Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

Smart UniAssist Testing

Course Title: Software Testing Quality Assurance Lab
Course Code: CSE 453
Section: 213-D1

Students Details

Name	ID
Md.Shaon Howlader	213902064
Md. Robin Hossain	213902105

Submission Date: 24.05.25
Course Teacher's Name: Md. Zahidul Hasan

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

1 Introduction **2**

1.1 Overview 2

1.2 Motivation 2

1.3 Problem Definition 2

1.3.1 Problem Statement 2

1.4 Applications 3

2 Design/Development/Implementation of the Project **4**

2.1 Introduction 4

2.2 Project Details 4

2.3 Implementation 5

2.4 Code 5

3 Testing and Test Cases **8**

3.1 Test Cases 9

4 Conclusion **12**

4.1 Discussion 12

4.2 Limitation 12

4.3 Future Work 12

Chapter 1

Introduction

1.1 Overview

Smart UniAssist is an integrated smart university assistance platform developed to enhance the academic and administrative experience within educational institutions. It serves as a centralized system that provides a wide range of services to students, faculty, and administrative staff, ensuring smoother communication, better access to information, and automation of routine tasks.The project aims to bridge the gap between traditional administrative systems and modern educational needs by leveraging automation, AI, and user-centric design.

1.2 Motivation

Universities often struggle with outdated systems, scattered information, and inefficient communication, which negatively impact students and staff. The motivation behind Smart UniAssist is to modernize these processes by creating a smart, unified platform that simplifies academic and administrative tasks. It aims to enhance user experience, save time, and promote a more connected and efficient university environment through automation and real-time support.

1.3 Problem Definition

1.3 Problem Statement

In many universities, students, faculty, and administrative staff face challenges due to fragmented systems, manual processes, and lack of real-time communication. Students often struggle with accessing academic information, managing schedules, and receiving timely updates. Faculty members deal with inefficient tools for tracking attendance, managing coursework, and communicating with students. Administrators, on the other hand, face difficulties in managing data, disseminating information, and responding to student needs promptly. These issues lead to confusion, time delays, reduced produc-

tivity, and a lack of engagement across the academic community. There is a pressing need for a centralized, smart, and user-friendly platform that can streamline academic and administrative functions while improving overall coordination and communication within the university.

Design Goals / Objectives

The primary goal of **Smart UniAssist** is to develop a smart, unified platform that enhances the efficiency, accessibility, and user experience of academic and administrative processes in universities. The key objectives of the system are:

1. **Centralized Access:**
   Provide a single platform where students, faculty, and administrators can access all relevant information and services.
2. **User-Friendly Interface:**
   Design an intuitive and responsive interface that is easy to navigate for all user types.
3. **Automation of Routine Tasks:**
   Automate tasks such as course registration, attendance tracking, notifications, and grade reporting to reduce manual workload.
4. **Real-Time Communication:**
   Enable instant communication between students, faculty, and administration through alerts, messages, and announcements.
5. **Academic Progress Tracking:**
   Allow students to monitor their academic performance and degree progress through personalized dashboards.
6. **Smart Assistance:**
   Integrate an AI-powered virtual assistant to handle common queries and provide guidance on academic activities.

1.4 Applications

Smart UniAssist can be applied in a variety of ways to improve the academic and administrative functions of educational institutions. Smart UniAssist serves as a centralized platform for managing university services, including academic planning, online attendance, real-time notifications, and AI-powered student support. It also facilitates feedback collection, resource scheduling, and mobile access, making university processes more efficient and accessible.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

Smart UniAssist is a comprehensive digital platform designed to streamline university-related processes, enhance communication, and provide seamless academic assistance using the power of smart technologies. The project aims to bridge the gap between traditional administrative systems and modern educational needs by leveraging automation, AI, and user-centric design. Whether it's managing course registrations, tracking academic progress, accessing campus services, or receiving timely notifications, Smart UniAssist offers an all-in-one solution tailored to the dynamic environment of universities.

2.2 Project Details

Smart UniAssist is a comprehensive platform developed to enhance the management and interaction within university environments. The project involves designing and implementing a web and mobile-based system that integrates academic, administrative, and communication functionalities into a single user-friendly interface.

Key aspects of the project include:

- **Platform Development:**
  Building a responsive web application and mobile app compatible with various devices to ensure accessibility anytime, anywhere.
- **User Roles:**
  Supporting multiple user roles such as students, faculty, and administrators with role-specific dashboards and permissions.
- **Technology Stack:**
  Utilization of modern technologies such as React or Angular for the frontend,

Node.js or Django for the backend, and databases like MySQL or MongoDB for data storage.

The project aims to deliver an efficient, scalable, and easy-to-use system that improves communication, automates routine tasks, and enhances the overall educational experience

2.3 Implementation

Smart UniAssist is developed through a phased approach including requirement analysis, system design, and development of responsive frontend and robust backend using modern technologies. It incorporates secure database management and integrates an AI-powered assistant for user support. Rigorous testing ensures reliability before deployment on cloud or university servers, with ongoing maintenance for smooth operation.

2.4 Code

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.common.exceptions import
   NoAlertPresentException
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as
    EC
from selenium.webdriver.support.ui import WebDriverWait
import time


Service = Service(executable_path="chromedriver.exe")
driver = webdriver.Chrome(service=Service)


driver.get("http://localhost:3000/") #website call

time.sleep(3)
driver.find_element(By.CLASS_NAME, "btn").click()


time.sleep(2)

driver.find_element(By.NAME, "username").send_keys("Testser")
driver.find_element(By.NAME, "email").send_keys("test@example
   .com")
driver.find_element(By.NAME, "phone").send_keys("01700000000"
   )
```

```
time.sleep(1)
driver.find_element(By.NAME, "student_id").send_keys("123456789")
driver.find_element(By.NAME, "password").send_keys("123")

driver.find_element(By.XPATH, "//button[text()='Register']").click()

# Wait to ensure alert comes
time.sleep(3)

try:
    alert = driver.switch_to.alert
    print(" Alert Text:", alert.text)
    alert.accept()
except NoAlertPresentException:
    print("not find")

time.sleep(3)
driver.find_element(By.CLASS_NAME, "logout").click()
time.sleep(3)
driver.find_element(By.CLASS_NAME, "login").click()

student_id_input = driver.find_element(By.NAME, "student_id")
password_input = driver.find_element(By.NAME, "password")

student_id_input.send_keys("213902105")  # Replace with your test student ID
password_input.send_keys("123")   # Replace with your test password

    # Click the Log In button
login_button = driver.find_element(By.XPATH, "//button[@type='submit']")
login_button.click()

    # Wait for alert or navigation
time.sleep(2)

    # Try to switch to alert
try:
    alert = driver.switch_to.alert
    print(" Alert Text:", alert.text)
    alert.accept()
except:
    print(" No alert popup detected")


time.sleep(3)
reminder_link = driver.find_element(By.ID, "reminderlink")
reminder_link.click()
```

```
time.sleep(5)
recommendation_link = driver.find_element(By.ID, "
    recommendationlink")
recommendation_link.click()
time.sleep(8)
career_link = driver.find_element(By.ID, "career")
career_link.click()
time.sleep(3)
driver.find_element(By.CLASS_NAME, "logout").click()
time.sleep(3)


\driver.quit()
```

Chapter 3

Testing and Test Cases

3.1 Test Cases

I. Login Functionality Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remark
TC01	Valid login	1. Go to Login page 2. Enter valid StudentID and password 3. Click Login	User is redirected to dashboard	User is redirected to dashboard	Pass	
TC02	Invalid password	1. Enter valid StudentID but wrong password 2. Click Login	Error message: "Invalid credentials"	Error message: "Invalid credentials"	Pass	
TC03	Empty fields	1. Leave StudentID and password empty 2. Click Login	Validation error: "StudentID and password required"	Validation error: "StudentID and password required"	Pass	
TC04	Session persistence	1. Login successfully 2. Refresh page or reopen tab	User remains logged in	User remains logged in	Pass	

II. Signup Functionality Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remark
TC05	Valid signup	1. Open signup form 2. Enter valid StudentID, Name, Email, Phone and password 3. Submit	User is redirected to dashboard	User is redirected to dashboard	Pass	
TC06	Duplicate StudentID	1. Enter existing StudentID 2. Submit	Error: "StudentID already exists"	Error: "StudentID already exists"	Pass	
TC07	Password mismatch	1. Fill form but use different passwords in fields	Error: "Passwords do not match"	Error: "Passwords do not match"	Pass	

III. View Course Test Case

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remark
TC08	View course list	1.Login 2.Navigate to "My Courses"	List of all added courses shown	List of all added courses	Pass	

IV. View Reminders

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remark
TC09	View reminder list	1. Login 2. Navigate to "My Reminders"	List of all added courses shown	List of all added shown	Pass	

V. View Recommendations

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status	Remark
TC10	View based on performance	1. Login 2. Navigate to "Recommendations" 3. Check suggested courses	Courses recommended for retake and Future Courses	Courses recommended for retake and Future Courses	Pass	
TC11	No grades available	1. Login with a new user without results	Message: "No recommendations available yet"	Message: "No recommendations available yet"	Pass	

Chapter 4

Conclusion

4.1 Discussion

The development of Smart UniAssist addresses several critical challenges faced by modern universities, including fragmented administrative systems, inefficient communication, and lack of centralized academic management. By integrating multiple functionalities into a unified platform, Smart UniAssist aims to enhance the overall efficiency and user experience for students, faculty, and administrators alike.One of the key strengths of the project is its modular and scalable design, which allows it to adapt to different university requirements and grow with institutional needs. The incorporation of AI-powered assistance further improves accessibility and responsiveness, helping users resolve queries quickly without overburdening support staff.

4.2 Limitation

Smart UniAssist faces several key limitations. Firstly, the platform's success heavily relies on widespread adoption by students, faculty, and administrators, which can be hindered by resistance to change or lack of training. Secondly, ensuring robust data privacy and security remains a challenge due to the sensitive nature of academic and personal information handled by the system.Additionally, integrating Smart UniAssist with existing legacy systems or third-party software can be complex, requiring significant customization and technical effort.

4.3 Future Work

Future enhancements of Smart UniAssist will focus on integrating advanced analytics to provide personalized academic insights and early identification of student challenges. Mobile app features will be enhanced for better accessibility and offline use. Additionally, strengthening security measures and increasing customization options will ensure the platform remains secure, scalable, and adaptable to diverse institutional needs.