

**STUDENT PROJECT MANAGEMENT PORTAL
A MINI-PROJECT REPORT
(DATABASE MANAGEMENT SYSTEM)**

Submitted by

NIVEDITHAAS

241901076

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

**COMPUTER SCIENCE AND ENGINEERING
(CYBERSECURITY)**



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

**CHENNAI
NOVEMBER 2025**

BONAFIDE CERTIFICATE

Certified that this project “**STUDENT PROJECT MANAGEMENT PORTAL**” is the bonafide work of “**NIVEDITHAA S**” who carried out the project work under my supervision.

SIGNATURE

Ms. Fowzia Sihana .B

ASSISTANT PROFESSOR (SS)

Dept. of Computer Science and Engineering
(Cyber Security),
Rajalakshmi Engineering College,
Chennai.

This mini project report is submitted for the viva voce examination to be held on

_____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

“SkillConnect” is a web-based project management platform designed exclusively for students to upload, organize, manage, and showcase their academic projects. In most academic institutions, project documentation and approval processes are handled manually, leading to inefficiency, missing records, and lack of visibility of student work.

SkillConnect solves this issue by providing a centralized platform where students can maintain profiles, upload project details, submit files, and share links related to their work. At the same time, administrators can view submitted projects, approve them, manage student lists, and generate digital certificates for approved projects.

The system is built using a Java frontend, Flask (Python) backend, and MySQL database, ensuring reliability, scalability, and ease of integration. The platform also includes a visually rich dashboard where students can explore projects uploaded by others, creating a collaborative environment and promoting peer learning.

SkillConnect eliminates manual paperwork, increases transparency in academic project evaluation, and enhances accessibility for both students and faculty, making it a modern solution for academic project management.

ACKNOWLEDGEMENT

I express my sincere thanks to my beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

I am greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance. No words of gratitude will suffice for the unquestioning support extended to me by my Head Of The Department **Mr. Benedict JN** and my Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

I also extend my sincere and hearty thanks to my internal guide **Ms. Fowzia Sihana .B,** for her valuable guidance and motivation during the completion of this project.

My sincere thanks to my family members, friends and other staff members of Computer Science and Engineering (Cybersecurity).

NIVEDITHAA S

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO
	ABSTRACT	iv
1	INTRODUCTION	1
1.1	INTRODUCTION	8
1.2	SCOPE OF THE WORK	8
1.3	PROBLEM STATEMENT	8
1.4	AIM AND OBJECTIVES OF THE PROJECT	8
2	SYSTEM SPECIFICATIONS	9
2.1	HARDWARE SPECIFICATIONS	9
2.2	SOFTWARE SPECIFICATIONS	9
3	MODULE DESCRIPTION	10
4	CODING	11
5	SCREENSHOTS	16
6	CONCLUSION AND FUTURE ENHANCEMENT	18
	REFERENCES	19

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
5.1	USER REGISTRATION PAGE	17
5.2	USER LOGIN DETAILS	17
5.3	STUDENT DASHBOARD	18
5.4	UPLOADING A PROJECT	18
5.5	MY PRJECTS PAGE	19
5.6	ADMIN LOGIN	19
5.7	ADMIN DASHBOARD	20
5.8	PROJECT SHOWCASE	20

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

SkillConnect is a web-based student project management system developed to simplify and digitalize the way students upload and manage their academic projects. Instead of traditional paperwork or scattered submissions, SkillConnect provides a unified platform for project tracking, storage, approval, and certification. It improves the academic workflow, enhances visibility of student work, and enables administrators to evaluate projects efficiently.

1.2 SCOPE OF THE WORK

The scope of SkillConnect includes:

- A dedicated login system for both students and administrators.
- Secure user profile and project management interface.
- Project uploads with media, documentation, links, and technical stack.
- Admin portal for viewing, approving, or rejecting projects.
- Automatic certificate generation for approved submissions.
- Showcase dashboard where all approved student projects can be viewed.
- A scalable and secure backend with MySQL integration.

1.3 PROBLEM STATEMENT

Educational institutions often rely on manual or unorganized methods to collect, verify, and store student project data. This leads to:

- Difficulty in tracking project approvals.
- No centralized history of student project work.
- Delayed certificate generation.
- No platform for students to explore peer projects.
- Administrative overhead and lack of transparency.

SkillConnect aims to eliminate these challenges with a fully automated digital platform.

1.4 AIM AND OBJECTIVES OF THE PROJECT

Aim: To develop an efficient, user-friendly project management platform for students and administrators in an academic environment.

Objectives

- To allow students to upload and manage their projects digitally.
- To enable administrators to easily review and approve projects.
- To store project details in a structured database.
- To generate downloadable certificates upon approval.
- To make student projects publicly viewable through the Showcase dashboard.
- To provide a simple, intuitive interface for both students and admins.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i7
Memory Size	:	8GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 11
Front - End	:	HTML, CSS, JS
Back - End	:	Python
Database	:	SQLite

CHAPTER 3

MODULE DESCRIPTION

The SkillConnect platform consists of two main types of users:

1. Student Module

1.1 Student Login

Students can securely log in using email and password.

1.2 User Dashboard

- View and update profile
- Upload new projects
- View their previous project submissions
- Check approval status
- Download certificates after approval

1.3 Project Upload Module

Fields include:

- Project Name
- Roll Number
- Project Description
- Tech Stack
- Project Link (GitHub/Live Demo)
- Media Upload (images, PDFs, videos)

All uploads are stored in structured MySQL records.

1.4 Showcase Dashboard

A public project listing where students can:

- View others' approved projects
- Search by tech stack or keywords

2. Admin Module

2.1 Admin Login

Authorized academic members can access admin features.

2.2 Project Approval Dashboard

Admins can:

- View all submitted projects
- Approve/reject with feedback
- Auto-generate certificates for approved projects
- View student list and project history

2.3 Certificate Generation

Once approved, the system:

- Generates a PDF certificate template.
- Stores it in the database.
- Displays it in the student's "My Projects" section.

CHAPTER 4

SAMPLE CODE

Sample I:

This code defines a database model for a project submission system using Flask SQLAlchemy. It manages Users, their Projects, and the Certificates issued for approved projects. Each user can submit multiple projects, and each project can have one certificate linked to it. It also tracks important details like tech stack, media files, project status, and creation timestamps.

```
from flask_sqlalchemy import SQLAlchemy

from flask_login import UserMixin

from datetime import datetime


db = SQLAlchemy()


class User(UserMixin, db.Model):

    __tablename__ = 'users'

    id = db.Column(db.Integer, primary_key=True)

    name = db.Column(db.String(200), nullable=False)

    email = db.Column(db.String(200), unique=True, nullable=False)

    roll_number = db.Column(db.String(50), unique=True, nullable=False)

    password = db.Column(db.String(200), nullable=False)

    is_admin = db.Column(db.Boolean, default=False)

    created_at = db.Column(db.DateTime, default=datetime.utcnow)
```

```
projects = db.relationship('Project', backref='user', lazy=True)
```

```
class Project(db.Model):
```

```
    __tablename__ = 'projects'
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    name = db.Column(db.String(200), nullable=False)
```

```
    description = db.Column(db.Text, nullable=False)
```

```
    tech_stack = db.Column(db.String(500), nullable=False)
```

```
    project_link = db.Column(db.String(500))
```

```
    media_file = db.Column(db.String(500))
```

```
    roll_number = db.Column(db.String(50), nullable=False)
```

```
    status = db.Column(db.String(20), default='pending')
```

```
    user_id = db.Column(db.Integer, db.ForeignKey('users.id'), nullable=False)
```

```
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
```

```
    certificate = db.relationship('Certificate', backref='project', uselist=False,  
                                lazy=True)
```

```
class Certificate(db.Model):
```

```
    __tablename__ = 'certificates'
```

```

id = db.Column(db.Integer, primary_key=True)

project_id = db.Column(db.Integer, db.ForeignKey('projects.id'), nullable=False)

student_name = db.Column(db.String(200), nullable=False)

project_name = db.Column(db.String(200), nullable=False)

roll_number = db.Column(db.String(50), nullable=False)

issued_at = db.Column(db.DateTime, default=datetime.utcnow)

```

Sample II

The `upload_project` function receives project data from a POST request and checks if all required fields are filled. It also validates the description and processes an optional media file upload. After saving the file and project details, it commits the data to the database. The user is then notified and redirected to view their projects.

```

def upload_project():
    if request.method == 'POST':

        project_name = request.form.get('project_name')

        description = request.form.get('description')

        tech_stack = request.form.get('tech_stack')

        project_link = request.form.get('project_link')

        if not project_name or not description or not tech_stack:

            flash('Please fill all required fields.', 'error')

            return redirect(url_for('upload_project'))

        if len(description) < 10:

            flash('Description must be at least 10 characters.', 'error')

```

```

return redirect(url_for('upload_project'))

media_filename = None

if 'media_file' in request.files:

    file = request.files['media_file']

    if file and file.filename and allowed_file(file.filename):

        filename = secure_filename(file.filename)

        timestamp = datetime.now().strftime('%Y%m%d%H%M%S')

        media_filename =
f"{current_user.roll_number}_{timestamp}_{filename}"

        file.save(os.path.join(app.config['UPLOAD_FOLDER'], media_filename))

new_project = Project(

    name=project_name,

    description=description,

    tech_stack=tech_stack,

    project_link=project_link,

    media_file=media_filename,

    user_id=current_user.id,

    roll_number=current_user.roll_number

)

db.session.add(new_project)

```

```
db.session.commit()
```

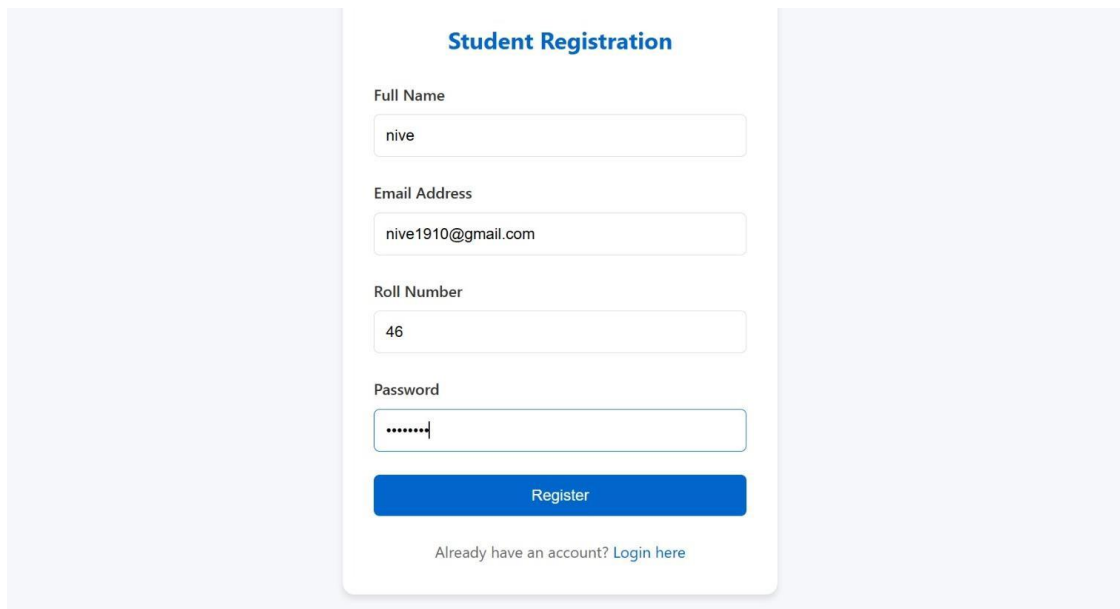
```
flash('Project uploaded successfully!', 'success')
```

```
return redirect(url_for('my_projects'))
```

```
return render_template('upload_project.html')
```

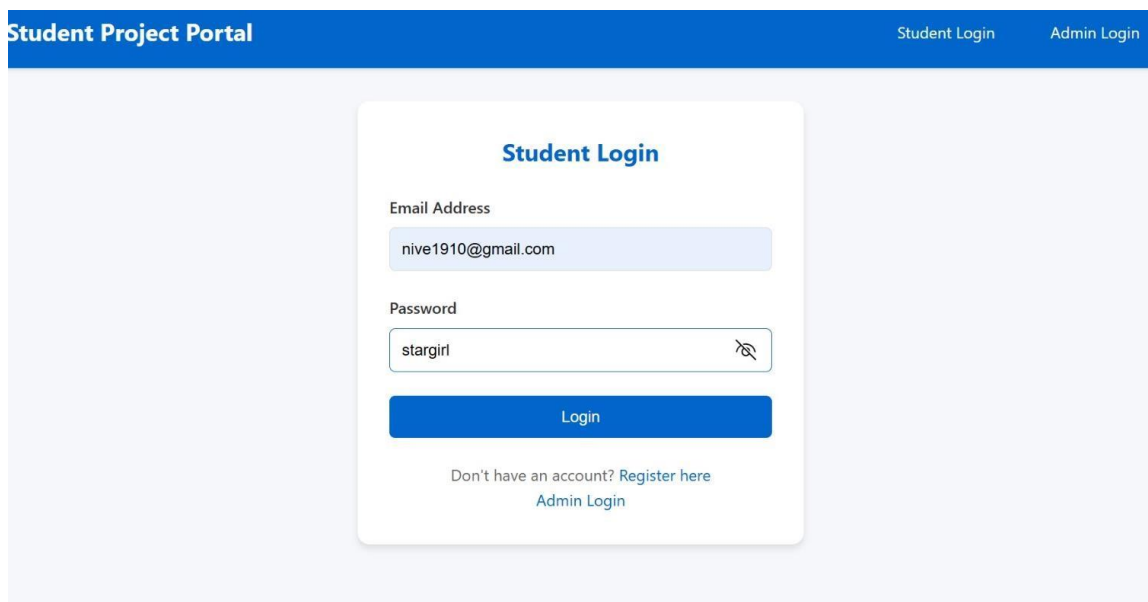

CHAPTER 5

SCREENSHOTS



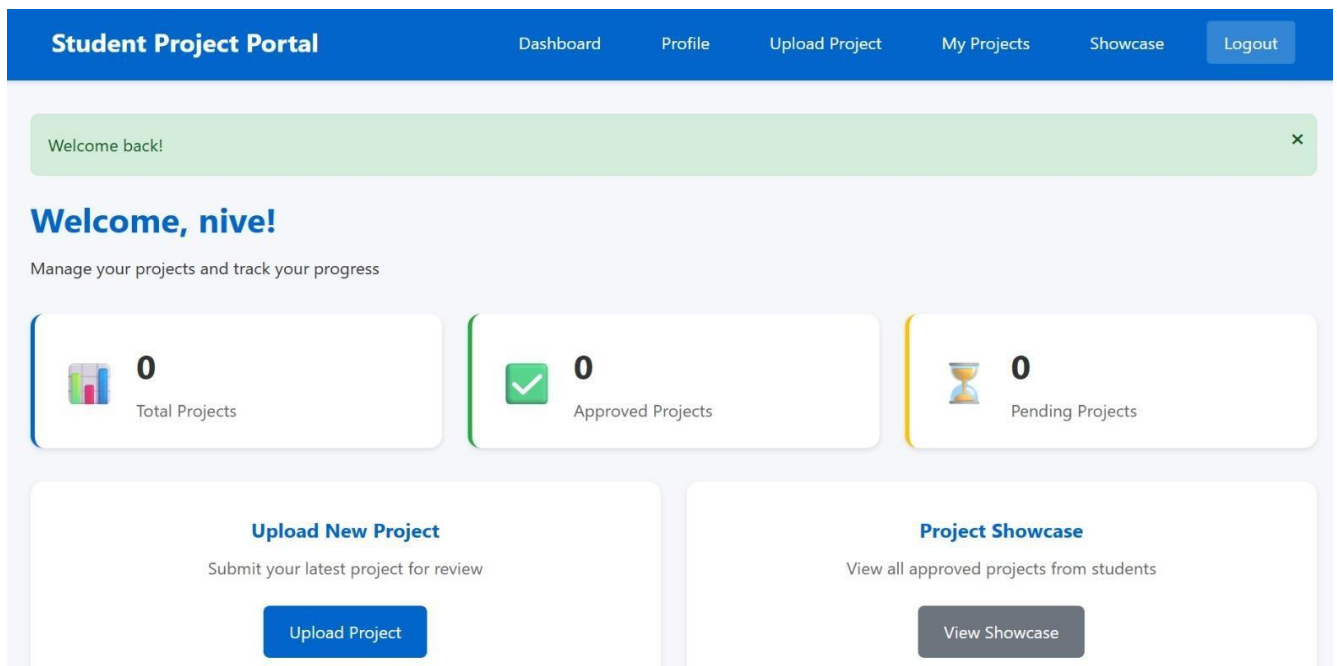
The screenshot shows a 'Student Registration' form. It has a title 'Student Registration' in blue. Below the title are four input fields: 'Full Name' with the value 'nive', 'Email Address' with the value 'nive1910@gmail.com', 'Roll Number' with the value '46', and 'Password' with masked characters '.....'. A blue 'Register' button is at the bottom of the form. Below the button is a link: 'Already have an account? [Login here](#)'.

Fig 5.1 User Registration page



The screenshot shows a 'Student Login' page. At the top is a blue header with 'Student Project Portal' on the left and 'Student Login' and 'Admin Login' on the right. The main content is a 'Student Login' form. It has a title 'Student Login' in blue. Below the title are two input fields: 'Email Address' with the value 'nive1910@gmail.com' and 'Password' with the value 'stargirl'. A blue 'Login' button is at the bottom of the form. Below the button are two links: 'Don't have an account? [Register here](#)' and '[Admin Login](#)'.

Fig 5.2 User Login details

**Fig 5.3 Student Dashboard**

The screenshot shows the 'Upload New Project' form. It contains the following fields and elements:

- Project Name ***: Text input field with the value 'jazz singer'.
- Roll Number**: Text input field with the value '46'.
- Description ***: Text area with the value 'hi i'm neviditha'. Below the text area is a note: 'Minimum 10 characters'.
- Tech Stack ***: Text input field with the value 'python,sql'. Below the field is a note: 'Comma-separated values'.
- Project Link (Optional)**: Text input field with the value 'https://github.com/username/project'.
- Upload Media File (Optional)**: A file upload button labeled 'Choose File' and a status 'No file chosen'.
- Below the file upload section is a note: 'Accepted: Images (JPG, PNG, GIF), Videos (MP4, AVI, MOV), PDF, Max size: 16MB'.
- At the bottom is a blue 'Submit Project' button.

Fig 5.4 Uploading a Project

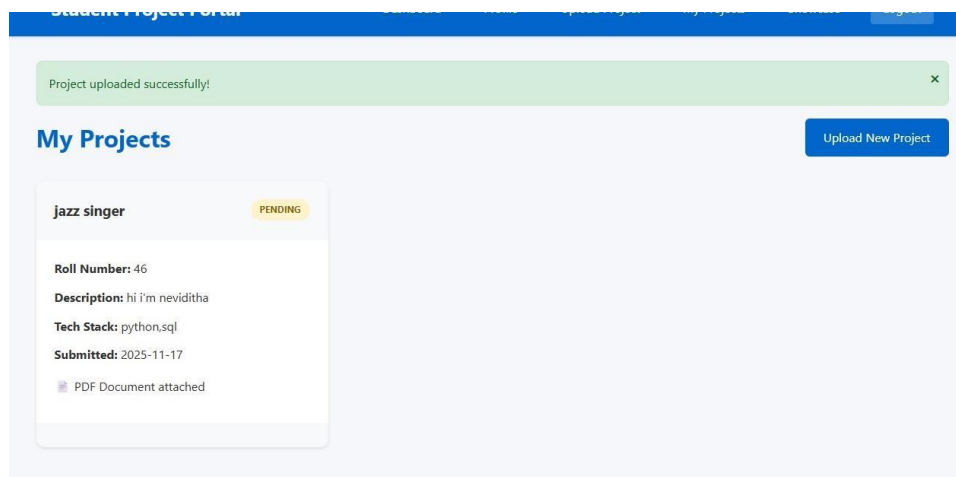


Fig 5.5 My Projects Page

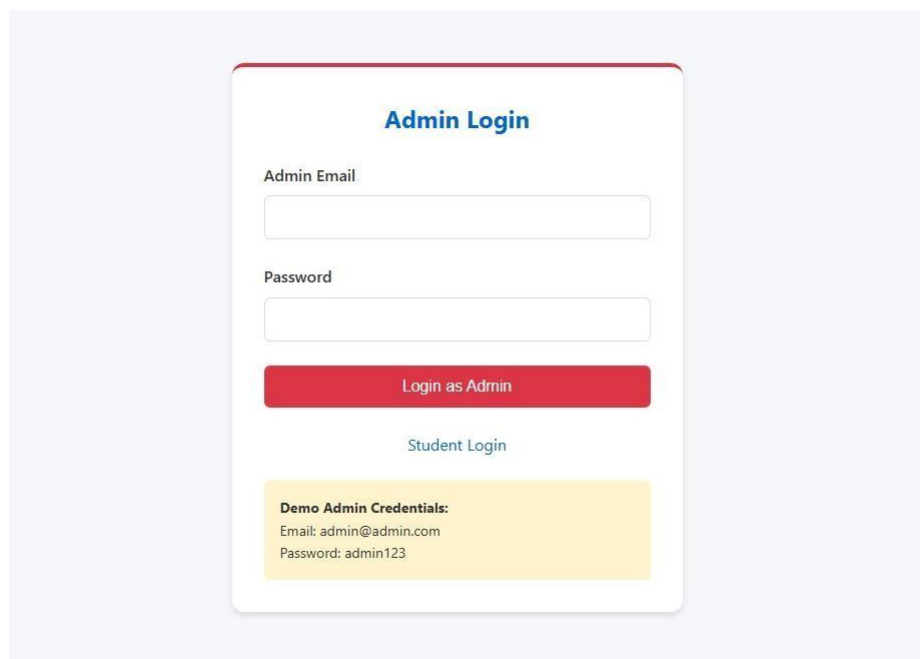


Fig 5.6 Admin Login

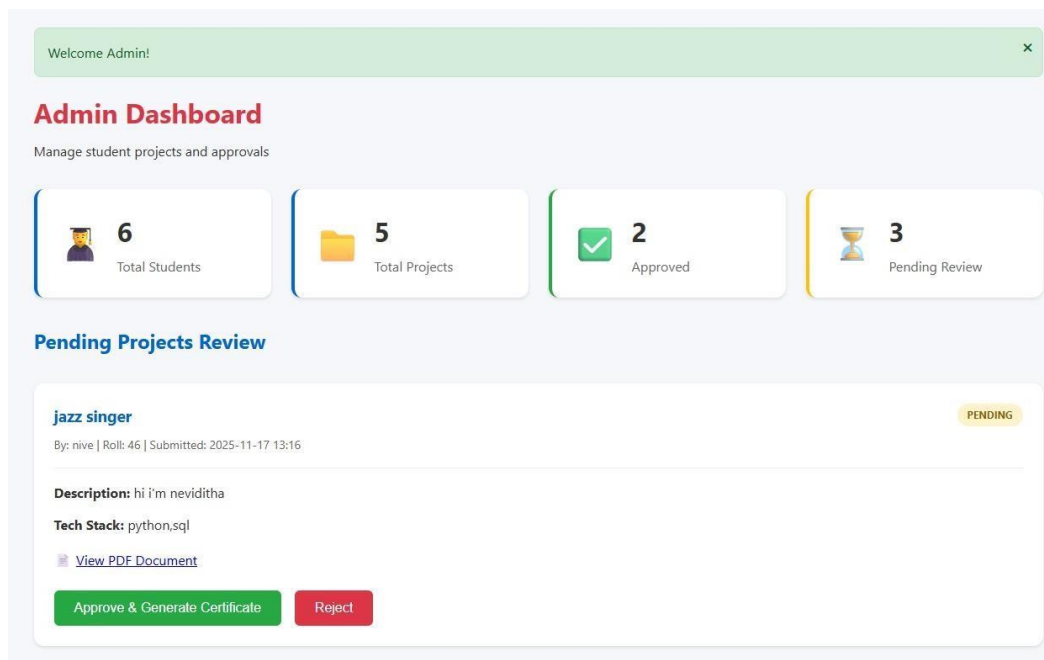


Fig 5.7 Admin Dashboard

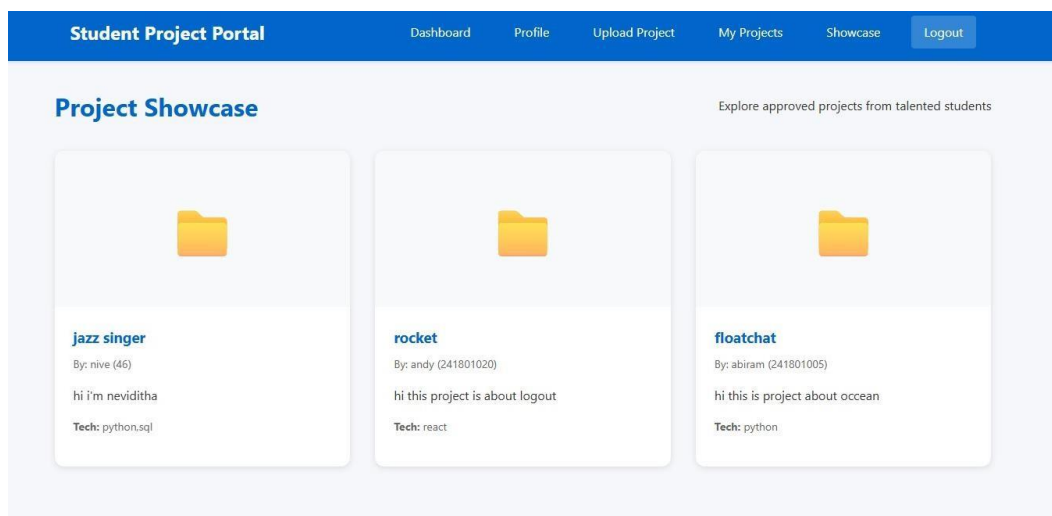


Fig 5.8 Project Showcase

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION:

SkillConnect successfully provides a digital solution for managing student projects within an academic environment. It streamlines project submission, approval, certification, and display, reducing manual work and increasing workflow efficiency. The system ensures proper organization of project records and enhances collaboration by allowing students to explore peer projects.

FUTURE ENHANCEMENTS:

- Live chat between students and faculty.
- Leader board for top-rated projects.
- AI-based plagiarism detection.
- Cloud integration (AWS/S3).
- Mobile application version.
- Real-time project analytics dashboard.

REFERENCES

1. <https://www.w3schools.com/sql/>
2. <https://www.python.org/>
3. <https://dev.mysql.com/doc/>
4. <https://docs.oracle.com/javase/>
5. <https://www.geeksforgeeks.org/dbms/>