****

**ST. JOSEPH’S COLLEGE OF ARTS AND SCIENCE(AUTONOMOUS),CUDDALORE**

**CASE STUDY PROJECT PRESENTATION**

**STUDENTS ERP**

**Submitted By**

**VISHAL M**

**A21CADB60**

**BE.COMPUTER SCIENCE AND ENGINEERING**

**COIMBATORE INSTITUTE OF TECHNOLOGY**

**Submitted To**

**Dr.CAUROLINE M.E.,Ph.D.,**

**ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER APPLICATION**

**ST. JOSEPH’S COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS),CUDDALORE**

Table Of Contents:

|  |
| --- |
| 1.ABSTRACT 3 |
| 2. LIST OF FIGURES 3 |
| 3.ABBREVATIONS 3 |
| 4.INTRODUCTION TO DOMAIN 3 |
| 5.SOFTWARE REQUIREMENT SPECIFICATION 4 |
| 5.1EXTERNAL INTERFACE REQUIREMENT 4 |
| 5.1.1 USER INTERFACE 4 |
| 5.1.2 HARDWARE INTERFACE 4 |
| 5.1.3 SOFTWARE INTERFACE 5 |
| 5.2FUNCTIONAL REQUIREMENT 5 |
| 5.2.1 DETAILS 5 |
| 5.2.2 TUTOR 5 |
| 5.2.3 CLASS 5 |
| 5.2.4 ACADEMICS 5 |
| 5.2.5 FEES 6 |
| 5.2.6 LIBRARY 6 |
| 5.2.7 HOSTEL 6 |
| 6.USE CASE MODEL 6 |
| 7. DESIGN 8 |
| 7.1 ER DIAGRAM 8 |
| 7.2 DATA FLOW DIAGRAM 8 |
| 8.MODULE DESCRIPTION 9 |
| 8.1 LOGIN ACTIVITY 10 |
| 8.2 DETAILS 10 |
| 8.3 TUTOR 10 |
| 8.4 CLASS 10 |
| 8.5 ACADEMICS 10 |
| 8.6 FEES 10 |
| 8.7 LIBRARY 10 |
| 8.8 HOSTEL 10 |
| 8.9 LOGOUT 10 |
| 9.IMPLEMENTATION 11 |
| 10.RESULT 14 |
| 11.TESTING 15 |
| 12.CORPORATE SOCIAL RESPONSIBILITY 17 |
| 13.FUTURE IMPLEMENTATION 17 |
| 14.REFERENCES 17 |

**1.Abstract:**

A student portal project that acts as an online portal between students and the admin. The system is designed for the students of St. Joseph’s College of Arts and Science. It contains an admin who can enter details of students. Students can then login using provided user id password and view their details. Now admin can add details of Students marks. When students login they can see their own marks details. It is also integrated with hostel so all student can view all instruction and fees payment through this.it is designed to replace the problem of no due form as the admin can view the pending amount of particular student.

**2.LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| FIGURE 1 | LOGIN | 6 |
| FIGURE 2 | HOME PAGE | 6 |
| FIGURE 3 | DETAIL PAGE | 7 |
| FIGURE 4 | TUTOR PAGE | 7 |
| FIGURE 5 | FEES PAGE | 7 |
| FIGURE 6 | CONTEXT DIAGRAM | 8 |

**3.ABBREVATION**

**BD:** Block Diagram

**UC:** Use Case.

**4.INTRODUCTION TO DOMAIN**

Domain web development refers to the process of creating and maintaining a website for a specific domain or subject. It involves various activities such as designing, coding, and implementing web pages, as well as integrating functionality and content relevant to the domain. The domain can be related to a particular industry, business, organization, or any specific area of interest.

In domain web development, the focus is on building websites that cater to the unique needs and requirements of the target audience within a specific domain. This approach allows for a more specialized and tailored user experience, as the website content, features, and design elements are optimized for that particular domain.

The development process typically starts with understanding the goals, objectives, and target audience of the website within the domain. This involves conducting thorough research and analysis to gain insights into the industry or subject matter. By understanding the specific requirements and expectations of the target audience, developers can create a website that effectively communicates and engages with them.

Once the research phase is complete, the actual web development work begins. This includes designing the website's layout, user interface (UI), and user experience (UX) to ensure easy navigation and intuitive interaction. The website is then built using various programming languages such as HTML, CSS, and python, and additional frameworks and tools may be used to enhance functionality and efficiency.

Lastly, domain web development includes ongoing maintenance and updates to ensure the website remains functional, secure, and up-to-date. This involves regular monitoring, bug fixes, security patches, and content updates to provide a seamless and relevant user experience.

**5.SOFTWARE REQUIREMENTS SOECIFICATION**

**5.1 Introduction**

**5.1.1 Purpose**

The purpose of this Document is to describe about the newly Introduced we-based application called “Students portal”. The Student Portal is a web-based application designed to provide a comprehensive platform for students to access their academic records, college, library and hostel details, communicate with faculty members and other students. This document outlines the requirements for the development of the Student Portal.

**5.1.2 Scope**

The website is fully Online so the main goal is to reduce the time and save the time in knowing the information about them. It is a simple and user friendly software so any one can login and access the website. This Software would be more easier to access than other website.

**5.1.3 Overview**

This Software makes the replacement for the queuing and waiting of students to get the various information of various sector. This Software uses our local storage for storing the data. Only the students of the particular college can able to login to the webpage.. This Software is more secure to use and data cannot be stolen by another. therefore it can help to improve the overall student experience.

**5.2 EXTERNAL INTERFACE REQUIREMENTS**

**5.2.1 USER INTERFACE**

A user interface for a student portal should be intuitive, easy to use, and designed with the needs of students in mind. Here are some elements that can be included:

1. Login page: The login page should be the first thing students see when they visit the portal. It should have fields for students to enter their username and password to access their account.

2. Dashboard: Once logged in, students should see a dashboard that displays important information such as their details, fees to paid library and other details. It should be easy for students to navigate to different areas of the portal from the dashboard.

**5.2.2 HARDWARE INTERFACE**

A hardware interface for a student portal would typically involve the use of a computer, laptop, tablet, or smartphone to access the portal. Here are some hardware considerations for a student portal:

1. Compatibility: The portal should be designed to work seamlessly with a variety of hardware devices, including computers, laptops, tablets, and smartphones.

2. Display: The portal should be designed to display properly on a range of screen sizes, from large desktop monitors to small smartphone screens.

3. Input: The hardware interface should support input methods such as keyboard, mouse, touchpad, and touch screen, depending on the device being used.

4. Internet connectivity: The hardware interface should support a range of internet connectivity options, from high-speed wired connections to mobile data networks, to ensure that students can access the portal from anywhere.

5. Security: The portal should be designed with security in mind, to ensure that student data is protected from unauthorized access or hacking attempts.

**5.2.3 SOFTWARE INTERFACE**

The Student Portal System will be developed using the following programming languages and platforms:

- Client-side programming languages: HTML5, CSS3, and Python.

- Web application frameworks: flask

- Relational database management system: sqlalchamey.

**5.3 FUNCTIONAL REQUIREMENT**

This application having the 7 major Functions or Modules. It having a Login credentials. The major Functions are,

1) Details

2) Tutor

3)Class

4) Academics

5) Fees

6) Library

7) Hostel

**5.3.1 Details**

In this Function we will able to see the basic details about the student like name, father name, mother name, place etc.

**5.3.2 Tutor**

In this function we will able to see the details of students’s tutor, his/her contact details, mailing and calling function.

**5.3.3 Class**

In this function we will able to see the details of students’s classroom,building,lab room etc.

**5.3.4 Academics**

In this function we will able to see the academic details of students. such as his/her result, overall gpa, cgpa and coursewise grade for each semester.

**5.3.5 Fess**

In this function we will able to see the details of student’s pending tution fees details and other charges

**5.3.6 Library**

In this function we will able to see the details of book borrowed ,fine to be paid, renew status and old question papers.

**5.3.7 Hostel**

In this function we will able to see the details latest info of hostel, hostel fees to be paid, mess fees and fine charges only if the student is hosteller.

**6 USE CASE MODEL**

**6.1** **Use Case #1 : Login Activity**

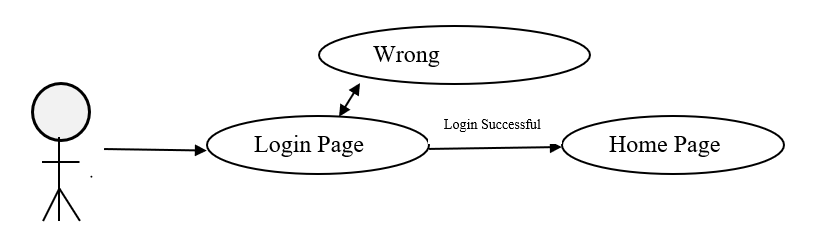


Figure 1: login

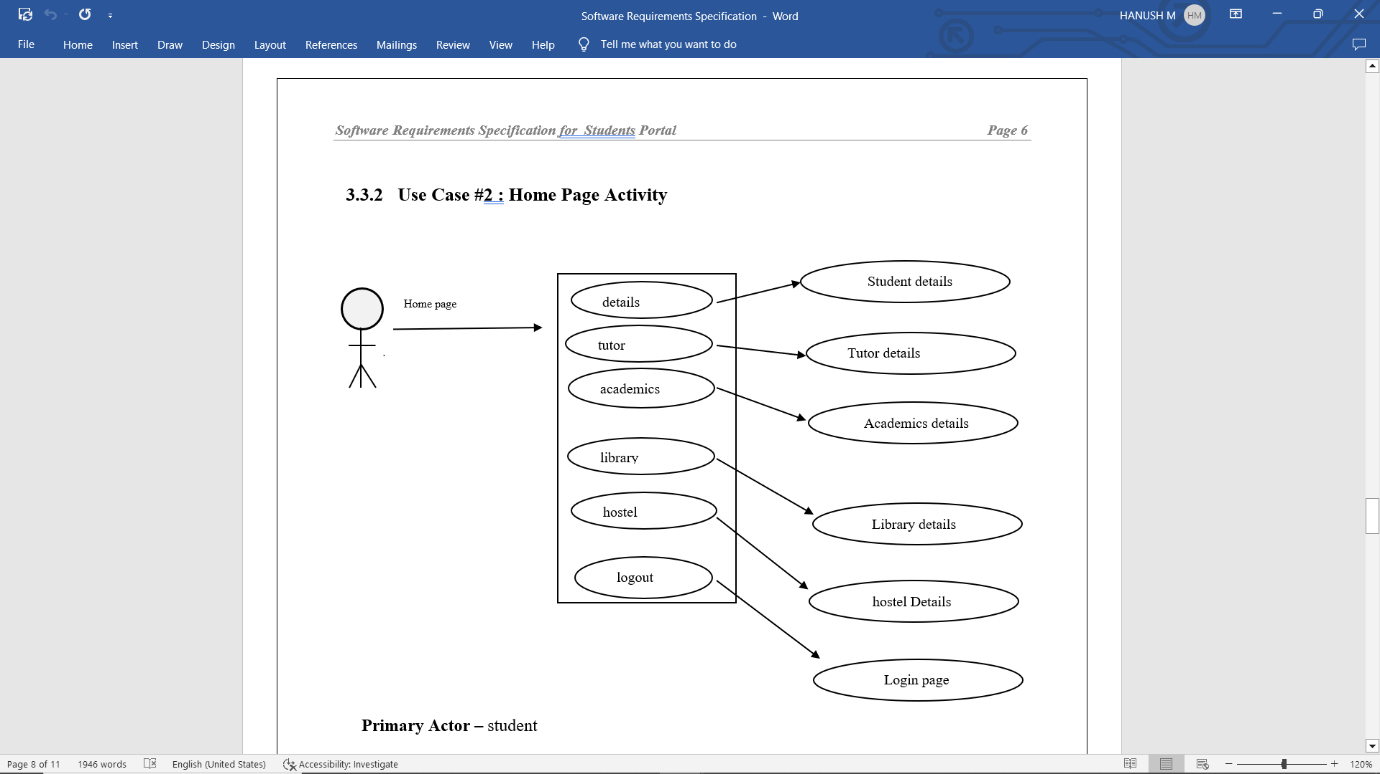
**6.2 Use Case #2 : Home Page Activity**

Figure 2:Home page

**6.3 Use Case #3 Details**

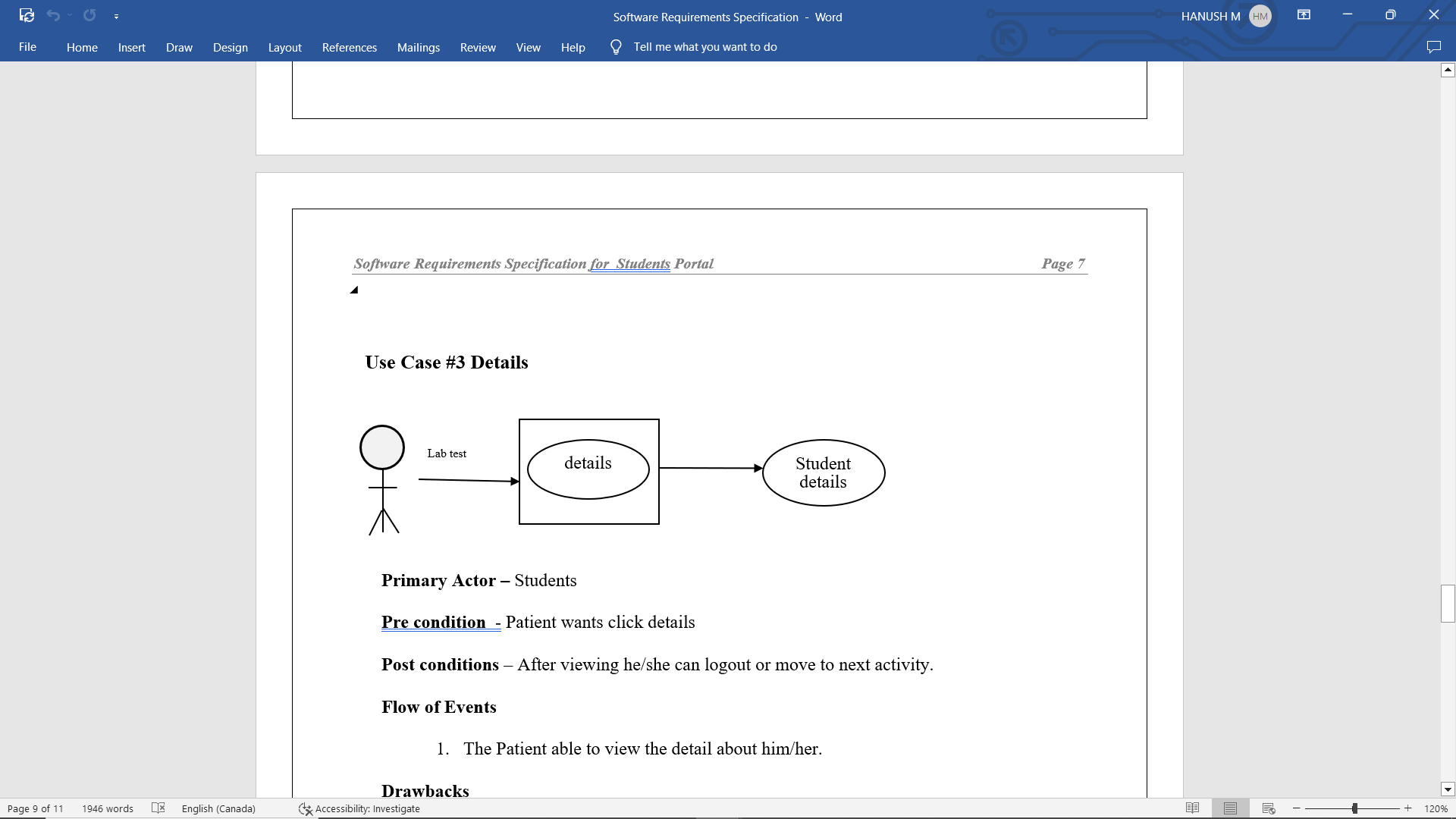


Figure 3:details page

**6.4 Use Case #4Tutor**

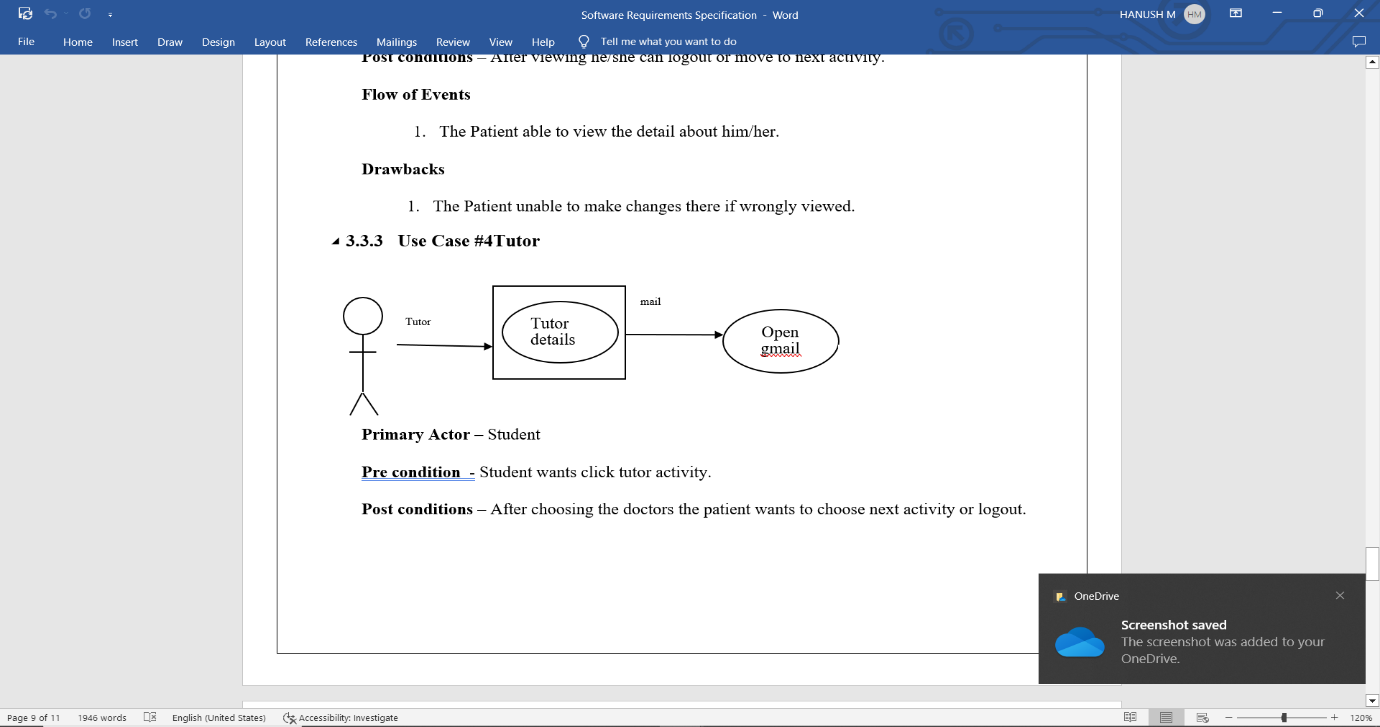


Figure 4:Tutor page

**6.5 Use Case #5 fees**

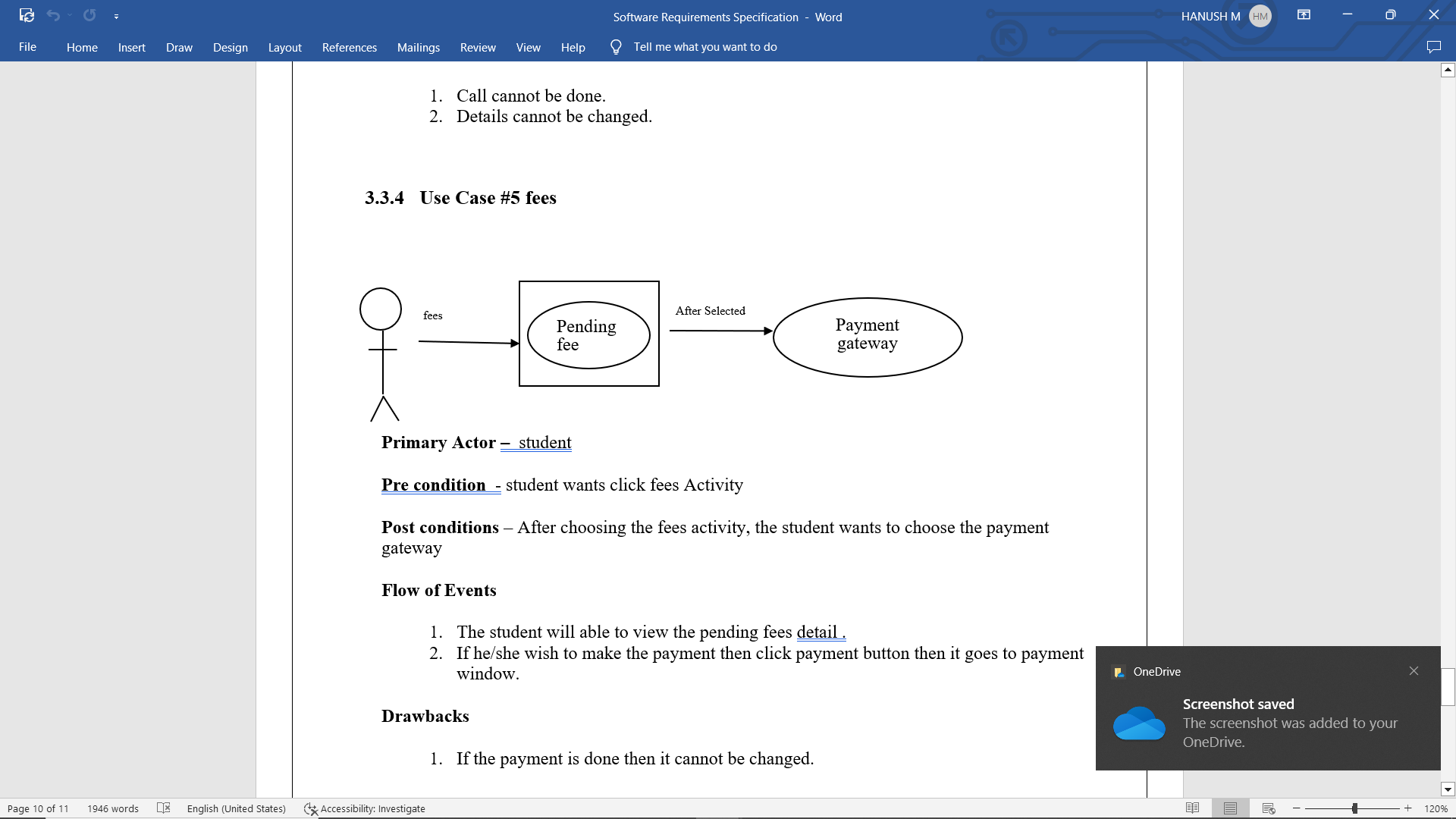
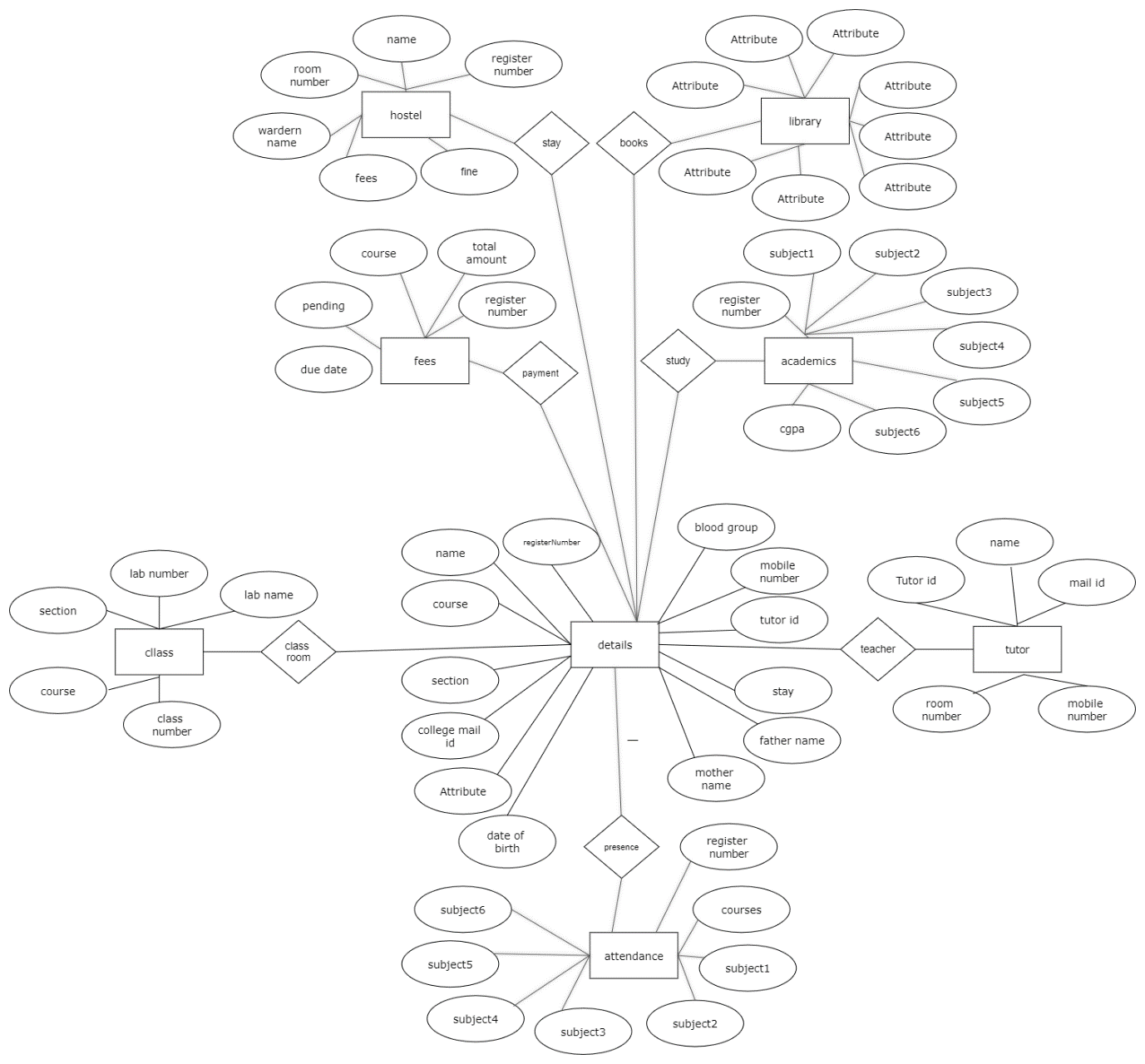


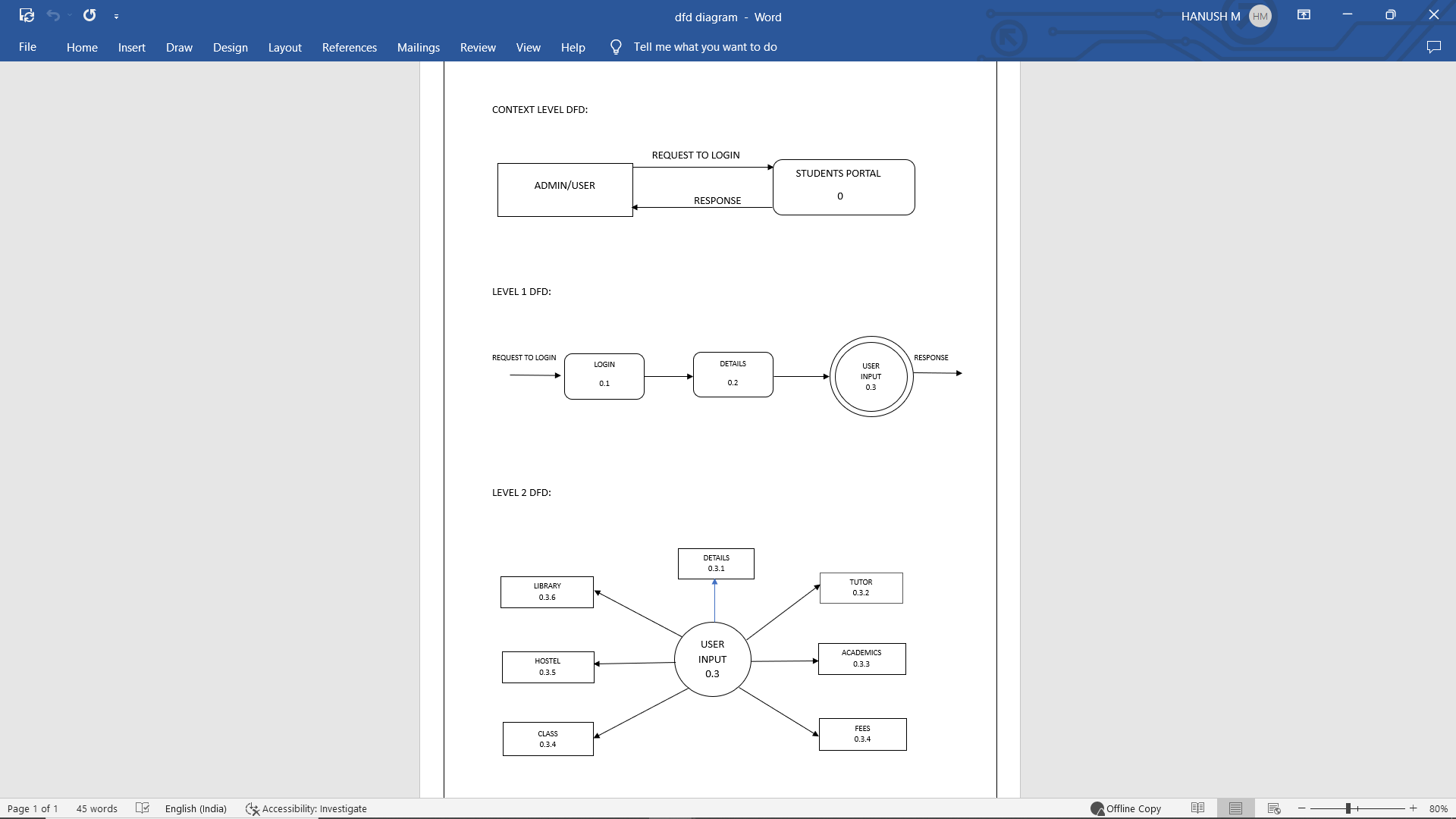
Figure 5:Fees page

**7.DESIGN**

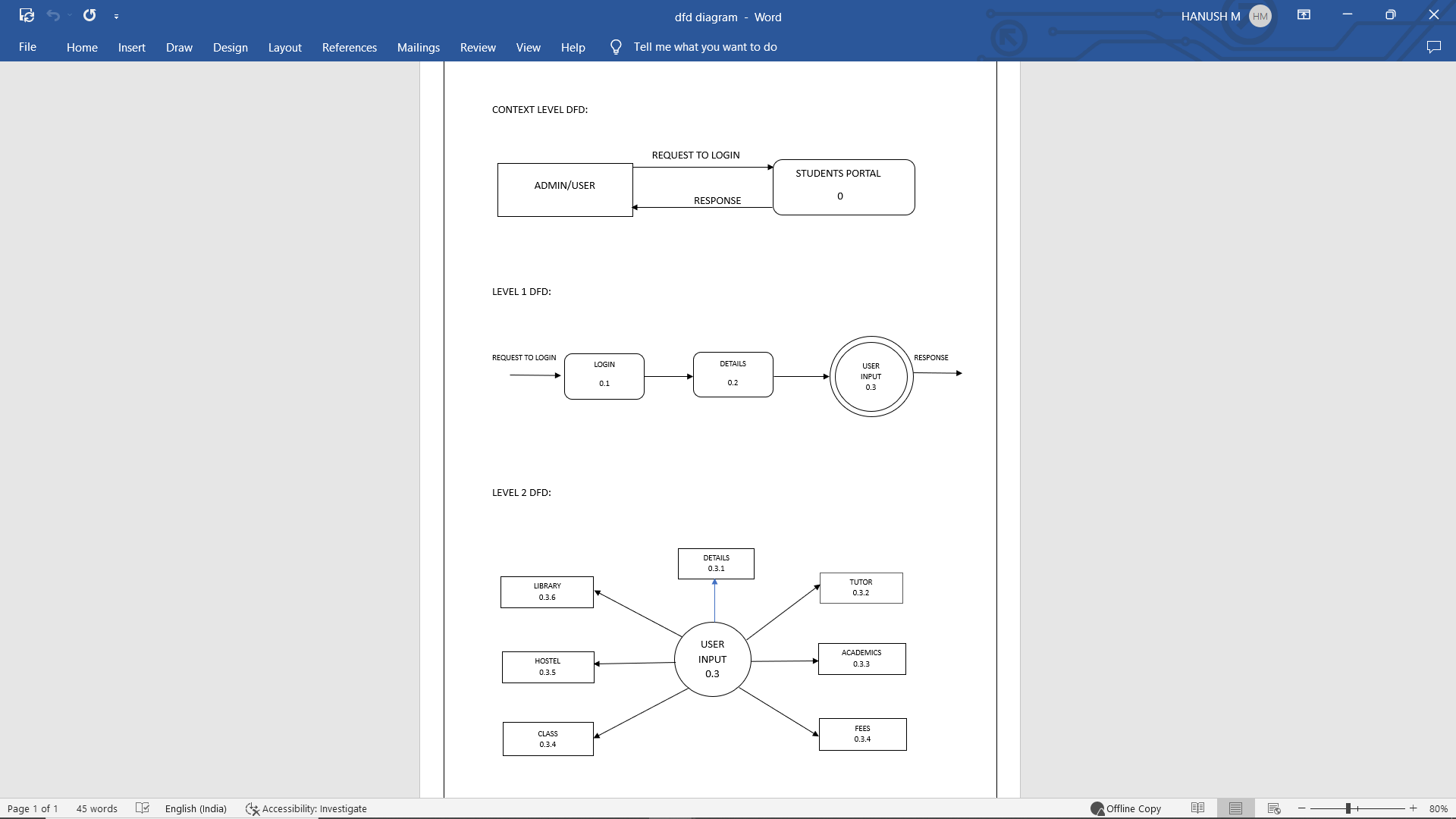
**7.1 ER DIAGRAM**

**7.2 DATA FLOW DIAGRAM**

**7.2.1 LEVEL 0**



**7.2.2 LEVEL 1**



**7.2.3 LEVEL 3**

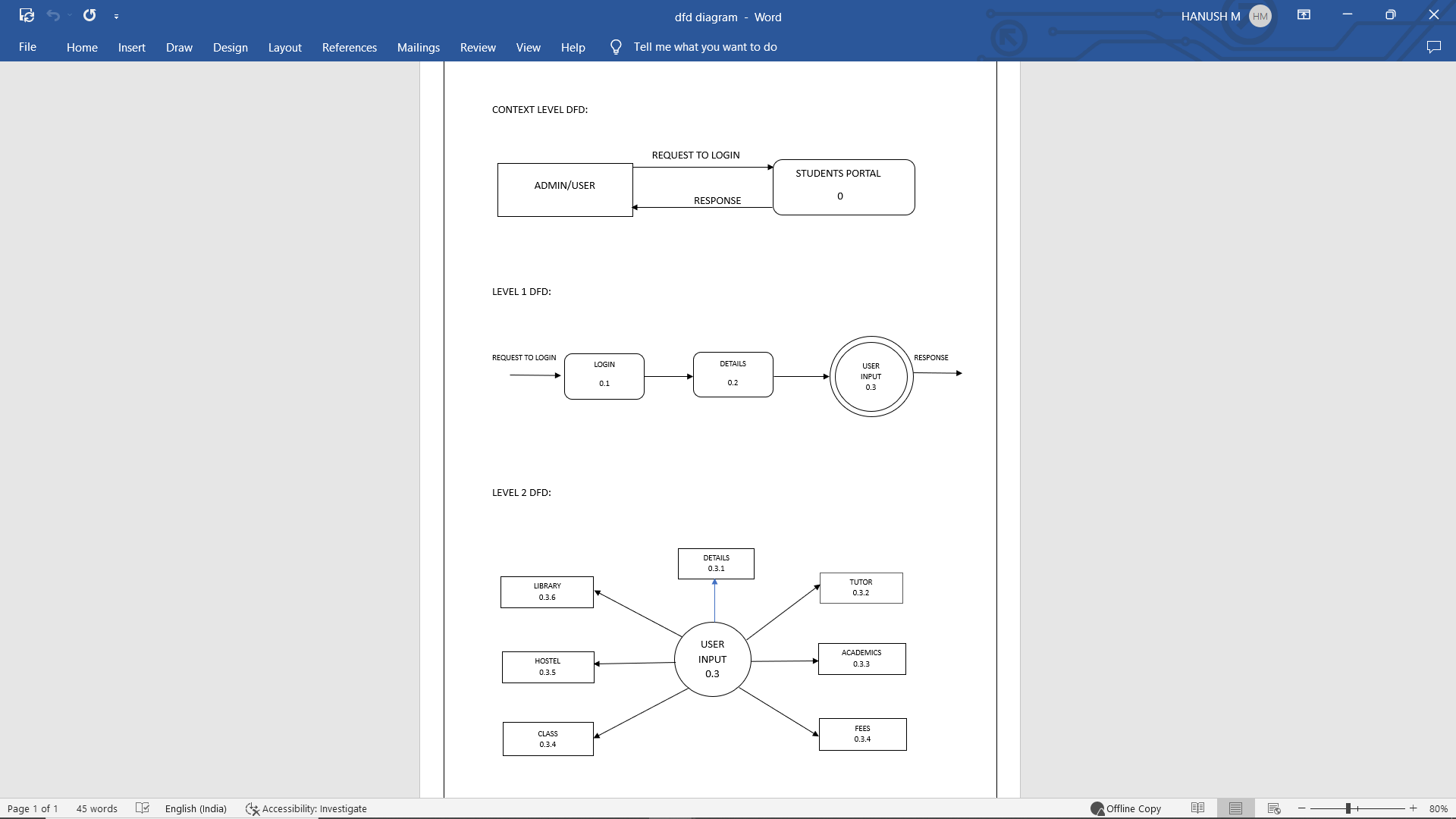


Figure 6:Context diagram

**8 MODULE DESCRIPTION**

**8.1 LOGIN ACTIVITY**

Primary Actor – student

Pre condition - The student will be issued login credential by respective college.

Post conditions – After successfully Login its goes to the Home page.

Flow of Events

1. The Patient wants to enter their Username and Password

2. If the username and password are correct then the page is redirected to Home page.

**8.2 HOME PAGE**

Primary Actor – student

Pre condition - student wants to Login

Post conditions – After completed all activity the patient wants to Logout.

Flow of Events

1. The Student will able to view the all activity.

2. He/ She click the activity as their wish then the page is redirected to respective activity.

3. The activities are details, tutor, academics, fees, library, hostel.

**8.3 DETAILS**

Primary Actor – Students

Pre condition - Patient wants click details

Post conditions – After viewing he/she can logout or move to next activity.

Flow of Events

1. The Patient able to view the detail about him/her.

Drawbacks

1. The Patient unable to make changes there if wrongly viewed.

**8.4 TUTOR**

Primary Actor – Student

Pre condition - Student wants click tutor activity.

Post conditions – After choosing the doctors the patient wants to choose next activity or logout.

Flow of Events

1. The Patient able to view the details of his/her tutor.

2. Then it shows contact details of tutor.

3. If he/she wish to make an mail to tutor then mail will be opened

Drawbacks

1. Call cannot be done.

2. Details cannot be changed.

**8.5 FEES**

Primary Actor – student

Pre condition - student wants click fees Activity

Post conditions – After choosing the fees activity, the student wants to choose the payment gateway

Flow of Events

1. The student will able to view the pending fees detail .

2. If he/she wish to make the payment then click payment button then it goes to payment window.

Drawbacks

1. If the payment is done then it cannot be changed.

**9.IMPLEMENTATION**

**9.1 LOGIN**

In login page one should provide username and password .if the username and password are correct then the we will we able to login and move to next page. If user forget the password he/she will be able to change the password using forget password.

This function open the login page initially.

@app.route('/')

def initail():

return render\_template('practice.html')

@app.route('/form\_login',methods=['POST','GET'])

This function check the username and password are correct and available in database.

def login():

name1=request.form['Username']

pwd=request.form['Password']

if name1 not in database:

return render\_template('practice.html',info='Invalid User')

else:

if database[name1]!=pwd:

return render\_template('practice.html',info='Invalid Password')

else:

database.clear()

database[name1]=pwd

a=db.session.query(details).all()

return render\_template('home.html', ptr=a,k=int(name1))

for forget password,

@app.route('/forget.html')

def forget():

return render\_template('forget.html')

@app.route("/form\_forget",methods=['POST','GET'])

def defpass():

regnum=request.form['registernumber']

mobnum=request.form['mobilenumber']

check=db.session.execute(login).filter\_by(username==int(regnum))

print(check)

**9.2 HOME PAGE**

After successful login user will able to choose his/her required option to view the details available in the sidebar panel.

@app.route('/details.html')

def detail():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(details).all()

return render\_template('details.html', ptr=a,k=k)

@app.route('/tutor.html')

def tutorr():

k=0

value=database.keys()

for i in value:

k=int(i)

a = db.session.query(tutor,details).join(details,tutor.tutorid==details.tutorid).all()

for i,j in a:

print(i.tutorid,j.registerno)

return render\_template('tutor.html',ptr=a,k=k)

@app.route('/cllass.html')

def cllasss():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(cllass,details).join(details,cllass.sec==details.sec).all()

for i,j in a:

print(i.cno,j.name)

return render\_template('cllass.html',ptr=a,k=k)

@app.route('/attendance.html')

def attendancee():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(attendance).all()

return render\_template('attendance.html',ptr=a,k=k)

@app.route('/academics.html')

def academicss():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(academics).all()

return render\_template('academics.html',ptr=a,k=k)

@app.route('/fees.html')

def feess():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(fees).all()

return render\_template('fees.html',ptr=a,k=k)

@app.route('/library.html')

def libraryy():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(library).all()

return render\_template('library.html',ptr=a,k=k)

@app.route('/hostel.html')

def hostell():

k=0

value=database.keys()

for i in value:

k=int(i)

a=db.session.query(hostel).all()

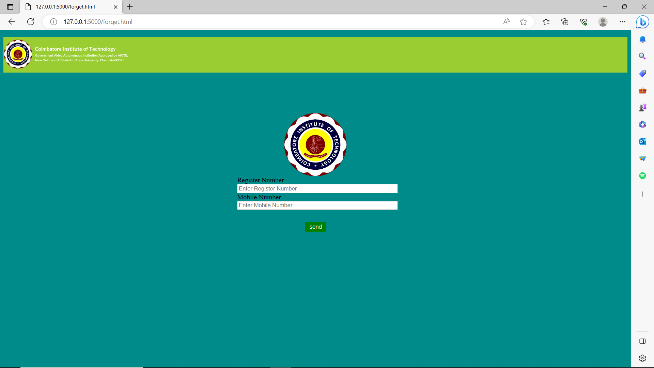
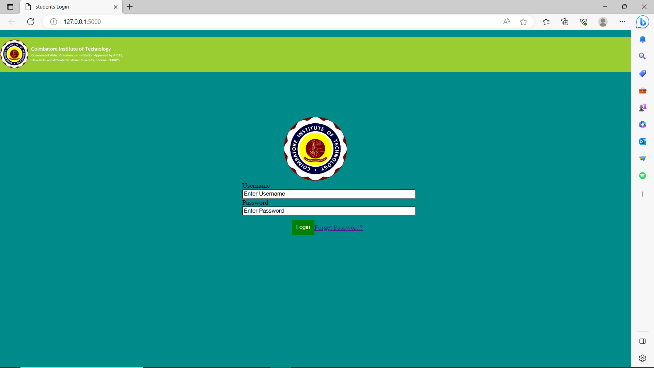
return render\_template('hostel.html',ptr=a,k=k)

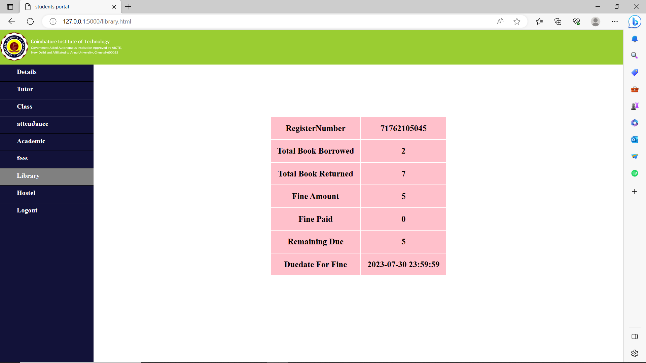
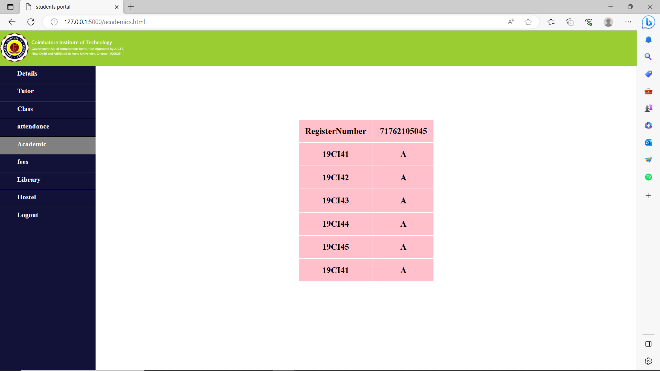
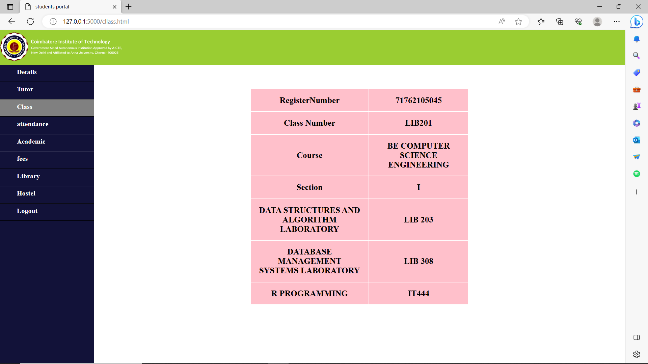
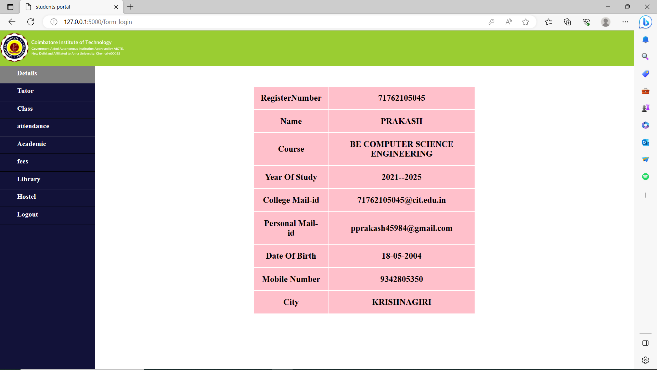
@app.route('/practice.html')

def practice():

return render\_template('practice.html')

**10.RESULT**

****

****

**11.TESTING**

**11.1 LOGIN TESTING**

**Test case 1:**

Input:

Username:71762105022

Password:Mhanush@21

Output:

Login successful and enter into home page.

(search the username and password in the database if it exists then able to

login)

**Test Case2:**

Input:

Username:71762105022

Password:Mhanush

Output:

Invalid password so unable to enter into home page.

(search the username and password in the database if it doesn’t exists then

unable to login)

**Test Case3:**

Input:

Username:717621050

Password:Mhanush@21

Output:

Invalid username so unable to enter into home page.

(search the username and password in the database if it doesn’t exists then

unable to login)

**11.2 FORGET PASSWORD**

**Test case 1:**

Input:

Username:71762105022

Mobile Number:9043148825

Output:

Password will be sent as message to registered mobile number if both are correct.

**Test Case2:**

Input:

Username:71762105022

Password:90341

Output:

Invalid mobile number so unable to send message.

**Test Case3:**

Input:

Username:717621050

Password:9043148825

Output:

Invalid username so unable send message.

**12.CORPORATE SOCIAL RESPONSIBILITY**

When developing a student portal web project, there are several social responsibilities that should be considered to ensure the well-being and inclusivity of the student community. Here are some key aspects of social responsibility in student portal web development:

**Accessibility:** Make sure the website is accessible to students with disabilities. Follow web accessibility guidelines and standards, such as the Web Content Accessibility Guidelines (WCAG), to ensure that students with visual, hearing, or mobility impairments can navigate and access the portal's content.

**Inclusivity:** Design the portal to be inclusive of all students, regardless of their background, ethnicity, gender, or abilities. Avoid any discriminatory or exclusionary practices in the design, content, or functionalities of the portal.

**Privacy and Data Protection:** Safeguard the personal information and data of students. Implement strong security measures to protect sensitive data, such as grades, attendance records, or personal contact information, and ensure compliance with relevant data protection regulations.

**Content Relevance and Accuracy:** Provide accurate and up-to-date information through the portal. Regularly update content such as academic schedules, course materials, campus news, and resources, ensuring that students have access to reliable and relevant information.

**Sustainable Practices:** Consider incorporating sustainable practices in the development and maintenance of the portal. Optimize the website's performance, minimize energy consumption, and follow eco-friendly coding and hosting practices to reduce the environmental impact.

**13.FUTURE IMPLEMENTATIONS**

**By Centralized Database:**

To make the common database servers with the help of Firebase, aws or any other Servers to maintain the all records/data .

**Online Payment method:**

To make a payment for the Appointment for hostel fee, mess, tution and exam fees through online by connecting the online payment platform like UPI,G-Pay and so.

**Assignment**

To make assignment page so that teachers can assign assignment through this and students will be able to upload the assignment.

**14.REFERENCES**

1) IEEE 830 Template for Document Template.

2) Core Python Programming - Python for programming the Software.

3) W3Schools- HTML and CSS For Front End