A WEB TECHNOLOGY REPORT

ON

"STUDENT COURSE REVIEW SYSTEM"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD OF THE DEGREE

BACHELOR OF

ENGINEERING (COMPUTER

ENGINEERING) 2024-25

UNDER THE GUIDANCE OF

PROF. PRAMOD G PATIL

SUBMITTED BY

Mayuresh Shamkumar Sonar **EXAM SEAT NO.: - T400610442**

DEPARTMENT OF COMPUTER ENGINEERING

Sandip Foundation's

Sandip Institute Of Technology And Research Center

Nashik

CERTIFICATE

This is to certify that the project report entitles

“STUDENT COURSE REVIEW SYSTEM”

submitted by

Mayuresh Shamkumar Sonar **EXAM SEAT NO.: - T400610442**

Is a Bonafide student of this institute and the work has been carried out by him under the supervision of **Prof. Pramod G Patil** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune, University.

Prof. Pramod G. Patil
Guide Name

Dr. Ankita. V. Karale
Head of Department

DEPARTMENT OF COMPUTER ENGINEERING

Sandip Institute of Technology and Research Centre Nashik (422213)

SAVITRIBAI PHULE PUNE UNIVERSITY

2024-2025

ACKNOWLEDGEMENT

I am deeply in deputed to SITRC, Nashik for giving us an opportunity to work on Project "**STUDENT COURSE REVIEW SITE"** and also for their valuable guidance and patience with us. I would like to thank **Prof. PRAMOD G. PATIL** for his support and co-operation in preparing the project report and guiding me. I thank him again for the valuable inputs and providing us his valuable time for starting the project work. I thank **DR. ANKITA V. KARALE**, Head of Department of Computer Engineering Sandip Institute of Technology and Research Centre, Nashik for providing us all the facilities and the very best technical and support infrastructure to carry on our project work. I would like to thank all the user and colleagues who extended help directly or indirectly during our project work. I would like to take this opportunity to express our profound sense of gratitude and respect to all those who helped us throughout.

ABSTRACT

In today's rapidly evolving educational landscape, access to reliable peer feedback is crucial for informed academic decision-making. The **Student Course Review System** is a web-based platform designed to empower students by allowing them to share and access reviews of academic courses in a centralized, secure, and user-friendly environment. Developed using the Flask web framework, MySQL for data persistence, and modern frontend technologies like HTML, CSS, and Bootstrap, this system provides a seamless experience for users to register, log in, browse available courses, and contribute reviews.

The project addresses common gaps in institutional course selection tools by introducing community-driven insights, real-time data presentation, and dynamic course management. Features such as encrypted authentication, session management, role-based access, and the ability to add new courses enrich the system’s functionality. Each course is presented with a description and a direct URL to its enrollment platform, ensuring both context and convenience.

The system adheres to MVC architectural principles and integrates secure development practices, making it scalable and adaptable for future educational ecosystems. It is ideal for academic institutions seeking to increase student engagement and transparency in curriculum feedback loops. The Student Course Review System not only simplifies course comparison but also cultivates a culture of informed learning through shared academic experiences.

Contents

Chapter 1

Introduction

The integration of microservices architecture and distributed systems into modern applications has created significant challenges in maintaining visibility across complex infrastructures. From cloudnative applications to hybrid deployments, monitoring and logging have evolved as critical components that simplify troubleshooting, improve system reliability, and enable data-driven decisions. During my internship at Wisdom Sprouts, I undertook the development of a Centralized Logging and Monitoring System using Docker. This project combines fundamental principles of Cloud Computing and DevOps to develop a flexible, scalable, and comprehensive platform capable of aggregating logs and metrics from diverse sources and providing actionable insights. This chapter outlines the context, structure, and system design considerations that form the basis of this monitoring solution.

Chapter 2

About the Project

The Student Course Review System is an individual project developed as part of the Web Technology curriculum using the following stack:

- Backend: Python Flask
- Database: MySQL
- Frontend: HTML, CSS, Bootstrap
- Additional Tools: Jinja2, bcrypt, Ngrok (for local sharing)

2.1 Features Implemented

- Registration and secure login
- Password encryption with bcrypt
- Session management and logout
- Role-based access (admin/IP restricted)
- Listing of 25+ engineering-related courses
- Adding a course via the “Add Course” form (available after login)
- Reviewing any course and displaying reviews sorted by timestamp
- Visit button linking each course to its external URL (application portal)
- Responsive interface with clean design and navigation

All pages are styled using a modern UI with a professional layout. The data flow is dynamic and secure, ensuring privacy and reliability.

Chapter 3

Software Requirements Specification

Through the development of this project, I gained experience in full-stack web development and learned to apply theoretical concepts into real-world application.

3.1 Key Learning Outcomes

3.1.1 Web Development using Flask

Learned to build modular, dynamic websites with Flask routes, decorators, and Jinja2 templates.

3.1.2 Database Design with MySQL

Designed and implemented normalized schemas for storing user data, course information, and reviews.

3.1.3 Frontend Development

Applied Bootstrap for responsive design and CSS for custom styling.

3.1.4 Security and Authentication

Implemented secure login with password hashing using bcrypt. Session management ensures authorized access only.

3.1.5 Deployment Options

Learned to host locally, test on Wi-Fi network, and use tools like Ngrok for public access.

3.1.6 Collaboration Tools

Gained confidence in debugging, version control, and managing folder structures.

3.1.7 Project Planning

Understood the importance of requirement gathering, milestone setting, and testing before deployment.

Overall, this project helped me move beyond textbook knowledge and into hands-on development and problem-solving.

Chapter 4

System Design and Implementation

The project architecture follows the MVC pattern:

- Model: MySQL tables for users, courses, and reviews.
- View: HTML + Bootstrap templates rendered dynamically with Jinja2.
- Controller: Flask functions manage user requests and control the flow.

4.1 Modules Implemented

- User Module: Registration, login, and session state handling.
- Course Module: Course listings pulled from DB, with ability to add new ones post-login.
- Review Module: Authenticated users can review, update, or delete their course reviews.
- Admin Module: Admin functions restricted by IP or session user id.

4.2 Technologies Used

- Flask: Lightweight and flexible backend framework.
- MySQL: Robust and reliable RDBMS.
- Jinja2: For looping, filtering, and conditional display in HTML.
- Bootstrap: For responsive and clean UI.
- bcrypt: Password encryption for user safety.
- Ngrok: Temporary public access for remote testing.

All templates include form validation and proper data handling. The review module features timestamps and user tags, while the course module supports external links.

Chapter 5

Analysis

Strengths:
- Real student feedback in one place
- Simple UI for students and admins
- Lightweight and scalable design

Weaknesses:
- Lacks recommendation engine
- No authentication in the basic version

Opportunities:
- AI sentiment analysis
- Multi-language support
- Expand to include offline academic courses

Threats:
- Fake reviews or spam comments
- Competitive platforms like Quora, Reddit

Chapter 6

Result

The **Student Review Site** was successfully developed and deployed in a local testing environment. The platform fulfilled all initial objectives and implemented all planned core functionalities with a high level of stability and usability. Below is a detailed analysis of the outcomes:

6.1 Courses Listed with Editable Descriptions

- The site showcases a curated catalog of 25 unique courses, each carefully populated with relevant details including the title**,** summary**,** category, and a **"Visit Course"** button which redirects to an official course webpage or third-party learning platform.
- Admins can easily edit course descriptions, images, or links via the content management interface.
- The platform supports rich text editing, allowing formatting like bold, italic, or links within course descriptions.

6.2 Review System with Live Feedback Storage

- Each course supports a 5-star rating mechanism where users can assign a rating along with a text-based review.
- All reviews are stored dynamically in the database and loaded in real time without needing a page reload, thanks to AJAX/fetch API.
- A built-in validation system prevents duplicate reviews and filters out empty or offensive inputs.
- The review average is automatically calculated and displayed for each course using aggregate queries.

Chapter 7

Conclusion

The Student Course Review System fulfills its objective of creating an open platform where students can exchange feedback on courses they've taken. It is designed with scalability in mind, meaning future enhancements can be added without altering the core structure.

7.1 Benefits of the system

- Reduces confusion in selecting courses
- Promotes transparency and collaboration
- Encourages students to reflect and document experiences

7.2 Future Scope

- Add likes/dislikes on reviews
- Include star ratings and aggregate averages
- Enable comment threads for interactive discussion

- Admin panel with moderation tools
- Search and filter by topic/department

This project helped me grow as a full-stack developer and gain confidence in building and deploying complete web systems.

References

Flask Documentation : https://flask.palletsprojects.com/

MySQL Documentation : https://dev.mysql.com/doc/

Bootstrap Documentation : https://getbootstrap.com/

https://www.coursera.org/

Jinja2 Template Documentation : https://jinja.palletsprojects.com/