting project submissions, and ensuring compliance with guidelines.
  - Admins may also handle flagged content and generate reports on system activity, submission statuses, and user engagement.

2.2 Limitations of Existing Systems or Research Gap

Although existing platforms like the Yukti Portal serve as valuable tools for academic project management, several limitations remain in the context of creating an efficient, college-specific repository system. Here are the observed gaps:

- **Lack of Centralized Repository:**
  Many systems, including Yukti, require users to manage projects across multiple platforms, such as cloud storage tools, course management systems, and personal resources. This fragmentation makes project tracking and data accessibility time-consuming and inefficient.

- **Limited Customization and User Experience:**
  Current platforms often lack tailored user interfaces for specific groups, like students or faculty members within a single institution. Non-technical users may find workflows overwhelming, and existing systems do not offer intuitive dashboards or streamlined processes for various academic roles.

- **Security and Privacy Concerns:**
  Robust data security is not consistently implemented across all platforms. While Yukti offers authentication and CAPTCHA verification, data encryption and secure role-based access controls for sensitive academic content are areas requiring improvement. Without proper mechanisms, academic materials and user details may be vulnerable to unauthorized access.

- **Performance and Scalability Issues:**
  Existing systems face challenges in scaling efficiently when project data grows. Queries or searches across a large volume of materials may lead to slow performance or system overloads. Yukti, while functional, is not fully optimized for large-scale performance within a growing institution.

- **Insufficient Real-Time Updates:**
  Many existing platforms, including Yukti, do not dynamically display actionable insights in real time. Features like dashboards showing submission counts, department-specific trends, or evaluated projects are either absent or not fully developed, requiring manual intervention for data analysis.

- **Limitations in Dynamic Data Visualization:**

  While Yukti offers graphical insights like pie charts, these visualizations are often static or limited. Fully dynamic tools for fetching and displaying key project metrics, such as student participation or department-specific project activity, are needed to provide actionable information to faculty and administrators effectively.

2.3 Research Gap and the Need for a New System

Considering the limitations of existing platforms, there is a significant need for a dedicated and advanced system to effectively manage academic and research projects, particularly in a college-specific context.

This new system must address the following key areas:

- **User-Friendly Interface :**
  - A simple and intuitive interface is crucial to ensure ease of use for students, faculty, and administrators.
  - Navigation should be straightforward, allowing users to upload, manage, and share project materials seamlessly, even if they lack technical expertise.

- **Enhanced Security and Privacy :**
  - The system should implement robust security protocols, including secure user authentication, encrypted storage for sensitive project data, and role-based access controls.
  - These measures are essential to protect academic materials and personal information from unauthorized access or data breaches.

- **Centralized and Scalable Solution :**
  - A unified platform is needed to bring all project-related resources into one location, ensuring efficient and centralized management.
  - The system should use a scalable database solution like SQLite, capable of handling the growth in users and projects over time. Django's flexibility makes it a suitable choice for managing this dynamic data efficiently.

- **Dynamic Data Fetching and Reporting :**
  - The new system should dynamically fetch and display real-time statistics, such as the total number of projects, their categories, department-wise distribution, and other key metrics.
  - Comprehensive reporting tools must allow administrators to generate detailed reports, aiding in better management and analysis of project submissions and user activity.

- **Real-Time Collaboration and Updates :**
  - Real-time collaboration features should enable multiple users to work on projects simultaneously.
  - The system must also provide dynamic updates on project progress, making it easier for students and faculty to stay informed.

- **Seamless Integration of External Resources :**
  - The platform should support linking external resources like YouTube or Google Drive directly to projects, enhancing accessibility.
  - This would allow users to consolidate diverse project materials in a unified interface without leaving the portal.

- **Comprehensive Reporting Tools for Administrators :**
  - Administrators must have access to advanced tools that enable real-time tracking of project submissions, user activity, and system performance.
  - Visual dashboards, charts, and dynamic reports should help in identifying trends and areas that need improvement for better academic project management.

2.4. Mini Project Contribution

This mini project addresses the gaps in academic project management by creating an Institute Project Repository Portal with the following key features:

- **Prioritizing User Experience :**

  The system offers a simple and clean interface, intuitive for students, faculty, and administrators. Developed using HTML and CSS, the front end ensures responsive design and easy navigation, enhancing project management efficiency.

- **Implementing Robust Security Measures :**

  Secure authentication protocols and data encryption protect user information and academic materials from unauthorized access, ensuring the privacy and safety of data.

- **Centralizing Project Management :**

  All academic projects and materials are centralized within the portal, streamlining access and organization for users. External resources like YouTube and Google Drive can be seamlessly integrated for enhanced sharing and accessibility.

- **Facilitating Real-Time Updates and Collaboration :**

  Users can collaborate on projects in real-time, with instant updates to project details, materials, and transaction histories. This ensures that users can access the latest versions of project files and share them effortlessly.

- **Dynamic Data Fetching and Reporting :**

  The system dynamically extracts data from the SQLite database, offering real-time metrics like project submissions, user activity, and more. Administrators can generate insightful reports to assess user engagement and system performance.

- **Enhancing Security Features :**

  Features like CAPTCHA verification during login and the "Forgot Password" functionality ensure secure system access and further safeguard user data.

Comparative Features Analysis: S.I.E.S Repository Portal vs Yukti Website

Features	S.I.E.S Reposistory Portal	Yukti Website
Login System	• College email login only • Captcha protection • Forgot password via email	• No such restriction (general login) • Captcha protection • Forgot password via email • Verify email id
User Roles	• Student & Teacher	• General stakeholders (students, startups, faculty)
Registration	• Separate for teacher & student • Validates college domain	• More generic, not college-specific
Student Dashboard	• Home, Projects, Profile • Can **add/view Idea, Prototype, Startup** • Modal view for full project info	• View projects, submit ideas • No custom profile/dashboard • view per student
Teacher Dashboard	• Dynamic analytics (pie chart, graphs) • Student Projects Viewer • Evaluation panel with grading • Teacher Profile section	• Project viewer • No personalized evaluation dashboard • No graph-based evaluation insight
Project Submission	• Categorized (Idea, Prototype, Startup) • Modal popup with full detail view • Track previous submissions	• Idea submission • No such detailed category-wise modal view
Visualization	• Real-time graphs and charts • Per-branch stats for teachers	• Static or limited visual data
College Specificity	• Built only for your college • Verifies college email	• Open to multiple institutes

CHAPTER 3
PROPOSED SYSTEM

3.1 Introduction

The proposed Institute Project Repository Portal aims to streamline the management and sharing of academic and research projects within the institute. By integrating modern technologies, this system will provide a centralized, user-friendly platform that enhances the academic experience for students, faculty, and researchers. It is designed to support a secure and efficient environment where users can upload, share, manage, and access various project materials, including documents, code, presentations, videos, and external resources such as YouTube and Google Drive links.

Centralized Project Management: Unlike existing fragmented systems such as Google Drive, GitHub, and Moodle, the proposed system integrates all project-related materials in one platform. This reduces inefficiencies and simplifies project tracking.

User-Friendly Interface: The portal is designed with a simple and intuitive interface, making it accessible for both technical and non-technical users including students, faculty, and administrators. It minimizes the learning curve and improves the overall user experience.

Enhanced Security Features: Robust security measures such as strong user authentication, data encryption, CAPTCHA verification, and access control are implemented to protect sensitive academic materials and personal user data.

Real-Time Collaboration and Updates: The portal supports real-time collaboration, allowing multiple users to work on the same project simultaneously. This eliminates the need for switching between different tools and ensures that users always have access to the most up-to-date information.

Dynamic Data Fetching and Reporting: Administrators can dynamically fetch data, providing real-time metrics and comprehensive reports on project submissions, user activity, and system performance. This enables efficient project management and decision-making.

Seamless Integration of External Resources: The system allows users to integrate external resources like YouTube videos and Google Drive links seamlessly into their project materials, enriching the project content without leaving the platform.

3.2 Architecture/Framework

The proposed system will be built on the following architecture and framework:
Client-Side (Frontend): HTML and CSS are used for developing the user interface, ensuring the system is responsive and easy to navigate. JavaScript is used for interactive elements and real-time updates on the portal.
Server-Side (Backend): Python with the Django framework handles backend logic, including user authentication, data management, and real-time updates. SQLite is used as the database to store user information, project details, and related materials securely. It is lightweight and scalable for a growing number of users and projects. Django ORM allows efficient querying of the database for generating dynamic data and reports.
Security: User authentication requires users to log in securely using a password, with email validation limited to gst.sies.edu.in addresses and CAPTCHA verification to prevent automated access. All sensitive data, including user details and project materials, is encrypted to ensure privacy. Different roles such as students, faculty, and administrators are assigned appropriate permissions to manage, view, and edit project data.
External Resource Integration: The Google Drive API allows users to link and share files stored on Google Drive within the portal. The YouTube API enables users to embed video resources directly into their projects.

3.3 Algorithm and Process Design

User Authentication
User registration and login
Algorithm:
User Registration:
Input: User details (name, email, password, mobile number, institute details, etc.)
Output: Confirmation of account creation
Steps:
Validate input fields for completeness and format (e.g., validate email).
Hash the password using a secure hashing algorithm (e.g., bcrypt).
Store the user details in the database with the hashed password.
Send a confirmation email to the user.

User Login:
Input: Email/Username and password
Output: Access to the user dashboard or error message
Steps:
Retrieve user record from the database using the provided email/username.
Compare the hashed password with the stored hash.
If the password matches, generate a session token (JWT) and grant access to the dashboard.
If the password does not match, return an error message.

Algorithm: Actions for Students Post-Login

Authenticate Student Session

After logging in, the system verifies the student's session to confirm active authentication. Once validated, the student is directed to their personalized dashboard.

View Projects

The dashboard displays a "Projects" section where students can see a summary of their submissions. This section includes links to project categories like Idea, Prototype, and Startup.

When the student clicks on a category link, the system retrieves all relevant project data from the database and displays it in an organized table.

If no submissions exist for a category, the dashboard informs the student that no projects have been submitted yet.

Add New Projects

The dashboard includes an "Add Project" button. When clicked, it opens a form allowing students to submit project details such as a title, description, category, and links to external resources like videos or documents.

The system validates all entries to ensure completeness and correctness.

Once validated, the project is saved to the database, and a success message appears, confirming the upload.

View Profile

Students can access their profile section via the dashboard.

This section displays their registered information, such as name, email, and account details.

The profile interface is straightforward, providing a snapshot of their user details.

Logout

A "Logout" option on the dashboard allows students to securely end their session.

The system processes the request by invalidating the session token and redirects the student to the login page with a logout confirmation message.

Algorithm: Actions for Professors Post-Login

Authenticate Teacher Session

After logging in, the system verifies the professor's session to ensure secure access. Once authenticated, the professor is redirected to their dedicated dashboard.

View All Student Projects

On the dashboard, professors can see a "Student Projects" link. Clicking this link fetches all student submissions, including categories like Idea, Prototype, and Startup.

These projects are displayed in a single table for convenience.

Professors can click on any project entry to view detailed information about the submission, such as title, description, and associated files.

Evaluate Student Projects

For each project displayed in the table, an "Evaluate" button is available.

When professors click the "Evaluate" button, an evaluation interface opens where they can assign marks to the respective project.

After submitting the evaluation, the system saves the marks to the database and confirms the successful submission with a notification.

Logout

Professors can securely end their session by clicking the "Logout" button on the dashboard.
The system processes the logout request, clears the session, and redirects the professor to the login page, displaying a "Successfully logged out" message.

3.4 Details of Hardware & Software

Developer Requirements:

Interpreters: Python (Django Framework)
Software: Django, HTML, CSS, JavaScript
Databases: SQLite
Hardware: 10GB storage, 4GB RAM

CHAPTER 4

DESIGN AND METHODOLOGY

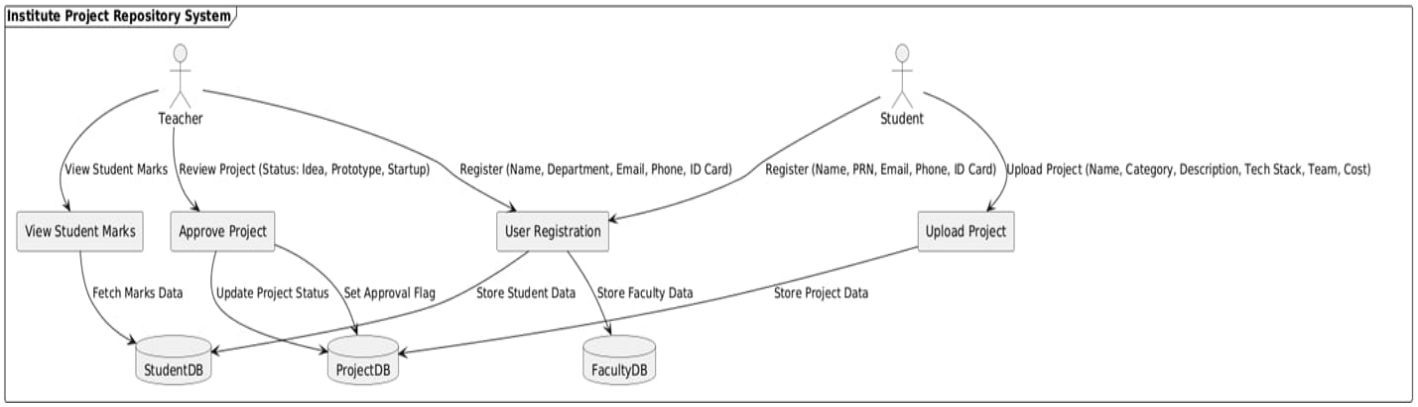


Figure1-User Flow Diagram

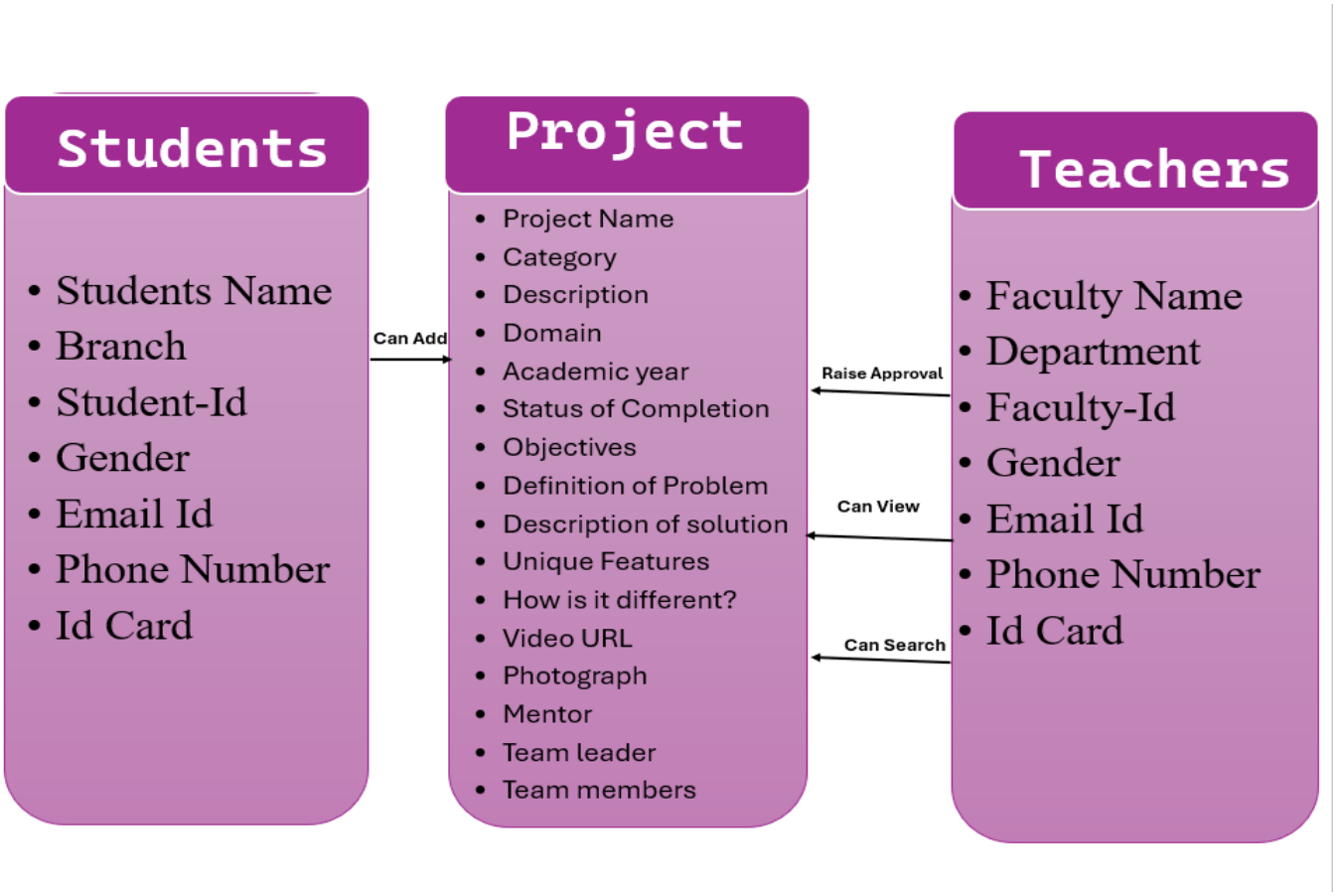


figure-2 DFD Diagram

4.2 Methodology

Methodology Overview

The Agile methodology is adopted for the development of the Institute Project Repository Portal. This approach supports iterative development and continuous user feedback. Agile is ideal for adapting to changing academic needs and user requirements. It promotes collaboration between developers, faculty, and students, ensuring that features evolve based on real-world usage and feedback.

- **Requirement Analysis**
- Stakeholders include students and researchers who upload, view, and manage academic projects, faculty and admins who review project submissions, manage user roles, and monitor data, and system admins who maintain the platform and ensure secure access control.
- Requirements were gathered through interviews with faculty and students to understand current project management practices, surveys to collect feedback from departments about project-sharing needs, and observation of current submission and storage methods such as email and Google Drive.
- Use cases include project submission by students, multimedia uploads including documents, presentations, and videos, integration of external links like YouTube and Google Drive, search functionality by department, year, and tags, admin approval of projects, password reset via email, and CAPTCHA-protected login for security.

- **System Design**

  - The system uses a Model-View-Controller (MVC) pattern provided by the Django framework. The Model manages project data, user profiles, roles, and metadata. The View renders the frontend using HTML/CSS templates for dynamic content. The Controller, implemented as view functions, handles logic for uploads, search, and role-based access.
  - The database includes entities such as User, Project, Department, Media, and Tags. Example attributes include email, name, PRN, role, and branch for the User entity, and title, description, upload date, status, and media links for the Project entity.

  - The interface is designed to be clean and responsive for both desktop and mobile devices. It features consistent buttons, form fields, and branding aligned with the institute’s identity.

Upload interfaces support drag-and-drop functionality and file previews.

- **Implementation**
    - The backend is developed using Python with Django, while the frontend is built using HTML, CSS, and Django templating. SQLite is used as the database due to its lightweight and scalable nature, making it suitable for institutional use.
    - Coding standards follow PEP 8 for Python, with descriptive variable and function names, and modular code structure for views, models, and forms.
    - Modules include user authentication with registration, login using CAPTCHA, and role-based dashboards; project upload and management with support for media files and external links; search and browse functionality with filters for department, project type, and tags; admin tools for viewing and approving or rejecting submissions; and a reporting dashboard displaying project counts, statistics, and charts using dynamic data.
    - All development is tracked using Git, with branches for feature development and bug fixes.

- **Testing**
    - Testing types include unit testing using Django's built-in test framework for models and views, integration testing to ensure smooth user flows like submission and approval, and system testing to confirm end-to-end functionality from login to project submission and search.
    - Test cases cover user registration with valid and invalid institute emails, CAPTCHA validation, password reset, project uploads with and without multimedia, and accuracy of search results.
    - Bug tracking is managed through collaborative tools like GitHub Issues or Trello.

- **Deployment**
    - User training is provided through video walkthroughs for students and faculty, along with PDF guides for uploading and reviewing projects.
- **Maintenance and Support**

- The maintenance plan includes regular updates to ensure compatibility and performance, monthly application of security patches, and a semester-wise review of feature requests.
- User support is offered through a helpdesk with email support and a dedicated FAQ section covering common issues such as upload limits and link formatting.

4.3 Algorithm Implementation

- **User Authentication**

  Purpose: To securely manage user sessions, role verification, and access.

  Algorithm:

  Registration:

  - Input: Institute email, PRN, password, ID upload.
  - Validate email domain (@gst.sies.edu.in).
  - Hash password with Django's in-built method.
  - Store data in the User table.
  - Output: Confirmation pop-up or error.

  Login:

  - Input: Email, password.
  - Fetch user by email and hashed password.
  - Perform CAPTCHA verification.
  - If valid: redirect to dashboard; else: error message.

- **Project Upload & Management**

  Purpose: To manage project entries with metadata and media.
  Algorithm:

  Upload:

- Input: Project title, description, files, YouTube/Drive links,etc.
- Validate file types and links.
- Store media and metadata.
- Assign project to student.
- Output: Success message with pending approval status.

Approval:

- Admin views pending list.
- Option to approve/reject with feedback.
- Status updated in DB and notification sent.

- **Search and Filter**

Purpose: To allow efficient retrieval of projects.
Algorithm:

Search Projects:

- Input: Department, tags, keywords, year.
- Query database using Django ORM filters.
- Return matching projects.
- Output: Paginated project cards with titles and preview.

- **Password Reset & Security**

Purpose: Allow users to reset forgotten passwords securely.
Algorithm:

Reset Request:

- Input: Registered email.
- Send email with OTP or reset link.
- Verify OTP or token.
- Allow user to set new password.

- Output: Success or failure.

- **Reporting & Analytics**

  Purpose: Display project metrics and statistics visually.

  Algorithm:

  Generate Stats:

  - Fetch total projects, department-wise distribution.
  - Use chart libraries (e.g., Chart.js).
  - Display on admin dashboard.
  - Output: Graphs and summary cards.

CHAPTER 5
RESULT AND DISCUSSION

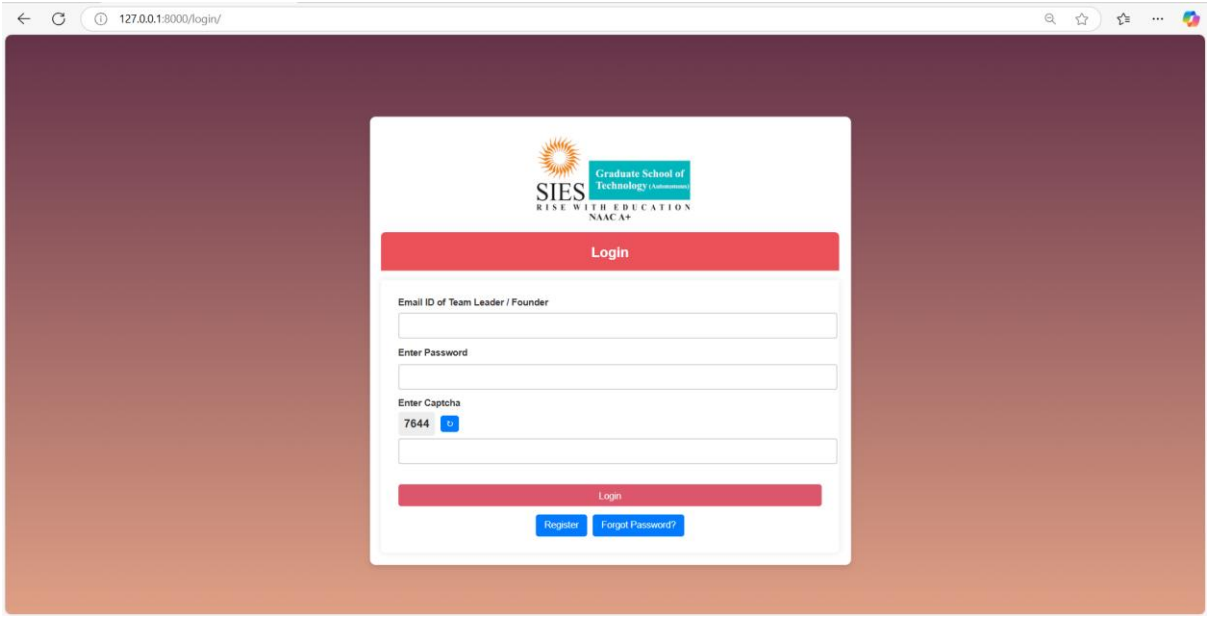
This chapter highlights the implementation, results, and analysis of the Institute Level Project Repository. It outlines how the system meets its core objectives, showcasing key features such as project submission, teacher evaluation, project filtering, and user-specific dashboards. Screenshots of the Teacher Dashboard, Student Dashboard, and submission interfaces demonstrate the design, functionality, and user experience of the system. The implementation effectively uses modern web technologies to ensure a smooth and responsive interface for both students and faculty.

The portal allows students to submit their work categorized into ideas, prototypes, or startups. Only designated team leaders or members are authorized to log in, ensuring secure and organized project submission. Teachers, on the other hand, have access to a dashboard displaying the number of projects and students, as well as evaluation graphs that provide quick insights into academic progress. The ability to filter projects based on branch and academic year enhances accessibility and improves usability.

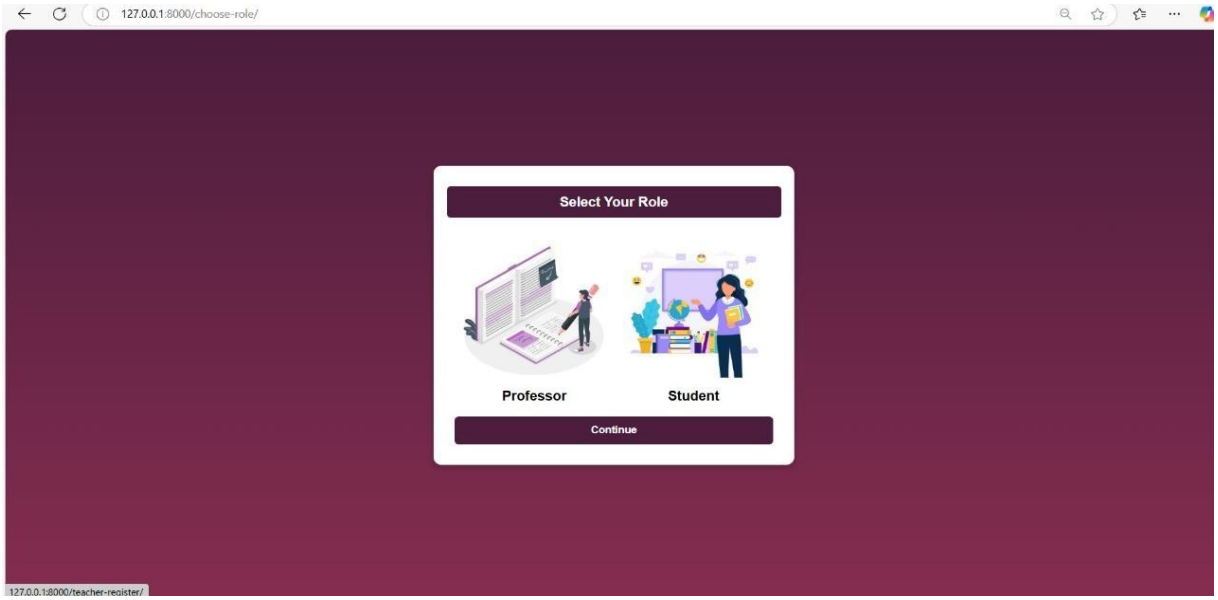
The analysis focuses on user experience, highlighting the simplicity and clarity of navigation for both roles. System performance is evaluated in terms of data handling, responsiveness, and the accuracy of submitted and evaluated data. Compared with traditional manual or semi digital approaches, this portal significantly improves workflow efficiency, transparency, and scalability. Future improvements may include adding real-time notifications, feedback modules, or integration with institutional databases for even broader functionality.

5.1 Implementation

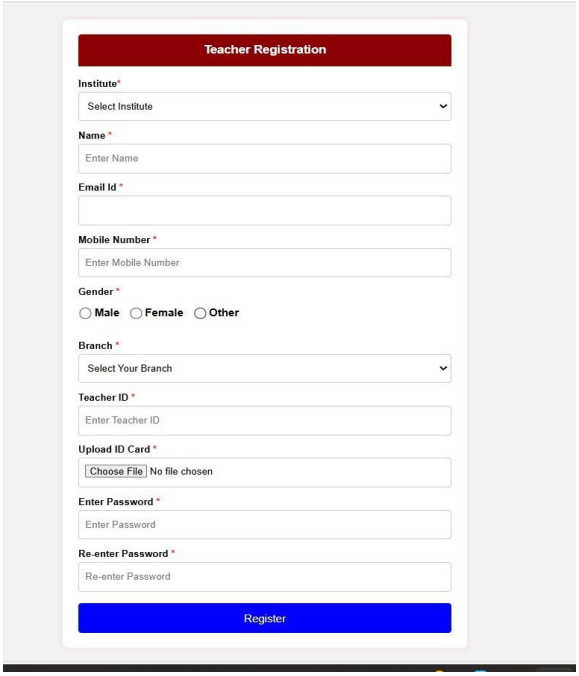
Login Page



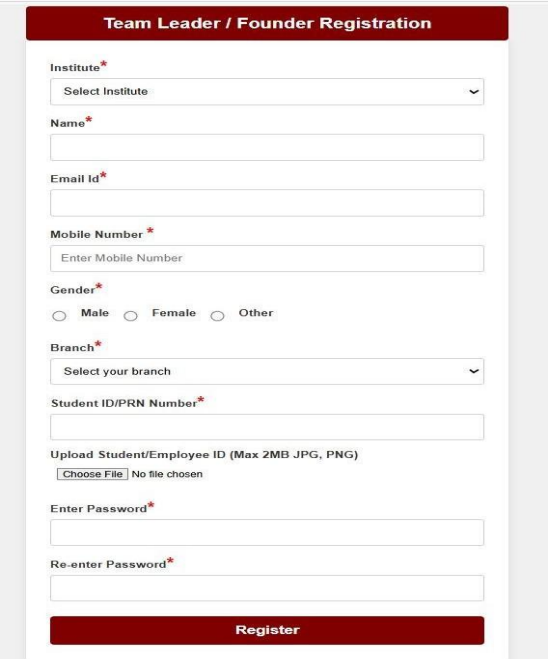
Registration Page



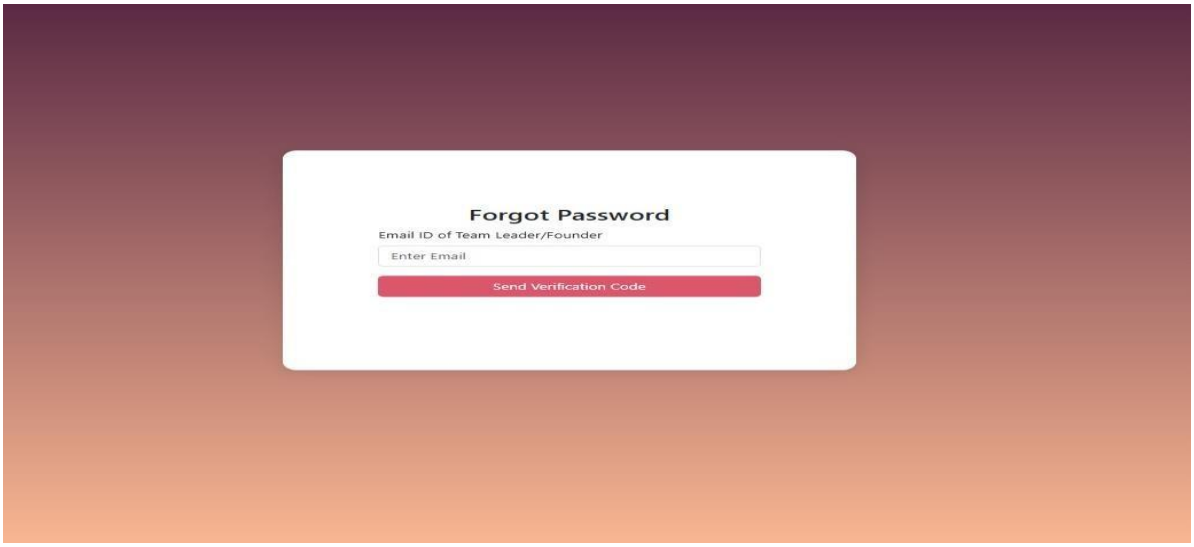
For Teacher Registration



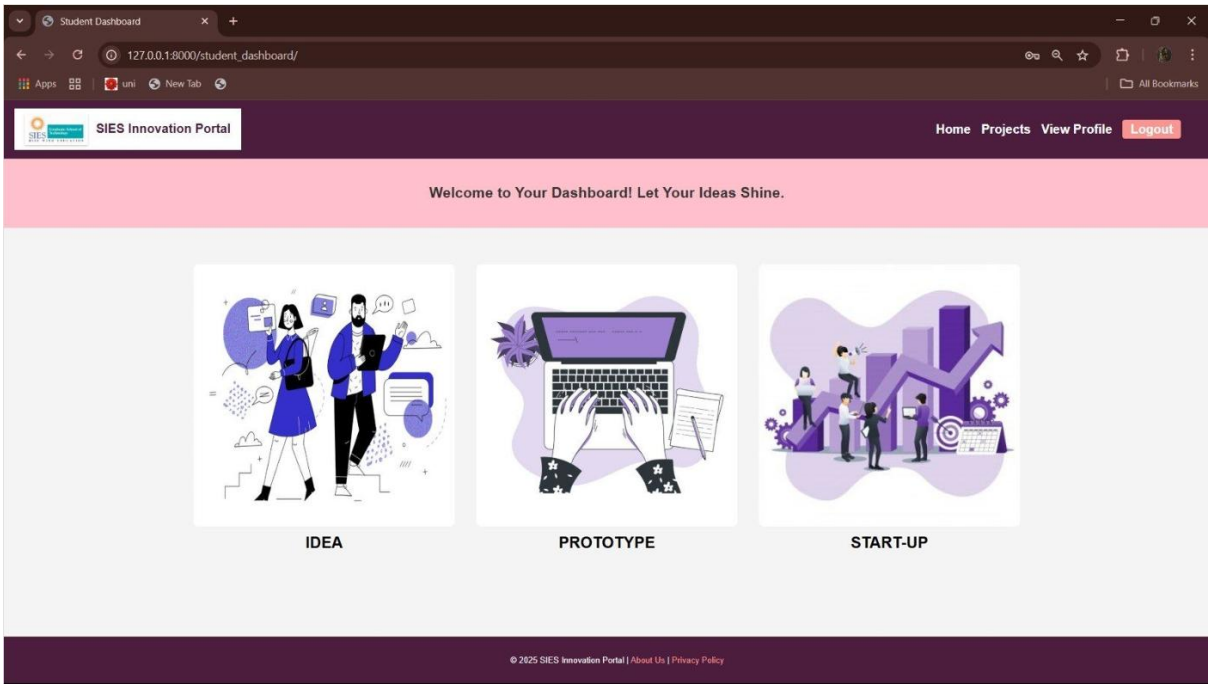
For Student Registration

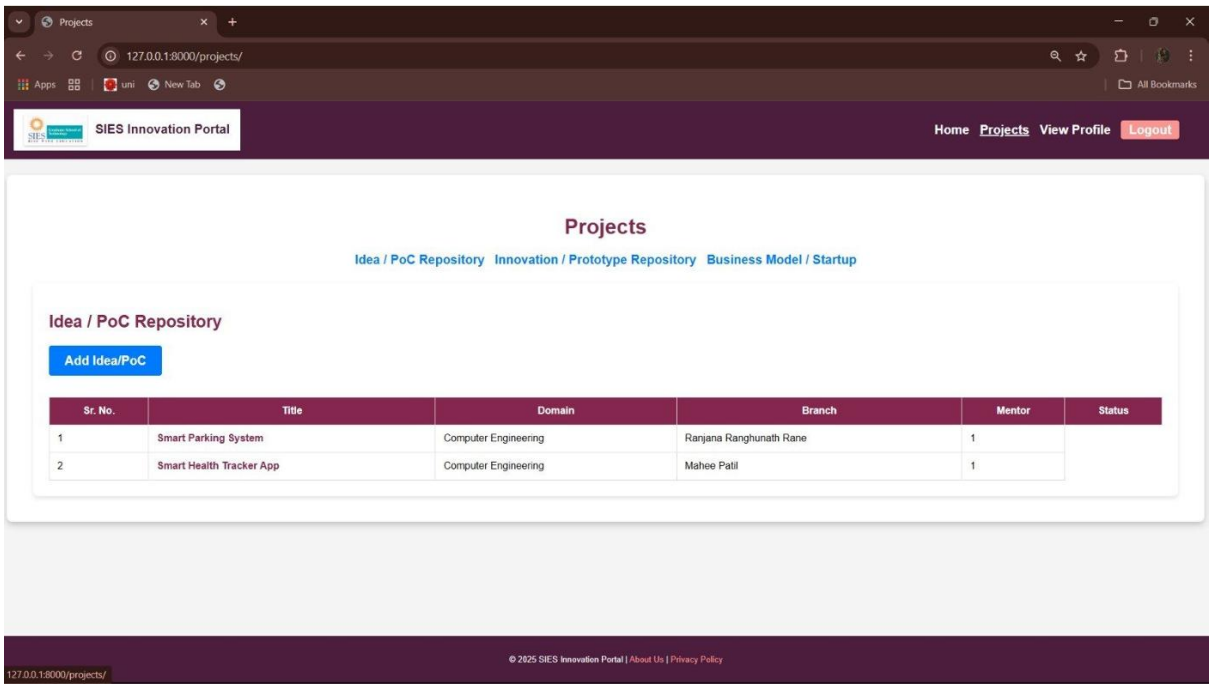


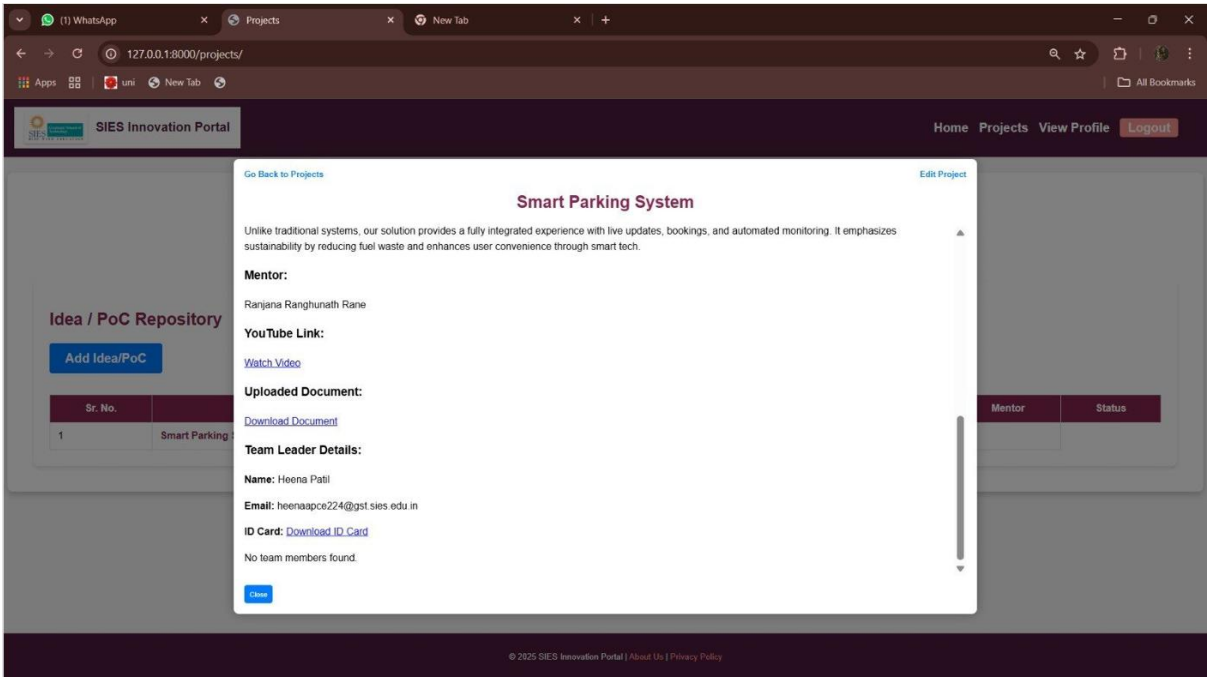
Forgot Password

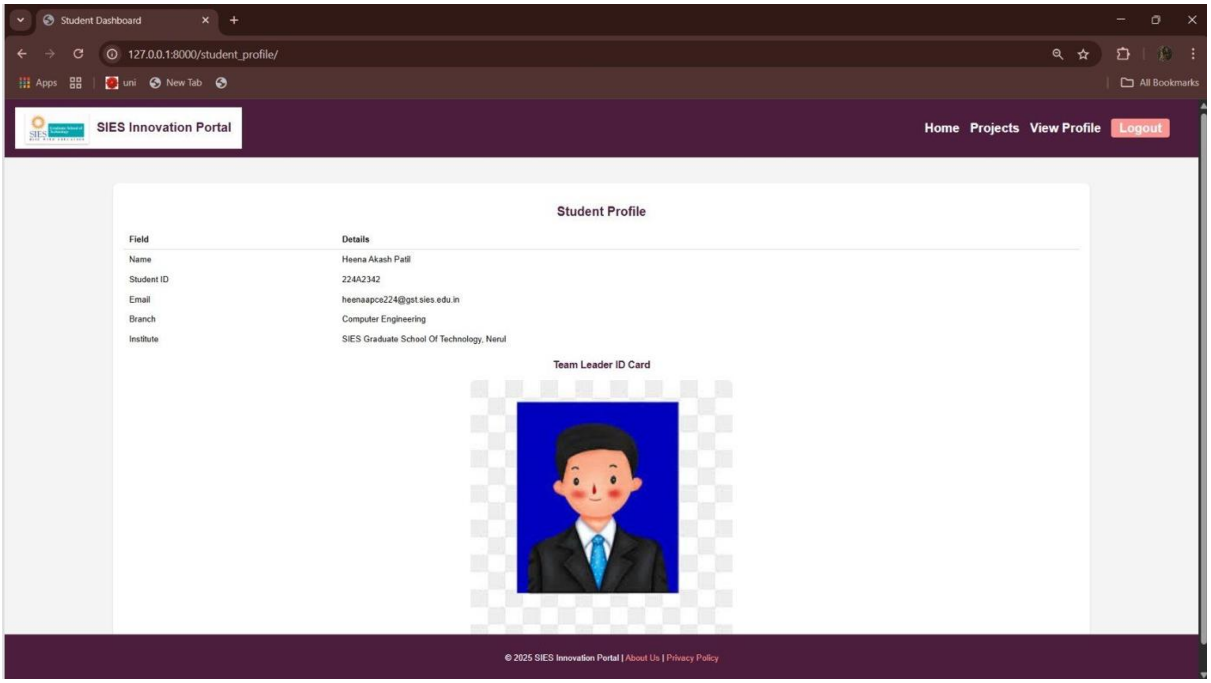


Student Dashboard

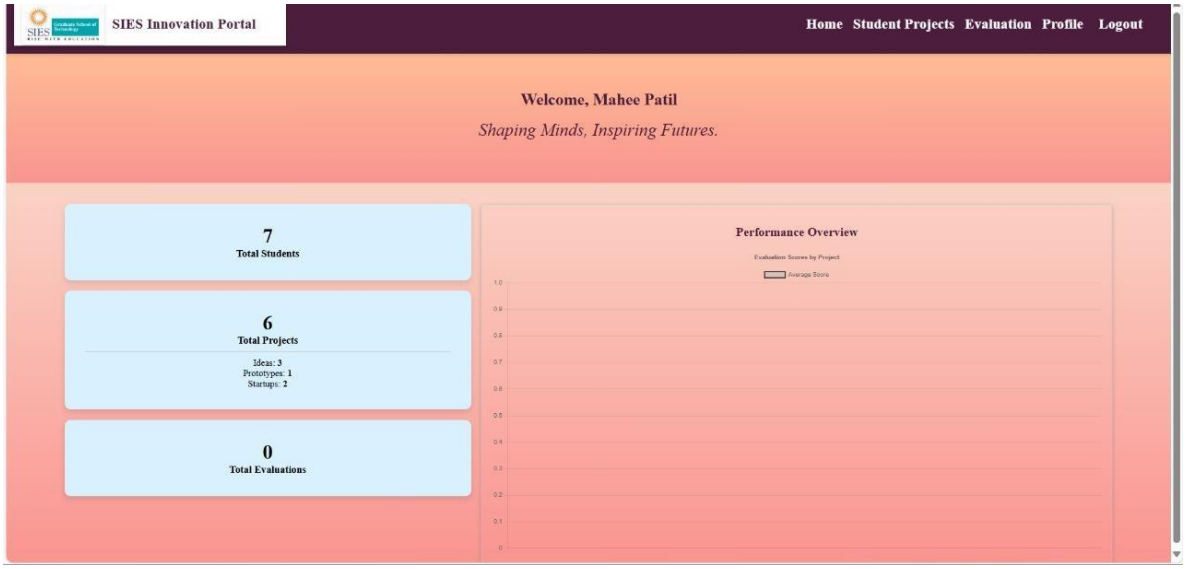


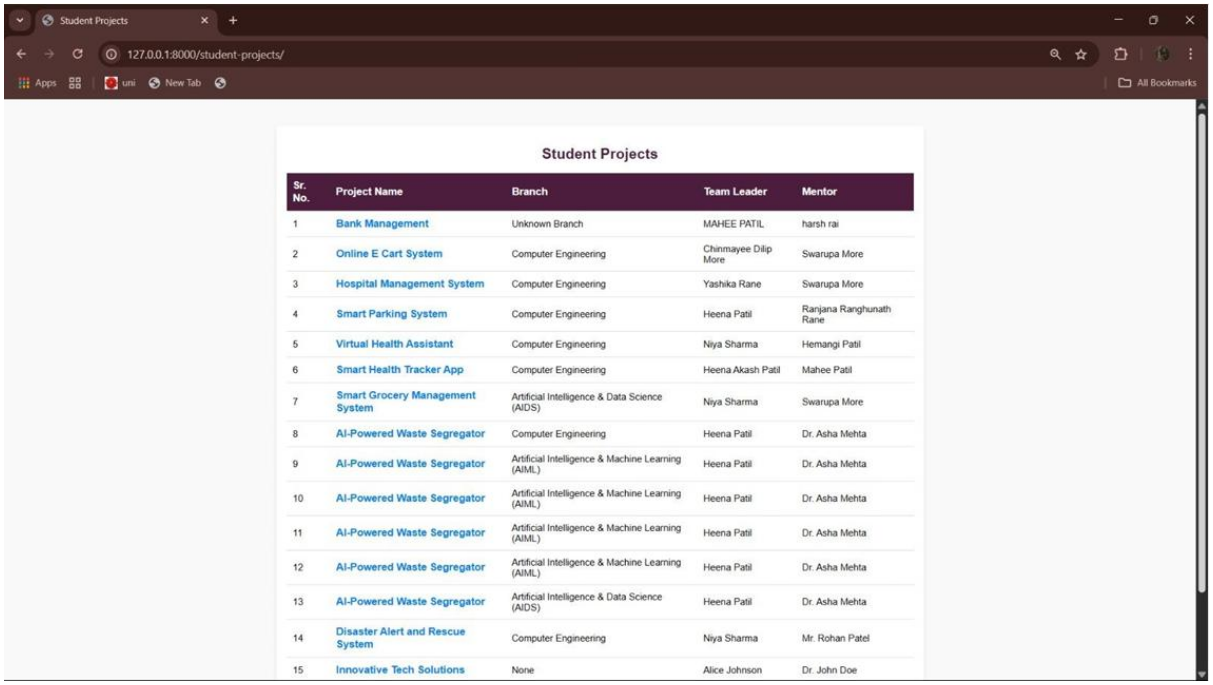



(1) WhatsApp
Projects
New Tab
127.0.0.1:8000/projects/
Apps
uni
New Tab
All Bookmarks
SIES Innovation Portal
Home
Projects
View Profile
Logout
Go Back to Projects
Edit Project
Smart Parking System
Unlike traditional systems, our solution provides a fully integrated experience with live updates, bookings, and automated monitoring. It emphasizes sustainability by reducing fuel waste and enhances user convenience through smart tech.
Mentor:
Ranjana Ranghunath Rane
YouTube Link:
Watch Video
Uploaded Document:
Download Document
Team Leader Details:
Name: Heena Patil
Email: heenaapce224@gst.sies.edu.in
ID Card: Download ID Card
No team members found.
Close
Idea / PoC Repository
Add Idea/PoC
Sr. No.
Mentor
Status
1
Smart Parking
© 2025 SIES Innovation Portal | About Us | Privacy Policy

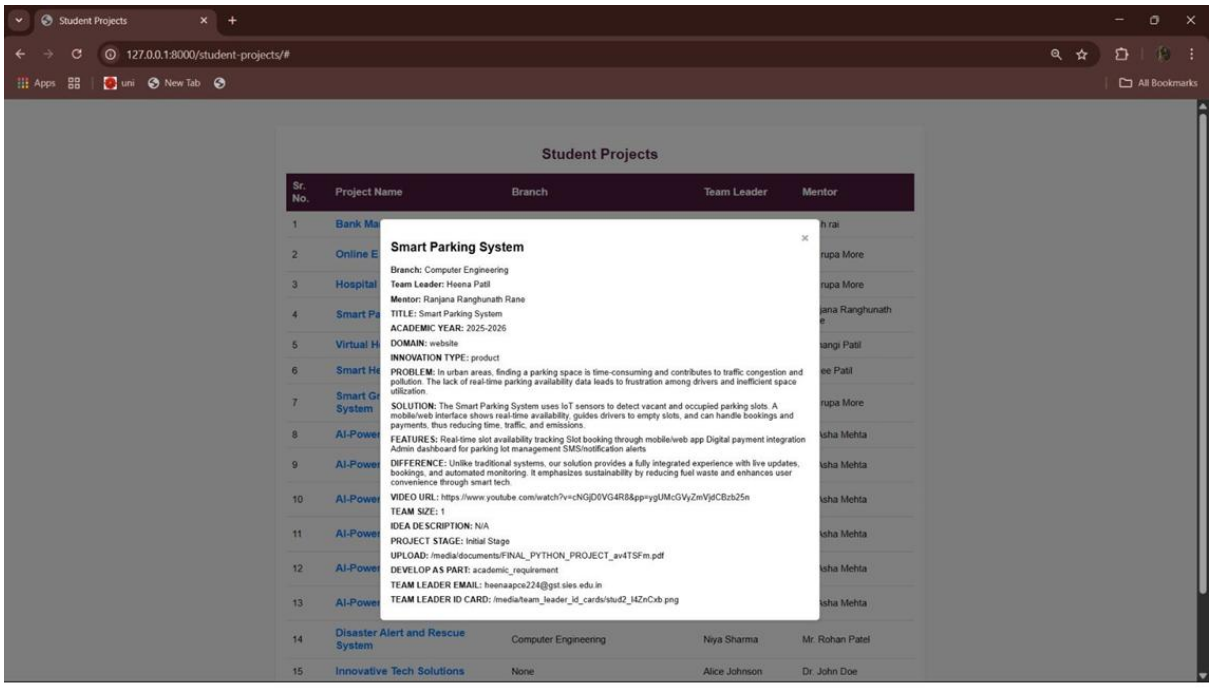

Student Dashboard
127.0.0.1:8000/student_profile/
Apps
uni
New Tab
All Bookmarks
SIES Innovation Portal
Home
Projects
View Profile
Logout
Student Profile
Field
Details
Name
Heena Akash Patil
Student ID
224A2342
Email
heenaapce224@gst.sies.edu.in
Branch
Computer Engineering
Institute
SIES Graduate School Of Technology, Nerul
Team Leader ID Card
© 2025 SIES Innovation Portal | About Us | Privacy Policy

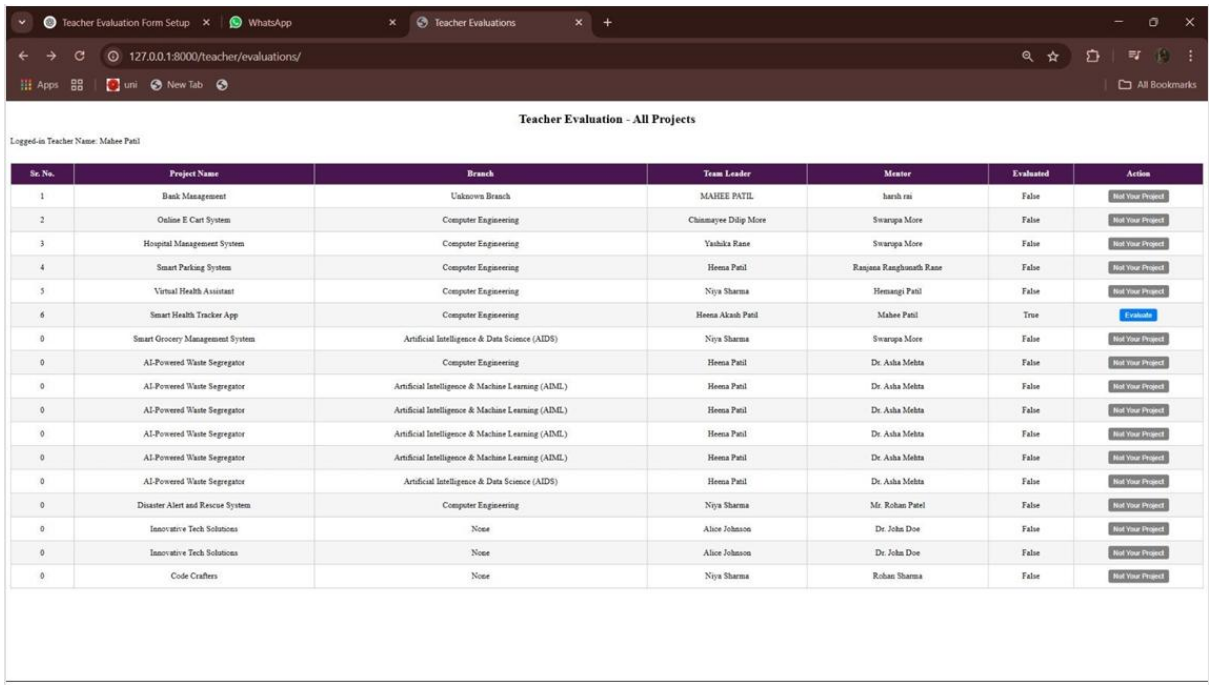
Teacher Dashboard

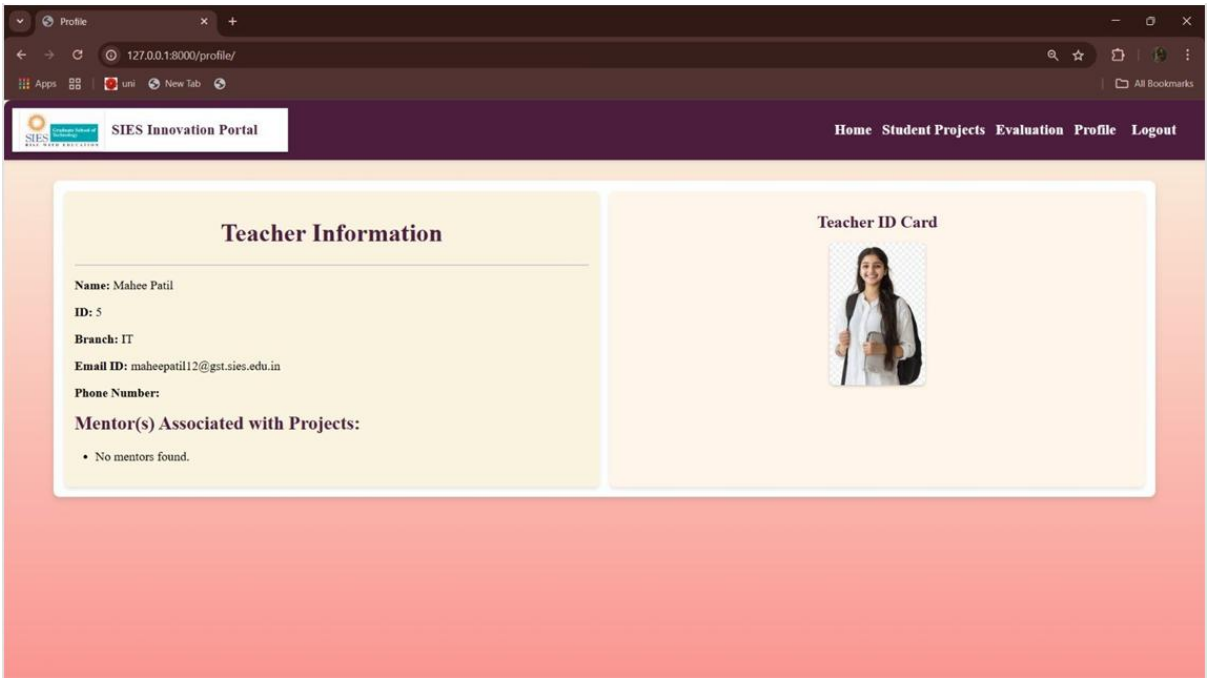




Sr. No.	Project Name	Branch	Team Leader	Mentor
1	Bank Management	Unknown Branch	MAHEE PATIL	harsh rai
2	Online E Cart System	Computer Engineering	Chinmayee Dilip More	Swarupa More
3	Hospital Management System	Computer Engineering	Yashika Rane	Swarupa More
4	Smart Parking System	Computer Engineering	Heena Patil	Ranjana Ranghunath Rane
5	Virtual Health Assistant	Computer Engineering	Niya Sharma	Hemangi Patil
6	Smart Health Tracker App	Computer Engineering	Heena Akash Patil	Mahee Patil
7	Smart Grocery Management System	Artificial Intelligence & Data Science (AIDS)	Niya Sharma	Swarupa More
8	AI-Powered Waste Segregator	Computer Engineering	Heena Patil	Dr. Asha Mehta
9	AI-Powered Waste Segregator	Artificial Intelligence & Machine Learning (AIML)	Heena Patil	Dr. Asha Mehta
10	AI-Powered Waste Segregator	Artificial Intelligence & Machine Learning (AIML)	Heena Patil	Dr. Asha Mehta
11	AI-Powered Waste Segregator	Artificial Intelligence & Machine Learning (AIML)	Heena Patil	Dr. Asha Mehta
12	AI-Powered Waste Segregator	Artificial Intelligence & Machine Learning (AIML)	Heena Patil	Dr. Asha Mehta
13	AI-Powered Waste Segregator	Artificial Intelligence & Data Science (AIDS)	Heena Patil	Dr. Asha Mehta
14	Disaster Alert and Rescue System	Computer Engineering	Niya Sharma	Mr. Rohan Patel
15	Innovative Tech Solutions	None	Alice Johnson	Dr. John Doe


Student Projects
127.0.0.1:8000/student-projects/#
Apps
uni
New Tab
All Bookmarks
Student Projects
Sr. No.
Project Name
Branch
Team Leader
Mentor
Smart Parking System
Branch: Computer Engineering
Team Leader: Heena Patil
Mentor: Ranjana Ranghunath Rane
TITLE: Smart Parking System
ACADEMIC YEAR: 2025-2026
DOMAIN: website
INNOVATION TYPE: product
PROBLEM: In urban areas, finding a parking space is time-consuming and contributes to traffic congestion and pollution. The lack of real-time parking availability data leads to frustration among drivers and inefficient space utilization.
SOLUTION: The Smart Parking System uses IoT sensors to detect vacant and occupied parking slots. A mobile/web interface shows real-time availability, guides drivers to empty slots, and can handle bookings and payments, thus reducing time, traffic, and emissions.
FEATURES: Real-time slot availability tracking Slot booking through mobile/web app Digital payment integration Admin dashboard for parking lot management SMS/notification alerts
DIFFERENCE: Unlike traditional systems, our solution provides a fully integrated experience with live updates, bookings, and automated monitoring. It emphasizes sustainability by reducing fuel waste and enhances user convenience through smart tech.
VIDEO URL: https://www.youtube.com/watch?v=cNGjD0VG4R8&pp=ygUMcGVyZmVjdCBzb25n
TEAM SIZE: 1
IDEA DESCRIPTION: N/A
PROJECT STAGE: Initial Stage
UPLOAD: /media/documents/FINAL_PYTHON_PROJECT_av4TSFm.pdf
DEVELOP AS PART: academic_requirement
TEAM LEADER EMAIL: heenaapce224@gst.sies.edu.in
TEAM LEADER ID CARD: /media/team_leader_id_cards/stud2_t4ZnCxb.png
14 Disaster Alert and Rescue System Computer Engineering Niya Sharma Mr. Rohan Patel
15 Innovative Tech Solutions None Alice Johnson Dr. John Doe


Teacher Evaluation Form Setup
WhatsApp
Teacher Evaluations
127.0.0.1:8000/teacher/evaluations/
Apps
uni
New Tab
All Bookmarks
Teacher Evaluation - All Projects
Logged-in Teacher Name: Mahee Patil
Sr. No. | Project Name | Branch | Team Leader | Mentor | Evaluated | Action
1 | Bank Management | Unknown Branch | MAHEE PATIL | harsh rai | False | Not Your Project
2 | Online E Cart System | Computer Engineering | Chinmayee Dilip More | Swarupa More | False | Not Your Project
3 | Hospital Management System | Computer Engineering | Yashika Rane | Swarupa More | False | Not Your Project
4 | Smart Parking System | Computer Engineering | Heena Patil | Ranjana Ranghunath Rane | False | Not Your Project
5 | Virtual Health Assistant | Computer Engineering | Niya Sharma | Hemangi Patil | False | Not Your Project
6 | Smart Health Tracker App | Computer Engineering | Heena Akash Patil | Mahee Patil | True | Evaluate
0 | Smart Grocery Management System | Artificial Intelligence & Data Science (AIDS) | Niya Sharma | Swarupa More | False | Not Your Project
0 | AI-Powered Waste Segregator | Computer Engineering | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | AI-Powered Waste Segregator | Artificial Intelligence & Machine Learning (AIML) | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | AI-Powered Waste Segregator | Artificial Intelligence & Machine Learning (AIML) | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | AI-Powered Waste Segregator | Artificial Intelligence & Machine Learning (AIML) | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | AI-Powered Waste Segregator | Artificial Intelligence & Machine Learning (AIML) | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | AI-Powered Waste Segregator | Artificial Intelligence & Data Science (AIDS) | Heena Patil | Dr. Asha Mehta | False | Not Your Project
0 | Disaster Alert and Rescue System | Computer Engineering | Niya Sharma | Mr. Rohan Patel | False | Not Your Project
0 | Innovative Tech Solutions | None | Alice Johnson | Dr. John Doe | False | Not Your Project
0 | Innovative Tech Solutions | None | Alice Johnson | Dr. John Doe | False | Not Your Project
0 | Code Crafters | None | Niya Sharma | Rohan Sharma | False | Not Your Project



5.2 Result And Discussion

Overview of Implementation

The implementation of the Project Repository Portal follows a structured and user- focused approach to managing academic project workflows between students and teachers. Key functionalities such as user authentication, project submission, project evaluation, and data visualization have been developed to meet the requirements defined in the design phase. The system leverages modern web technologies along with a relational database to ensure reliable data handling and a seamless user experience. The development process followed Agile methodology, allowing iterative improvements and integration of user feedback at various stages.

User Authentication

The user authentication module enables secure login functionality for both students and teachers. Each team is represented by a designated team leader or member, who has the authority to log in and submit projects on behalf of their group. Authentication is handled with secure password hashing and validation techniques to ensure privacy and protect user credentials.

During testing, the authentication module successfully handled different scenarios, such as invalid login attempts, duplicate registrations, and incorrect password entries. Login and registration

processes responded efficiently, with average response times of under two seconds, ensuring a smooth user experience for both roles.

Project Submission Management

The student-side submission module enables users to submit projects under clearly defined categories: **Idea**, **Prototype**, or **Startup**. The intuitive tabbed interface helps users organize and manage their submissions easily. Each project includes essential metadata and is stored securely in the backend for evaluation.

Testing confirmed that project submissions were successfully recorded, validated, and retrievable based on category. Students appreciated the ability to track their submissions in a categorized view. Suggestions from early user testing included adding version tracking or editable drafts—features that may be considered in future iterations.

Teacher Dashboard and Evaluation

The teacher dashboard is a central feature that allows faculty members to view the total number of submitted projects, number of students, and access evaluation graphs. Teachers can review individual projects and add assessments or evaluations as needed. The system also provides branch-wise and academic-year-wise filters, allowing teachers to easily locate specific submissions.

This feature was tested with multiple data sets and proved to be both functional and efficient. The evaluation graph component, built using dynamic charting libraries, successfully displayed real-time insights based on the uploaded data. Teachers reported that the dashboard was easy to use and appreciated the ability to filter projects for quicker review.

Filtering and Project Discovery

The filtering system allows teachers to filter submitted projects by **branch**, **academic year**, and **submission category** (idea, prototype, startup). This was implemented using a combination of frontend filters and backend query optimization to ensure responsiveness even with large data sets.

During testing, filters worked as expected and returned accurate results without noticeable lag. Suggestions from testers included adding keyword-based search and tag filters, which could be introduced in future updates to improve searchability.

Reporting and Data Visualization

The evaluation graphs offer a visual representation of student project data—helping teachers quickly assess trends and outcomes. Charts dynamically display metrics like the number of submissions per category or per branch.

These visual components were responsive and loaded quickly during performance testing. Teachers noted that the visual format made data interpretation easier compared to traditional lists or tables. However, it was noted that exporting data or charts in PDF/Excel formats would further enhance usability—this remains a potential feature for future development.

5.3 Challenges Faced

Throughout the development of the Student Project Management Portal, several challenges arose that required thoughtful problem-solving and adaptation. A primary challenge was managing **user roles and access control**, especially ensuring that only authorized team members or team leaders could log in and submit projects on behalf of their group. Implementing this securely required custom authentication logic and strict user role validation to maintain data integrity and prevent unauthorized access.

Another significant challenge was **designing an interface** that served the needs of both students and teachers while remaining intuitive and responsive. Aligning the frontend with backend logic, particularly for features like dynamic filtering, project categorization, and evaluation graphs, required careful synchronization to ensure a smooth user experience.

The **scope of the project** also presented hurdles during the development phase. As feedback was collected from potential users and stakeholders, new feature ideas—such as real-time notifications, editable submissions, and exportable evaluation reports—emerged. Balancing these ideas with time and resource limitations was crucial. As a result, the team focused on delivering core functionalities first, while structuring the system to support future enhancements without major architectural changes.

Lastly, ensuring **performance and scalability** was an ongoing concern. As the number of submissions and users increased during testing, optimizations in database queries and front-

end rendering became necessary to maintain responsiveness. These efforts laid a strong foundation for the platform's future growth and adaptability.

5.4 Conclusion

The implementation of the Student Project Management Portal has been a successful initiative, demonstrating the effectiveness of structured planning, iterative development, and user-centered design. The system effectively meets its core objectives by streamlining project submission for students and simplifying evaluation processes for teachers. Key functionalities—such as secure team-based access, categorized submissions, project filtering, and graphical evaluations—enhance usability and transparency within the academic workflow.

User feedback and testing confirmed that the portal offers a responsive and intuitive experience for both students and faculty members. It also lays a solid foundation for future upgrades, with scalable architecture and well-defined modules. Overall, this project not only addresses the immediate requirements of academic project management but also establishes a platform capable of adapting to future educational needs and technological advancements.

5.5 Future Work

Looking forward, future development will focus on further improving performance and expanding features based on user and stakeholder feedback. Planned enhancements include **real-time notifications**, **submission editing capabilities**, and **export options** for reports and evaluations. Integration with institutional databases or Learning Management Systems (LMS) may also be explored to streamline academic processes.

Additionally, improving the responsiveness of the platform for mobile and tablet devices will be a priority to increase accessibility. Advanced features such as **feedback modules**, **version tracking**, and **submission status alerts** are also under consideration. Continuous collaboration with end-users will ensure the platform evolves in alignment with academic standards and user exception.