LAB REPORT : SPRINT 2 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 Planning Overview 2

2.2.1 Project Backlog 2

2.2.2 Sprint 2 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 2 Retrospective Meeting **6**

5 Our TDD Approach **7**

5.1 Introduction 7

5.2 Test Cases 7

5.2.1 Test Case 1: Test Request Accommodation Success 7

5.2.2 Test Case 2: Test Request Accommodation Missing Document . 8

5.2.3 Test Case 3: Test Review Accommodation Success 8

5.2.4 Test Case 4: Test Review Accommodation Missing Comments . 8

5.2.5 Test Case 5: Test Accommodation Status Page 8

5.2.6 Test Case 6: Test Accommodation Status Page Not Found . . . 8

5.2.7 Test Case 7: Test Review Accommodation Access By Non Department User 8

6 Screenshots of My TDD Approach **8**

7 Outcome Result After Sprint 2 **10**

1. Introduction

The goal of this lab was to gain practical experience with the Agile Scrum methodology by completing Sprint 2 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:** Farhan Ahmed Onu(FA)
- **Scrum Master:** Mahfuz Anam(MA)
- **Team Members:**
  - Mohammed Tamjid Islam(TI)

- Suraiya Mahmuda(SM)
- Abdullah Al Mamun(AM)
- Kamrul Hasan Nahid(KH)

2.2 Sprint 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 2 Backlog

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

- **Publish Exam Calendar** – *Abdullah Al Mamun(AM)*
- **Manage Exam Materials** – *Kamrul Hasan Nahid(KH)*
- **Manage Answer Scripts** – *Farhan Ahmed Onu(FA)*
- **Remunerate Teachers** – *Mahfuz Anam(MA)*
- **Track Students' and Teachers' Attendance During Exam** – *Mohammed Tamjid Islam(TI)*
- **Approve Application for Physically Disabled and Sick Students** – *Suaiya Mahmuda(SM)*

2.3 My Feature

My feature is **Approve Application for Physically Disabled and Sick Students**. It has the workings to manage student applications for accommodations due to disability or illness.

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

- Revised the user story
- Created an app and branch for my feature

Problem Faced

- No Issues

What I Will Do Today

- I will try to implement my feature using TDD

3.2 Daily Scrum Meeting 2

What I Did Yesterday

- Created some test cases based on my feature for following TDD approach
- They are failed for not having the main code

Problem Faced

- Faced problem with test cases

What I Will Do Today

- Will try to solve test cases issues
- Will try to implement the all necessary things for sprint 2

3.3 Daily Scrum Meeting 3

What I Did Yesterday

- Tests the previous test codes and starts to create views.py

Problem Faced

- No Issues till

What I Will Do Today

- Will implement views.py and urls.py

3.4 Daily Scrum Meeting 4

What I Did Yesterday

- Made views.py, urls.py and test cases for views.py and urls.py
- Watching tutorias for integration testing. Trying to merge branches to one branch

Problem Faced

- No issues

What I Will Do Today

- Preparing for Integration testing using Circle CI

4. Sprint 2 Retrospective Meeting

What Went Well for Me

- Successfully completed the **"Approve Application for Physically Disabled and Sick Students"** feature
- Implemented the feature by following the Test-Driven Development (TDD) approach
- Ensured that all relevant test cases were written before code implementation
- Wrote unit tests to verify the functionality of attendance tracking using pytest.
- Tried to perform integration testing but failed
- Followed the coding standards determined by our team to ensure consistency and readability in the codebase

- Ensured that all test cases passed successfully before merging the feature into the main branch

What Went Wrong for Me

- Unable to perform integration testing
- Some test codes failed during execution

Learning Outcomes

- We can again try the integration testing so that we can successfully do the test

5. Our TDD Approach

5.1 Introduction

This document presents the test cases for the **"Approval of Application for Physically Disabled and Sick Students"** feature of the `Exam_Office_System` project. Initially, all test cases were failing due to the absence of corresponding functions in the `views.py` file. After implementing the required view functions, the tests were successfully passed.

5.2 Test Cases

5.2.1 Test Case 1: Test Request Accommodation Success

Objective: Tests that a student can successfully submit an accommodation request.

5.2.2 Test Case 2: Test Request Accommodation Missing Document

Objective: Tests submitting a request with a missing required document (failure).

5.2.3 Test Case 3: Test Review Accommodation Success

Objective: Tests the department successfully reviewing an accommodation request.

5.2.4 Test Case 4: Test Review Accommodation Missing Comments

Objective: Tests reviewing a request with missing required comments (failure).

5.2.5 Test Case 5: Test Accommodation Status Page

Objective: Tests that the student can view the status of their accommodation request.

5.2.6 Test Case 6: Test Accommodation Status Page Not Found

Objective: Tests the scenario where the accommodation request doesn't exist (failure).

5.2.7 Test Case 7: Test Review Accommodation Access By Non Department User

Objective: Tests that a non-department user cannot access the review page.

6. Screenshots of My TDD Approach

At first I wrote the test cases for the features as it is written in the previous section and they failed.

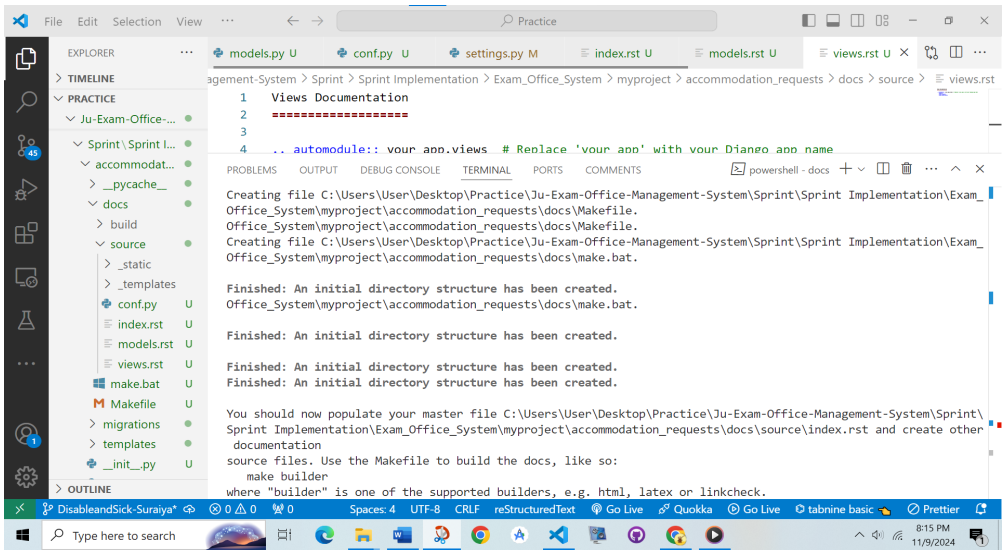

Figure 6.1: Failed Test Codes

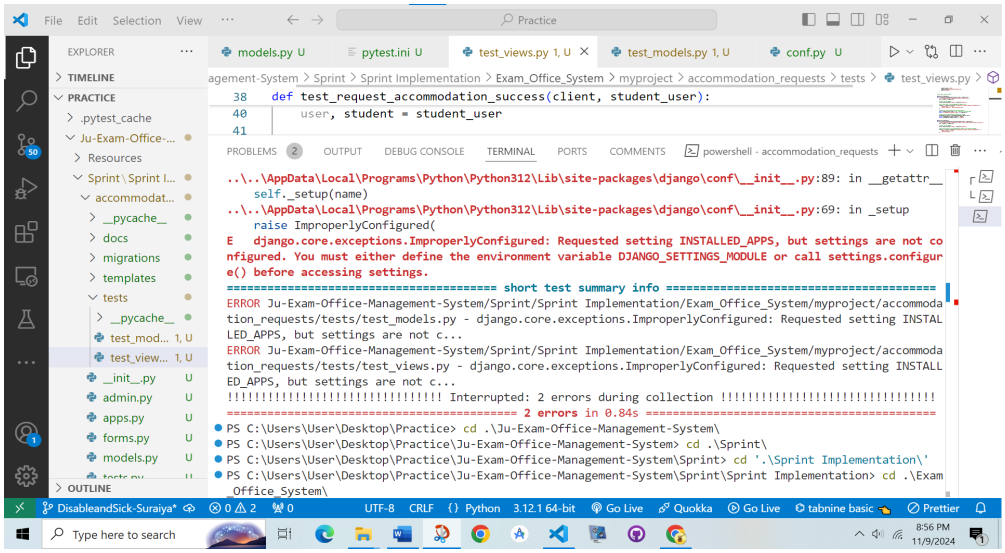

Figure 6.2: Failed Test Codes

Then we go for integration testing in circle CI. The Integration Testing is described in detailed Continuous Integration report. Here I am showing the result

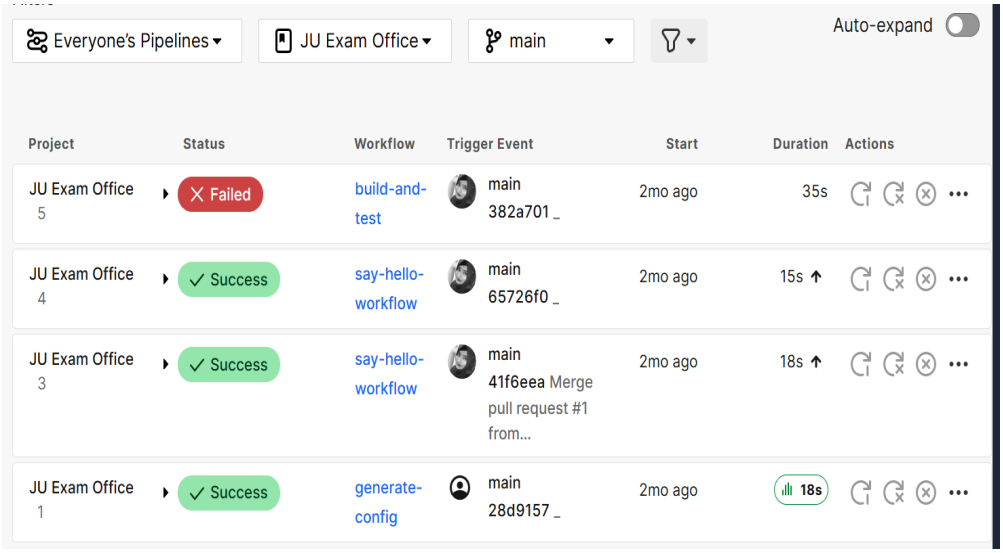

Figure 6.3: CI Result

7. Outcome Result After Sprint 2

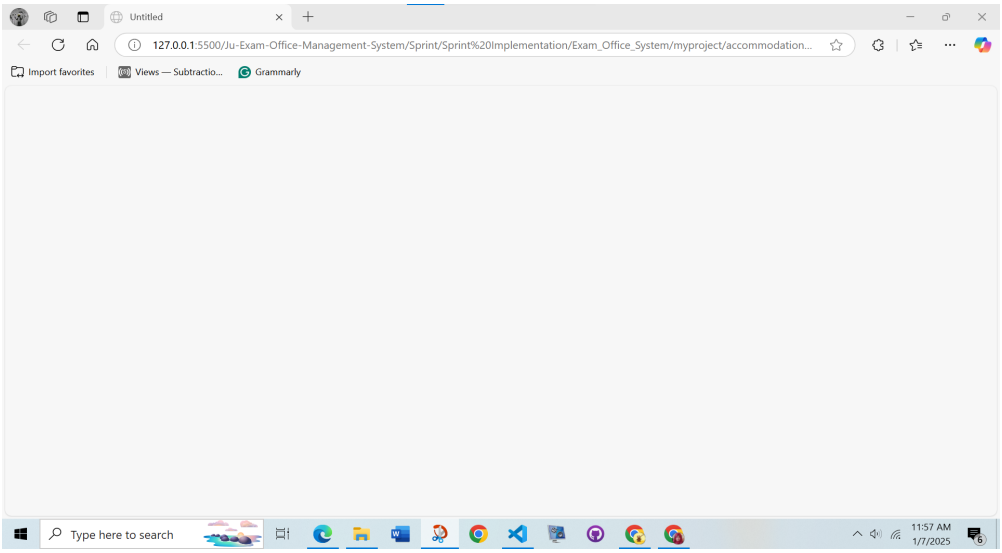

Figure 7.1: Output(Submission Page)

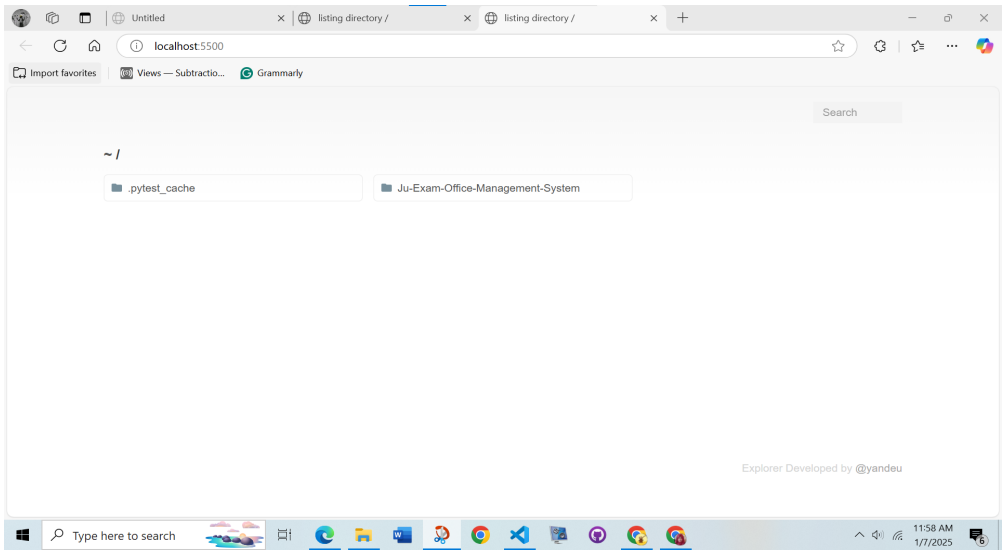

Figure 7.2: Output(Submission Page)