LAB REPORT : SPRINT 1 PROCESS

COURSE CODE: CSE 404
COURSE TITLE: SOFTWARE ENGINEERING AND ISD LABRATORY

Submitted by

Suraiya Mahmuda (364)

Submitted to

Dr. Md. Mushfique Anwar, Professor

Dr. Md. Humayun Kabir, Professor

Computer Science and Engineering

Jahangirnagar University

Dhaka, Bangladesh

January 07, 2025

Contents

1 Introduction **1**

2 Sprint Planning Meeting **1**

2.1 Project Overview 1

2.2 Sprint 1 Planning Overview 2

2.2.1 Project Backlog 2

2.2.2 Sprint 1 Backlog 3

2.3 My Feature 3

2.4 Tools for Sprint Management 3

3 Daily Scrum Meeting **4**

3.1 Daily Scrum Meeting 1 4

3.2 Daily Scrum Meeting 2 5

3.3 Daily Scrum Meeting 3 5

3.4 Daily Scrum Meeting 4 6

4 Sprint 1 Retrospective Meeting **10**

5 Screenshots Git logs **11**

6 Outcome Result After Sprint 1 **13**

1. Introduction

The goal of this lab was to gain practical experience with the Agile Scrum methodology by completing Sprint 1 of the software development process. The sprint emphasized delivering a functional project increment within a specified timeframe. Key activities included creating the sprint backlog, conducting sprint planning sessions, attending daily Scrum meetings, and performing a sprint retrospective. This report summarizes the main activities carried out during the sprint, highlighting meeting outcomes, tasks accomplished, and lessons learned along the way.

2. Sprint Planning Meeting

2.1 Project Overview

Our project, **Exam Office Management System**, was initially developed with Jahangirnagar University as its primary focus. The software aims to improve the **accuracy, efficiency, and transparency** of exam-related processes. It provides a **centralized platform** enabling the exam office to manage various tasks seamlessly and effectively.

Roles and Team Members

- **Product Owner:** Mahfuz Anam(MA)
- **Scrum Master:** Kamrul Hasan Nahid(KH)
- **Team Members:**
  - Mohammed Tamjid Islam(TI)

- Suraiya Mahmuda(SM)
- Abdullah Al Mamun(AM)
- Farhan Ahmed Onu(FA)

2.2 Sprint 1 Planning Overview

During our Sprint 1 Planning meeting, we selected key features from the **Project Backlog** to focus on. Below are the details:

2.2.1 Project Backlog

The following features were identified for implementation during the Agile Scrum process:

1. Registration
2. Login Functionality
3. Publish Exam Schedule
4. Register for Upcoming Exams
5. Publish Results
6. View Results
7. Apply for Marksheet
8. Apply for Certificate
9. Publish Exam Calendar
10. Manage Exam Materials
11. Manage Answer Scripts
12. Remunerate Teachers
13. Track Students' and Teachers' Attendance During Exam
14. Approve Application for Physically Disabled and Sick Students

2.2.2 Sprint 1 Backlog

The team committed to completing the following features in Sprint 1, with specific tasks assigned to team members:

- **Registration** – *Mohammed Tamjid Islam(TI)*
- **Login Functionality** – *Mahfuz Anam(MA)*
- **Publish Exam Schedule** – *Abdullah Al Mamun(AM)*
- **Register for Upcoming Exam** – *Kamrul Hasan Nahid(KH)*
- **Publish Result** – *Farhan Ahmed Onu(FA)*
- **View Results** – *Mahfuz Anam(MA)*
- **Apply for Marksheets** – *Mohammed Tamjid Islam(TI)*
- **Apply for Certificates** – *Suaiya Mahmuda(SM)*

2.3 My Feature

My feature is **Apply for Certificates**. Student views and applies for a certificate. System checks eligibility and directs to payment options if eligible. Student completes the payment and receives confirmation messages as appropriate.

2.4 Tools for Sprint Management

To ensure smooth tracking and collaboration during the sprint, the following tools were chosen:

- **Toggl:** To monitor work hours and estimate the time needed to complete tasks during a sprint
- **Discord:** To enhance team communication and share updates on current tasks
- **Trello:** To track task progress effectively, classify tasks into three categories: ”To Do”, ”In Progress”, and ”Done” ensuring the entire team has visibility into the workflow

Agreed Git Conventions

Branch Naming Convention

For each feature or task, create a new branch in the repository using the following format: `<functionality>-<DeveloperName>`
Example: `ApplyForCertificate-Suraiya`

Commit Message Convention

We established a standard for commit messages to ensure clarity and consistency. The format for commit messages is: `<tag>: <What did you do>`

3. Daily Scrum Meeting

Our Scrum meetings are held once every two days.

3.1 Daily Scrum Meeting 1

What I Did Yesterday

- Reviewed my user story
- Familiarized myself with our coding conventions
- Reviewed Toggl
- Reviewed tools for the project

Problem Faced

- Challenges in documenting the Architectural Pattern(MVT)

What I Will Do Today

- Work on resolving the issues related to MVT implementation

3.2 Daily Scrum Meeting 2

What I Did Yesterday

- Create a new branch locally and started works
- Set up all necessary models

Problem Faced

- No issues encountered

What I Will Do Today

- I will develop django app for ”Apply for Certificate”

3.3 Daily Scrum Meeting 3

What I Did Yesterday

- Configured `models.py`, `views.py`, and `templates.py` for ”Apply for Certificate”

Problem Faced

- Facing some issues

What I Will Do Today

- I will solve my issues, perform unit testing, follow coding standard and write documentation for ”Apply for Certificate”

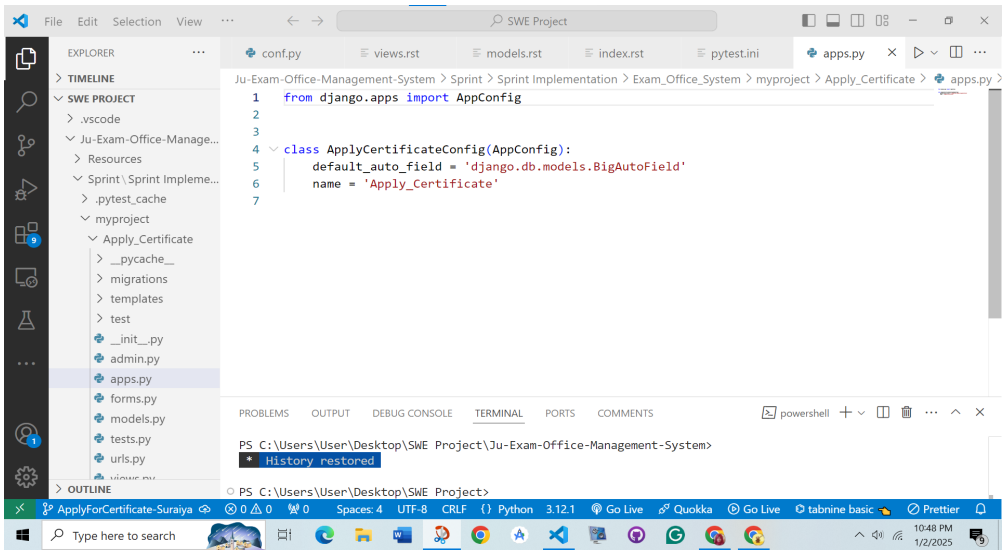

Figure 3.1: Created ApplyForCertificate App

3.4 Daily Scrum Meeting 4

What I Did Yesterday

- Solved my models issuesdocumentation using Sphinx
- Completed implementation of assigned functionalities
- Completed documentation using Sphinx
- Performed unit testing on two modules

Problem Faced

- Facing issues with unit testing

What I Will Do Today

- Will try to solve unit testing related issues

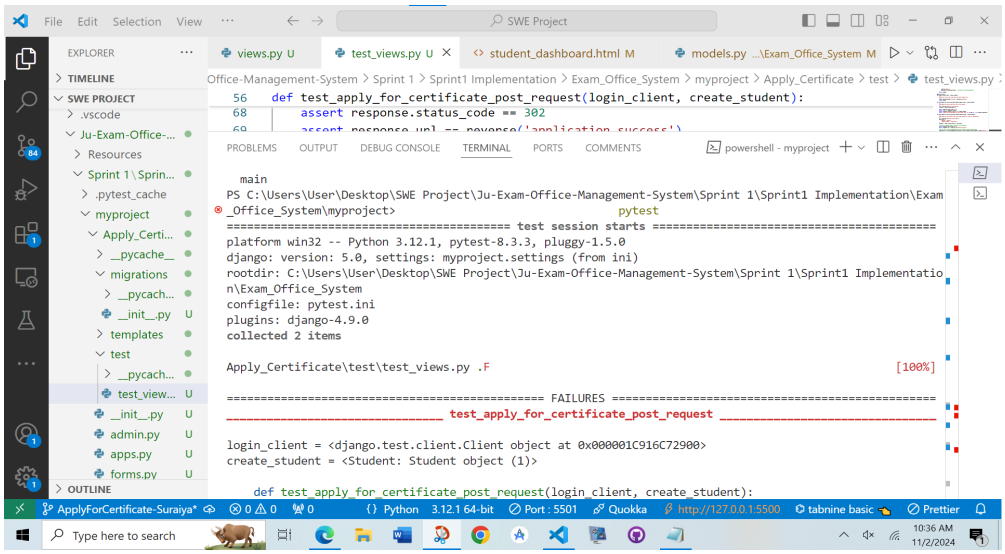

Figure 3.2: Unit Testing

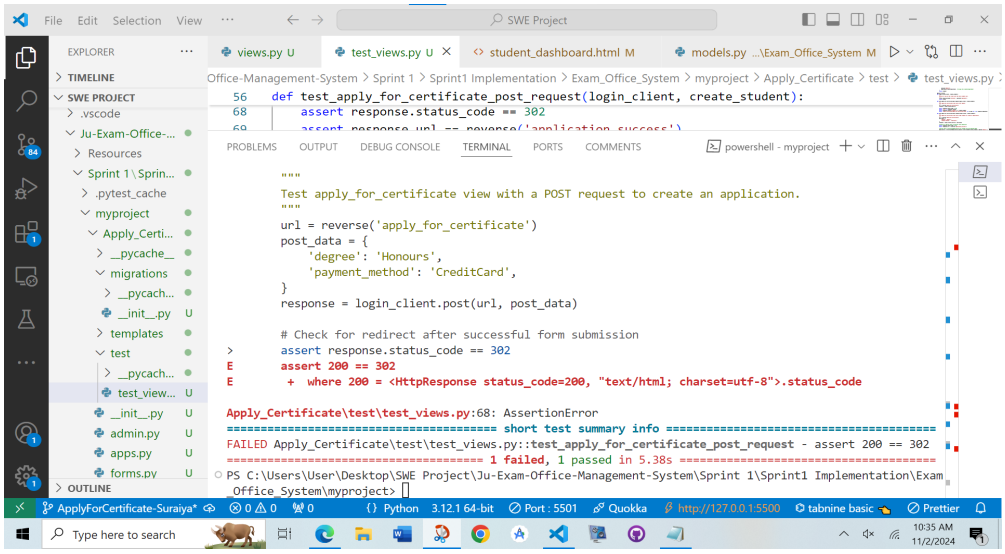

Figure 3.3: Unit Testing

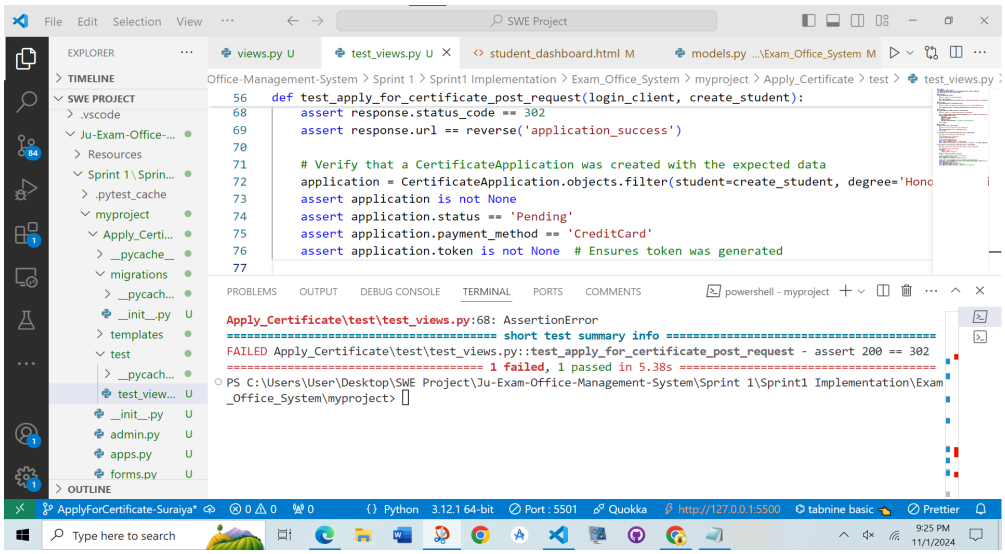

Figure 3.4: Unit Testing

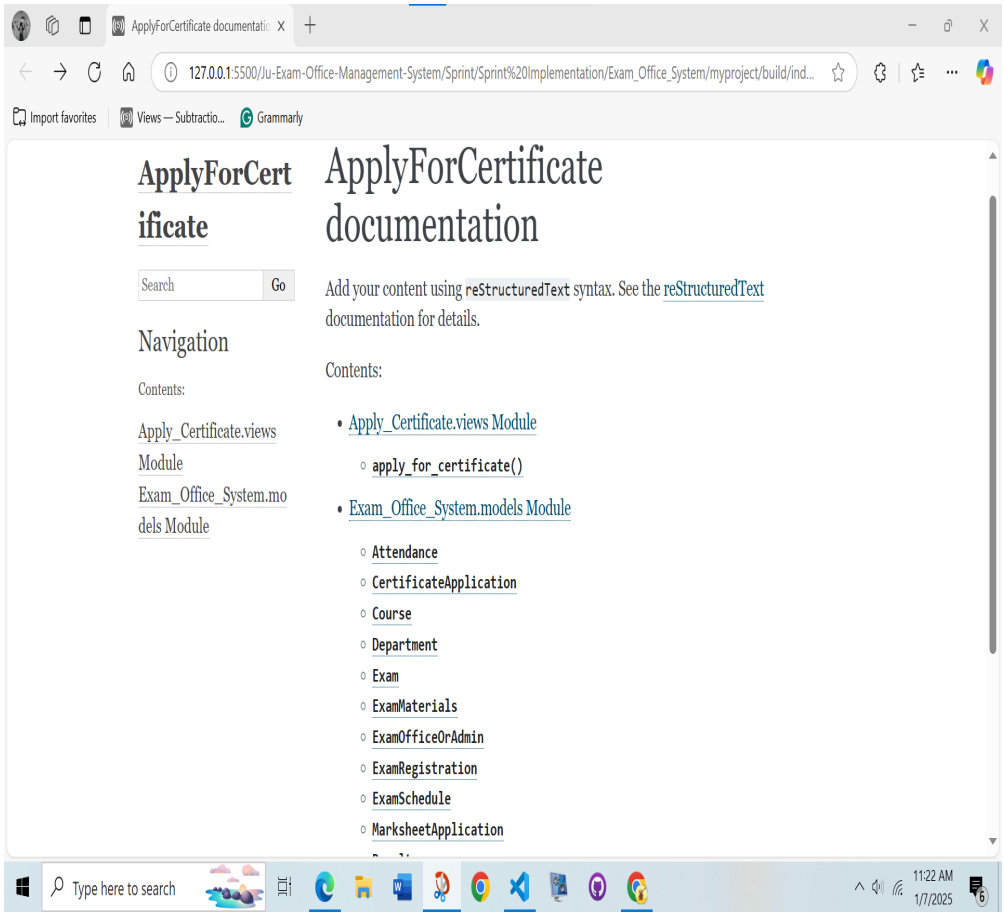

Figure 3.5: Performed Documentation

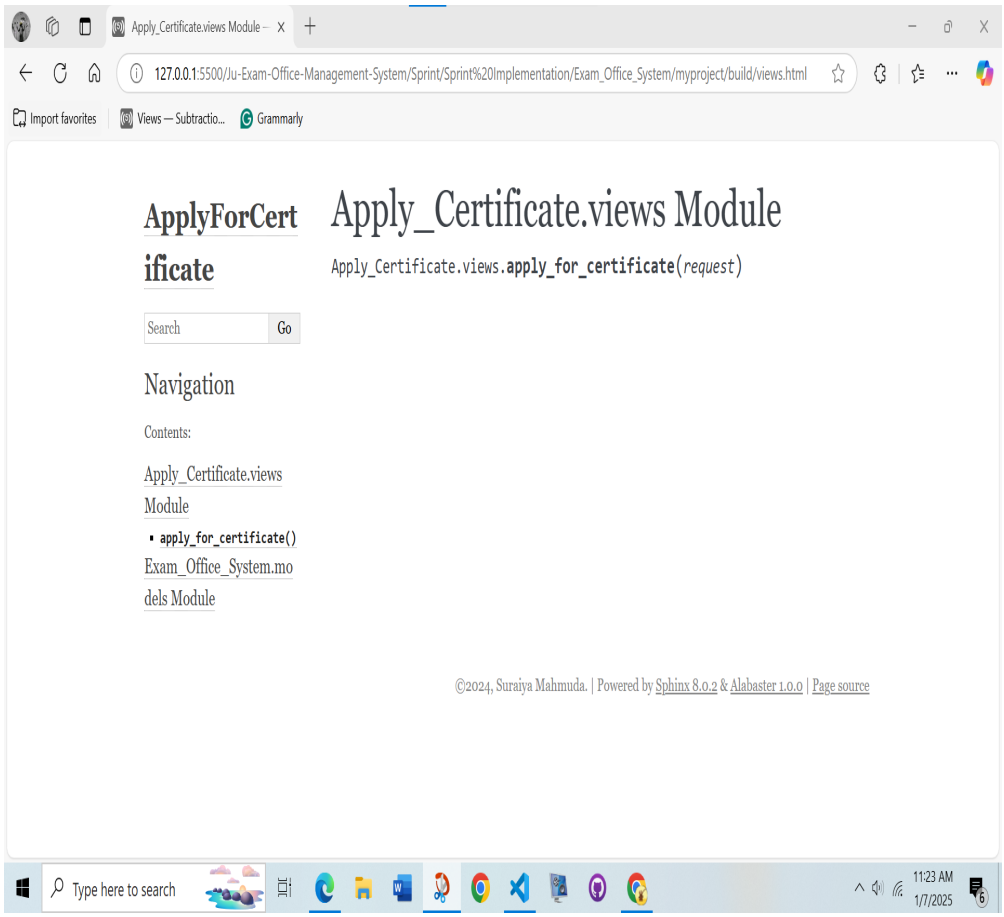

Figure 3.6: Performed Documentation

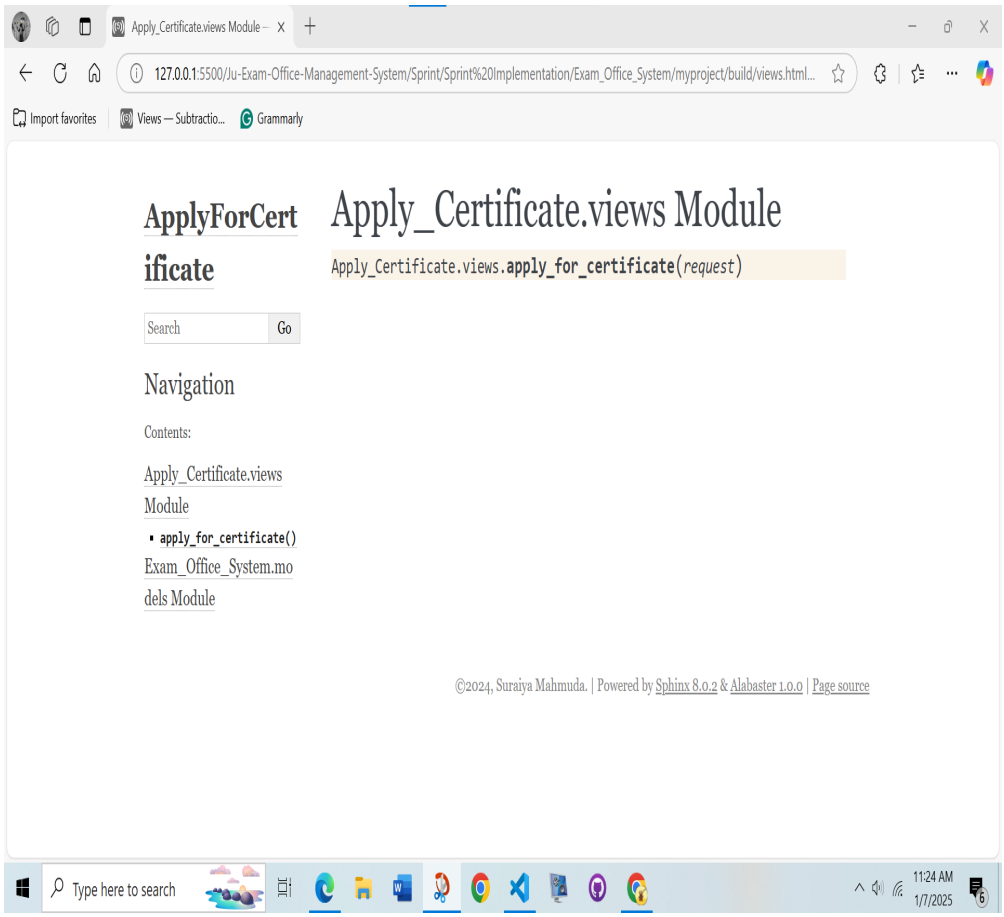

Figure 3.7: Performed Documentation

4. Sprint 1 Retrospective Meeting

Sprint 1 Retrospective Meeting

What Went Well for Me

- Successfully implemented the ”Apply for Certificate” feature, ensuring its functionality and meeting the requirements
- Performed unit testing and added thorough documentation for the `models.py` and `views.py` modules

- Consistently followed the team's established coding standards and best practices

What Went Wrong for Me

- While performing unit testing for all modules, I encountered issues in one module where the tests failed

Learning Outcomes

- Gained experience in writing and executing unit tests with well-defined test cases using PyTest
- Improved collaboration through the use of Discord for discussions and Trello for task management and distribution
- Learned to effectively track time for specific tasks using Toggl
- Applied coding standards and architectural patterns to maintain a consistent project-level codebase
- Enhanced code readability and maintainability by adding documentation with Sphinx
- Effectively utilized GitHub for version control and Git Wiki for maintaining task-based documentation

5. Screenshots Git logs

To begin working on the new feature or task, a branch was created. This ensures the main branch remains unaffected while development progresses.

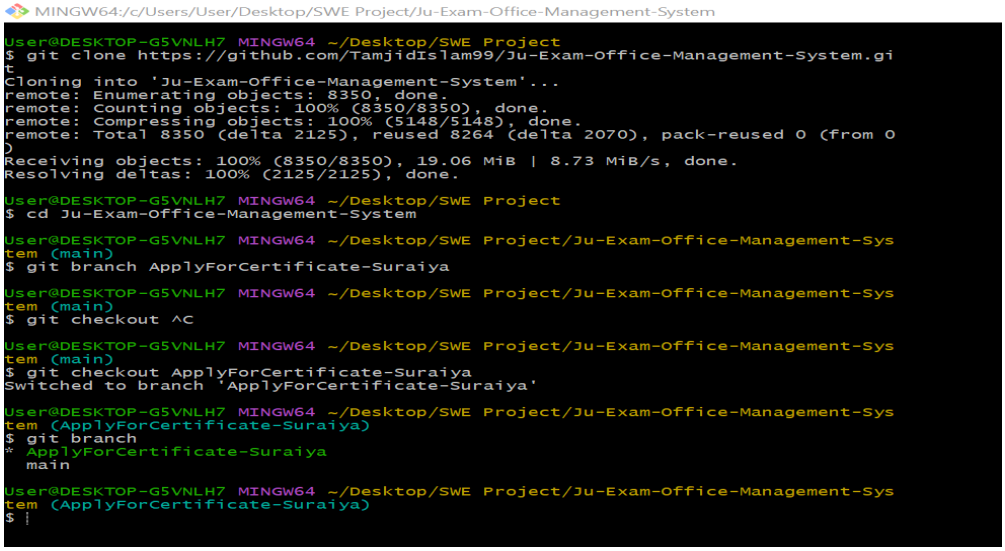

Figure 5.1: Creating a New Branch: ApplyForCertificate

After making necessary changes to the files, the `git add` command was used to stage the files, preparing them for a commit. This ensures only the intended modifications are included.

Once the files were staged, the `git commit` command was executed to record the changes along with a descriptive message, maintaining a clear history of modifications. The `git push` command was used to upload the committed changes to the remote repository, making them available for collaboration and further integration.

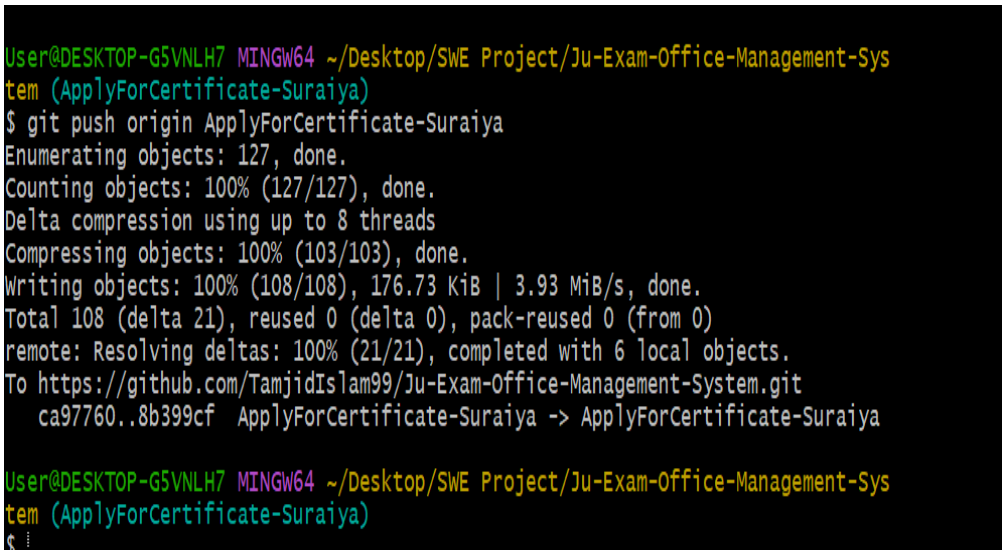

Figure 5.2: Pushing Changes to Remote Repository with `git push`

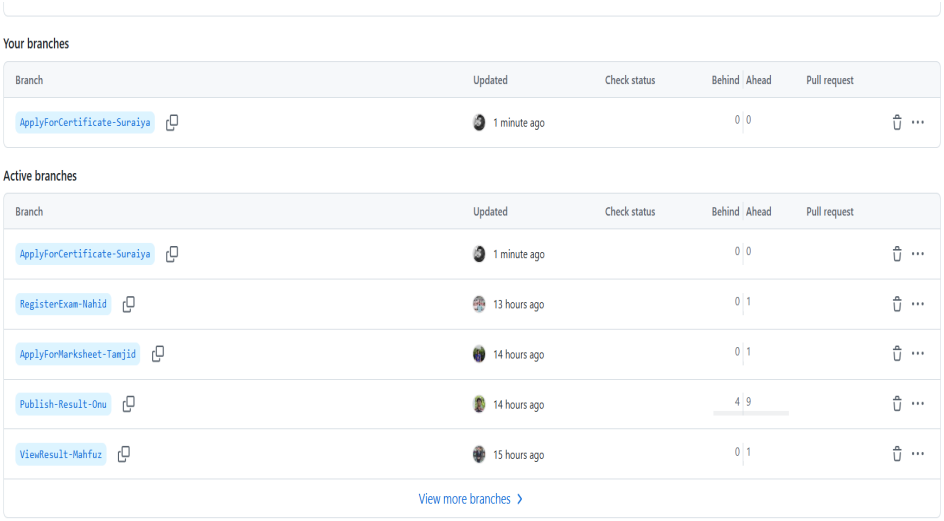

Figure 5.3: After pushing the ”ApplyForCertificate-Suraiya” branch

6. Outcome Result After Sprint 1

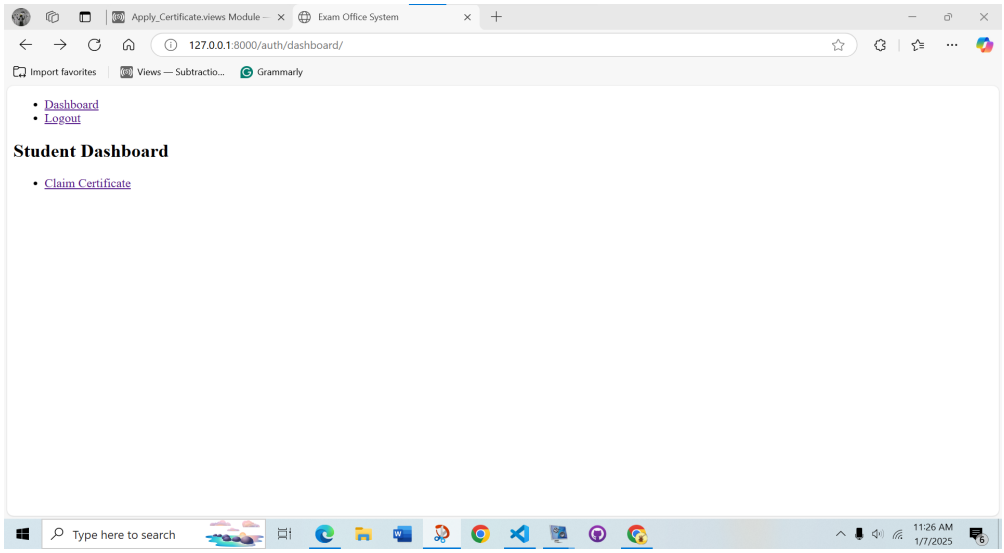

Figure 6.1: Submission page

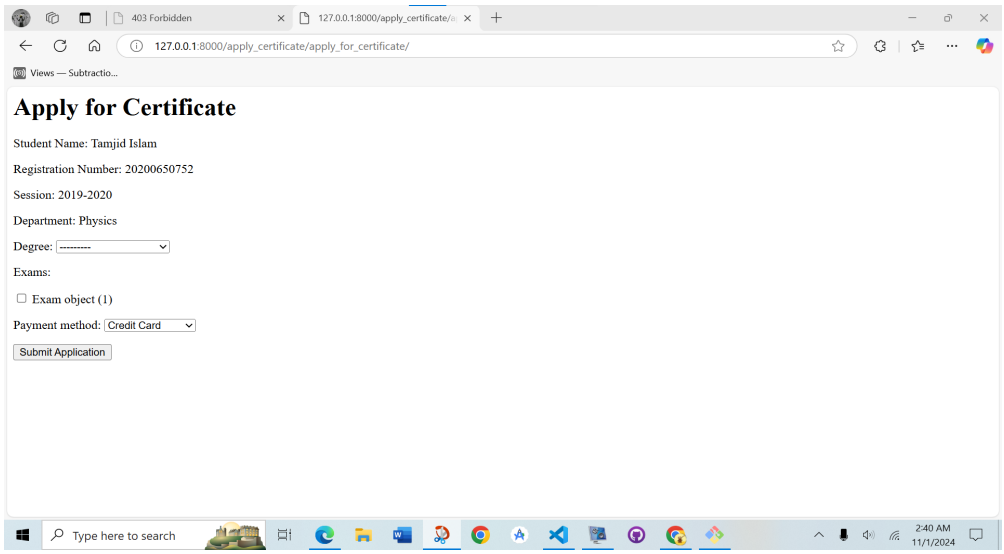

Figure 6.2: Submission page