

**A PROJECT REPORT ON**

# **WEB APPLICATION FOR TERMWORK SUBMISSION**

**By:**

**Parth Baudhanwala (roll no: CE012 id:17CEUBS013)**

**Meet Charola (roll no: CE017 id: 17CEUOS071)**

**B.Tech. CE Semester – VI**

**Subject: System Design Practice**

**Guided By:**

**Prof. Pandav K. Patel**

**Assistant Professor**

**Department of Computer Engineering**



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**



**Faculty of Technology**  
**Department of Computer Engineering**  
**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical/termwork carried out in the subject of System Design Practice and recorded in this journal is the bonafied work of

- 1) Parth Baudhanwala (17CEUBS013)**
- 2) Meet Charola (17CEUOS071)**

Of Department of **Computer Engineering, Semester VI**, academic year 2019-20, under your supervision and guidance.

---

**Guide**

**Prof. Pandav K. Patel**  
Assistant Professor of  
Department of Computer  
Engineering, Dharmsinh Desai  
University, Nadiad.

**HOD**

**Dr. C. K. Bhensdadia**  
Head of the Department of  
Department of Computer  
Engineering, Dharmsinh Desai  
University, Nadiad.

## Table of Contents

1. Abstract.....	4
2 Introduction.....	5
2.1 Project Details: Brief Introduction .....	5
2.2 Technology and Tools Used .....	6
3 Software Requirement Specifications .....	7
3.1 Scope .....	7
3.2 System Functional Requirements .....	7
3.3 Other-Non-Functional Requirements.....	8
4. Design.....	9
4.1 Use-Case-Diagram.....	9
4.2 Class-Diagram.....	10
4.3 Sequence-Diagram.....	10
4.4 Activity Diagram.....	11
4.5 E-R Diagram .....	14
4.6 data Dictionary.....	14
5 Implementation Details .....	15
5.1 Modules .....	15
5.2 Functional prototypes .....	17
6 Testing .....	20
6.1 Testing Method.....	20
6.2 Test Cases .....	20
7. Screen-shots of the System .....	21
8. Conclusion .....	23
9. Limitations and Future Extensions of System .....	24
10 .Bibliography .....	24

# 1. Abstract

---

As we are going through the digital era of these years, we need think towards educational functions that can be converted digitally. So we as a team, have created this web application to submit term-work online. This will eventually help us to reduce paper work and also save our time to keep tracking of details of students. Our main purpose to build this web application is to facilitate submission process and also solve storage problems. In this web application student can submit particular assignment to particular lab of according subject. And as we mentioned before, we have solved storage problems so there is automated storage system to maintain accurate records.

## 2. Introduction

---

### **2.1 Brief Introduction**

Our inspiration of this project is Moodle, which is currently in use by our university. We enhanced Moodle to our project with better user interface & better performance.

Faculties can create subjects which is associated with students. They can create labs & in particular lab they can create assignments & problem statements with the particular subject. Faculties can add lab-start-date & assignment-end-date. Before assignment-end-date students must have to submit their termwork through our website, if student miss the assignment-end-date, then his/her termwork will not be accepted. Faculty can also extend the assignment-end-date.

### **2.2 Technologies & Tools Used**

#### **Technologies:**

C#

ASP.NET

MVC 5

MySQL

Entity Framework 6.x

#### **Tools:**

Microsoft Visual Studio 2019

NuGet Package Manager

## **Platforms:**

IIS Express

localhost:44369

Local Machine

## 3. Software Requirement Specifications

---

### 3.1 Product Scope

This website provides you to upload or submit your termwork, assignment or solution of the problem statements before due date of the submission which is created by faculties to the local machine of the faculties. Also this website is providing to perform create, edit & delete of student, subject, lab & assignment. Enrol students to particular subject or delete enrolled students to particular subject is also provided.

### 3.2 Types of Users

1. Student
2. Faculty

### 3.3 System Functional Requirements

#### **R.1: Student**

##### **R.1.1: Manage Account**

###### **R.1.1.1: Login**

**DESCRIPTION:** Student enters email-id and password required for logging into his account.

**INPUT:** Email-ID & Password.

**OUTPUT:** Student is logged in.

###### **R.1.1.2: Logout**

**DESCRIPTION:** Enables student to logout.

**INPUT:** Student's selection.

**OUTPUT:** Student is logged out.

#### **R.1.2: View Subjects**

**DESCRIPTION:** Students can see all the subjects in which she or he is enrolled.

**INPUT:** Click on view subjects button.

**OUTPUT:** All subjects will be appeared.

#### **R.1.3: View Labs**

**DESCRIPTION:** Student can see all the labs related to subjects.

**INPUT:** Click on view labs button.

**OUTPUT:** All labs will be appeared.

#### **R.1.4: View Assignments**

**DESCRIPTION:** Students can see all the assignments related to labs.

**INPUT:** Click on view assignments button.

**OUTPUT:** All assignments will be appeared.

#### **R.1.5: Upload File**

**DESCRIPTION:** Students can see all the term-work related to assignments.

**INPUT:** Click on upload file button.

**OUTPUT:** Term-work will be submitted.



## **R.2: Faculty**

### **R.2.1: Manage Account**

#### **R.2.1.1: Login**

**DESCRIPTION:** Faculty enters email-id and password required for logging into his account.

**INPUT:** Email-ID & Password.

**OUTPUT:** Faculty is logged in.

#### **R.2.1.2: Logout**

**DESCRIPTION:** Enables student to logout.

**INPUT:** Student's selection.

**OUTPUT:** Student is logged out.

### **R.2.2: Manage Students**

#### **R.2.2.1: View Students**

**DESCRIPTION:** Faculty can view all the students which is created by him/her or other faculties.

**INPUT:** Click on view students button

**OUTPUT:** List of all students will be appeared.

#### **R.2.2.2: Create Student**

**DESCRIPTION:** Faculty can create new student.

**INPUT:** Name, Email-ID, Branch, Division, Batch, Semester, Roll-No, Date of Birth, Password.

**OUTPUT:** A new student will be created.

#### **R.2.2.3: Edit Student**

**DESCRIPTION:** Faculty can edit existing student.

**INPUT:** Name, Email-ID, Branch, Division, Batch, Semester, Roll-No, Date of Birth, Password.

**OUTPUT:** Existing student will be edited.

#### **R.2.2.4: Delete Student**

**DESCRIPTION:** Faculty can delete existing student.

**INPUT:** Click on delete button.

**OUTPUT:** Existing student will be deleted.

#### **R.2.2.5: Enrol Student**

**DESCRIPTION:** Faculty can enrol the existing student by batch wise with particular subject which is created by him/her.

**INPUT:** Select subject, branch, division, batch.

**OUTPUT:** Existing student will be enrolled to the subject.

#### **R.2.2.6: Delete Enrolled Student**

**DESCRIPTION:** Faculty can delete existing enrolled student by batch wise with particular subject which is created by him/her.

**INPUT:** Select subject, branch, division, batch.

**OUTPUT:** Existing enrolled student will be deleted to the subject.

### **R.2.3: Manage Subjects**

#### **R.2.3.1: View Subjects**

**DESCRIPTION:** Faculty can view all the subjects which is created by him/her.

**INPUT:** Click on view subjects button

**OUTPUT:** List of all subjects will be appeared.

#### **R.2.3.2: Create Subjects**

**DESCRIPTION:** Faculty can create new subject.

**INPUT:** Name & Semester.

**OUTPUT:** A new subject will be created.

#### **R.2.3.3: Edit Subjects**

**DESCRIPTION:** Faculty can edit existing subject.

**INPUT:** Name & Semester.

**OUTPUT:** Existing subject will be edited.

#### **R.2.3.4: Delete Subjects**

**DESCRIPTION:** Faculty can delete existing subject.

**INPUT:** Click on delete button.

**OUTPUT:** A existing subject will be deleted.

### **R.2.4: Manage Labs**

#### **R.2.4.1: View Labs**

**DESCRIPTION:** Faculty can view all the labs which is created by him/her related to the subject.

**INPUT:** Click on view labs button.

**OUTPUT:** List of all labs will be appeared

#### **R.2.4.2: Create Labs**

**DESCRIPTION:** Faculty can create new lab related to the subject.

**INPUT:** Lab-No & Lab-Start-Date.

**OUTPUT:** A new lab will be created.

#### **R.2.4.3: Edit Labs**

**DESCRIPTION:** Faculty can edit existing lab.

**INPUT:** Lab-No & Lab-Start-Date.

**OUTPUT:** Existing lab will be edited.

#### **R.2.4.4: Delete Labs**

**DESCRIPTION:** Faculty can delete existing lab.

**INPUT:** Click on delete button.

**OUTPUT:** A existing lab will be deleted.

## **R.2.5: Manage Assignments**

### **R.2.5.1: View Assignments**

**DESCRIPTION:** Faculty can view all the assignments which is created by him/her related to the lab.

**INPUT:** Click on view assignments button

**OUTPUT:** List of all assignments will be appeared.

### **R.2.5.2: Create Assignments**

**DESCRIPTION:** Faculty can create new assignment related to the lab.

**INPUT:** Assignment-No & Assignment-End-Date.

**OUTPUT:** A new assignment will be created.

### **R.2.5.3: Edit Assignments**

**DESCRIPTION:** Faculty can edit existing assignment.

**INPUT:** Assignment-No & Assignment-End-Date.

**OUTPUT:** Existing assignment will be edited.

### **R.2.5.4: Delete Assignments**

**DESCRIPTION:** Faculty can delete existing assignment.

**INPUT:** Click on delete button.

**OUTPUT:** A existing assignment will be delete.

### **3.4 Other Non-functional Requirements**

#### **1. Performance**

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays. In case of opening App components, of popping error messages and saving the settings or sessions there is delay much below 3 seconds.

#### **2. Safety**

User details should be securely stored to the server. The main security concern is for user account hence proper login mechanism should be used to avoid hacking.

#### **3. Reliability**

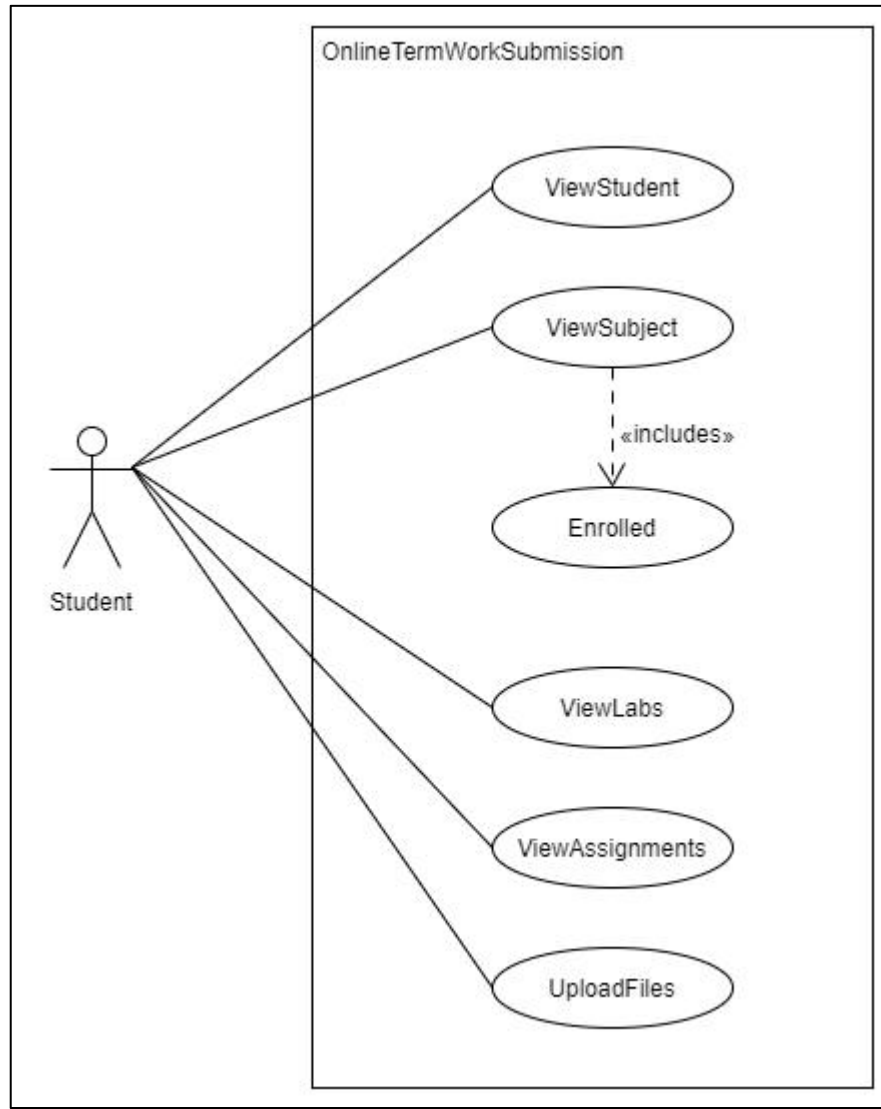
As the system provides the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

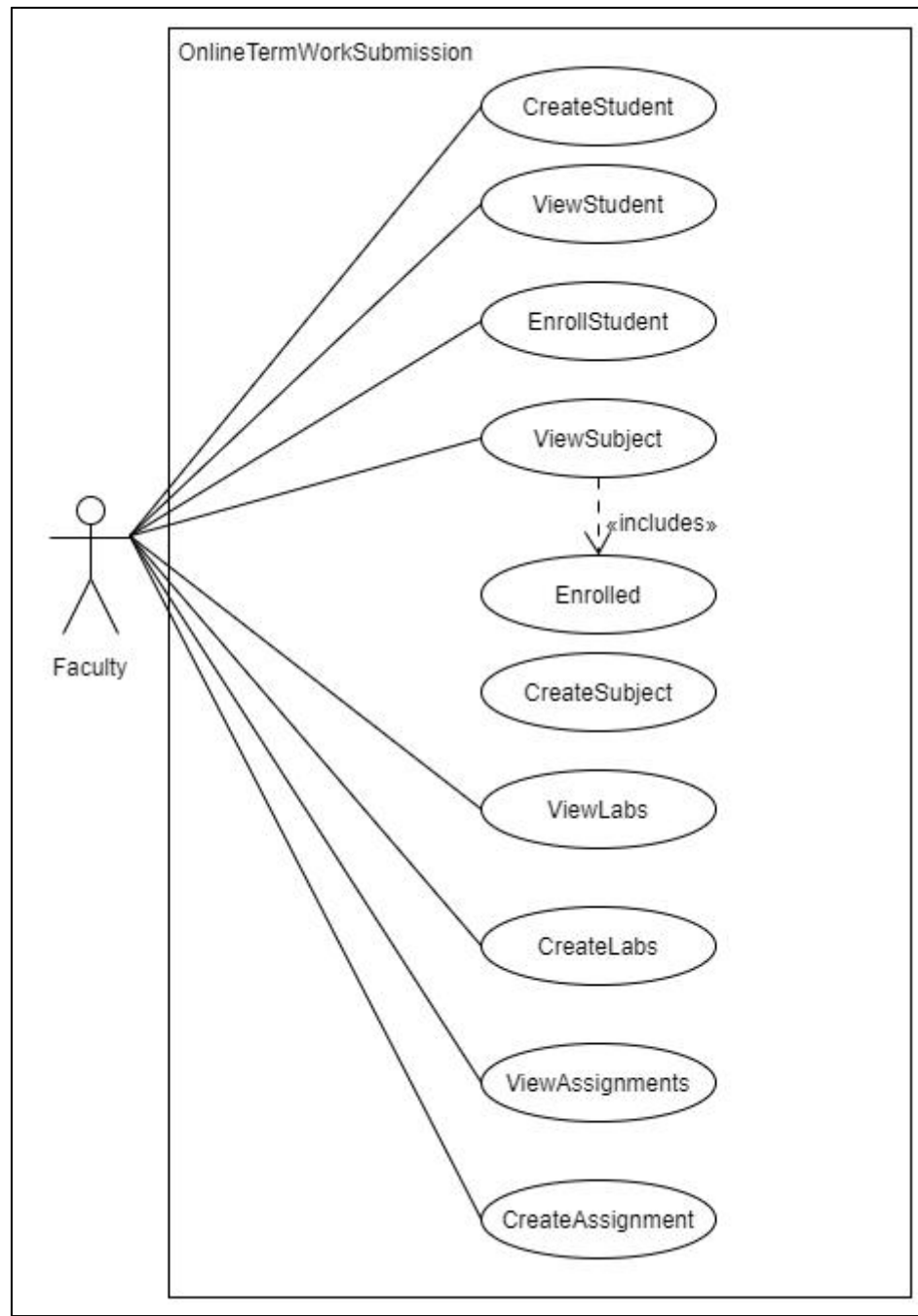
#### **4. Database**

System requires to access user's data fast to maintain the performance.

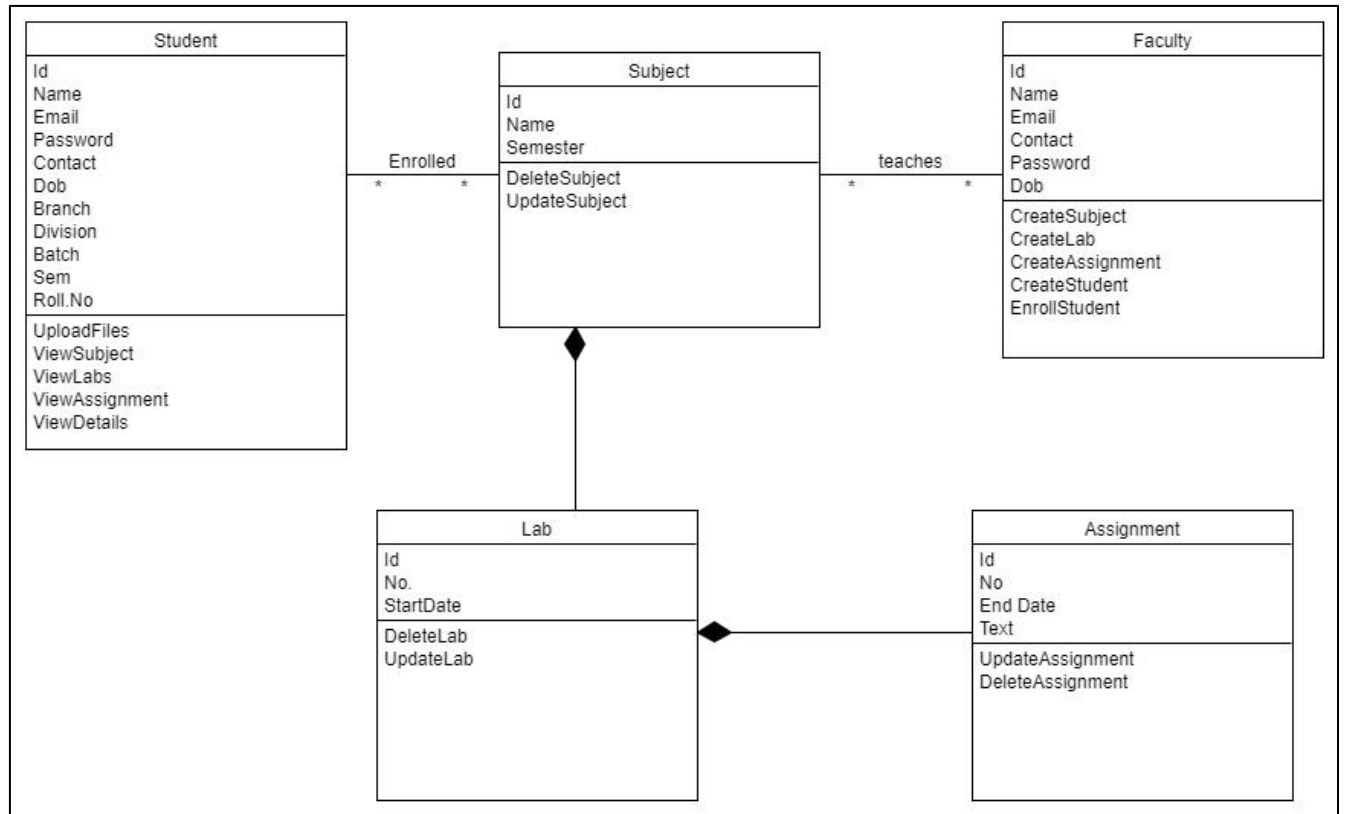
## 4. Design

### 4.1 Use Case Diagram



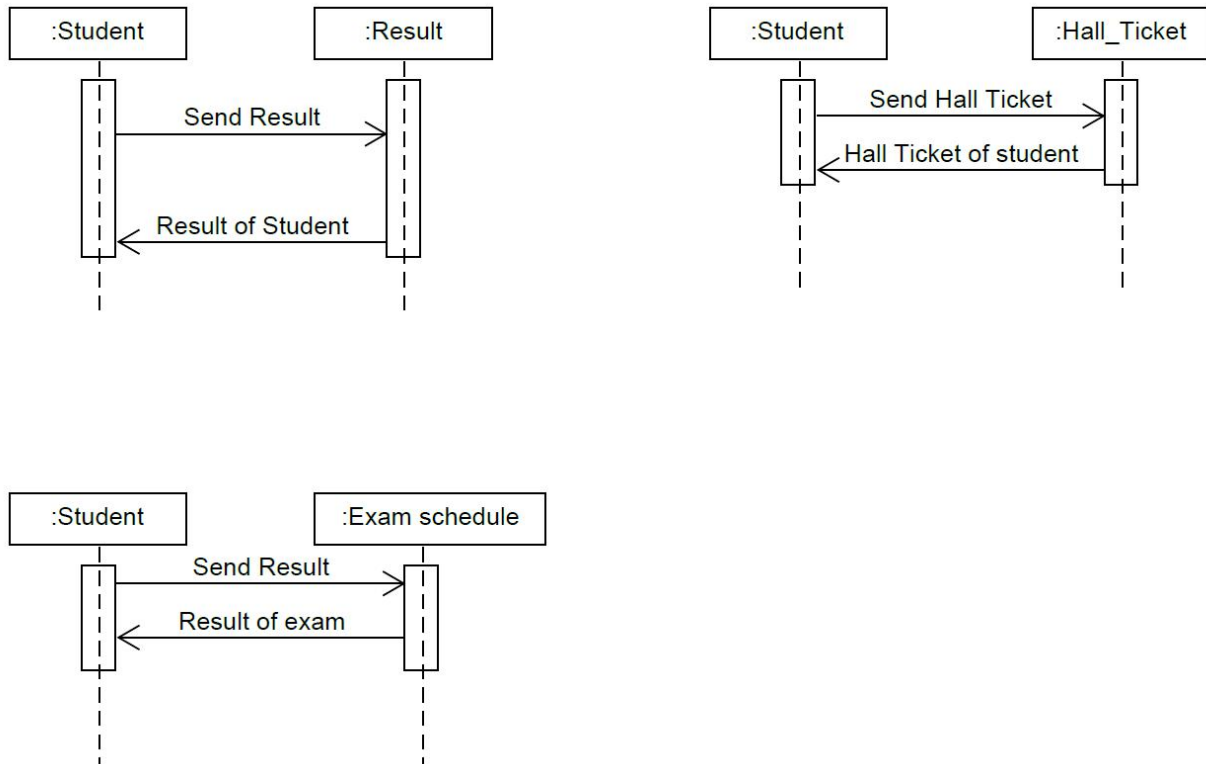


## 4.2 Class Diagram

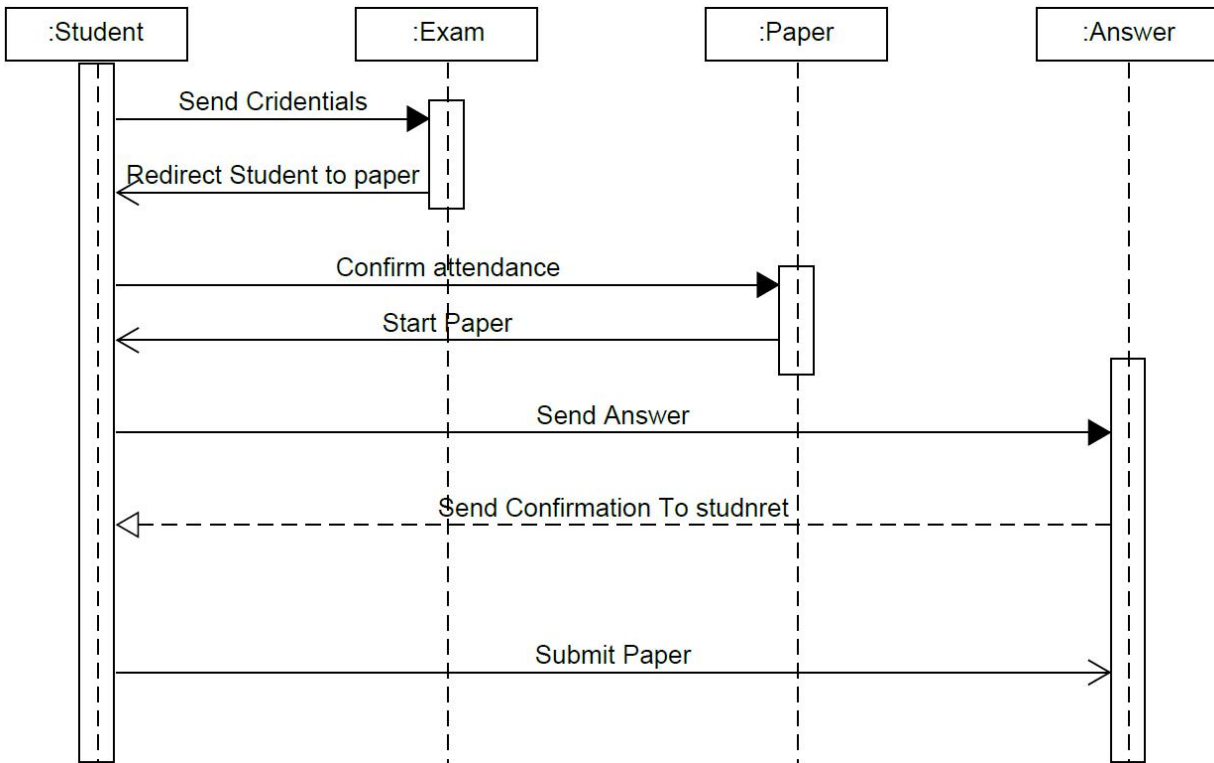




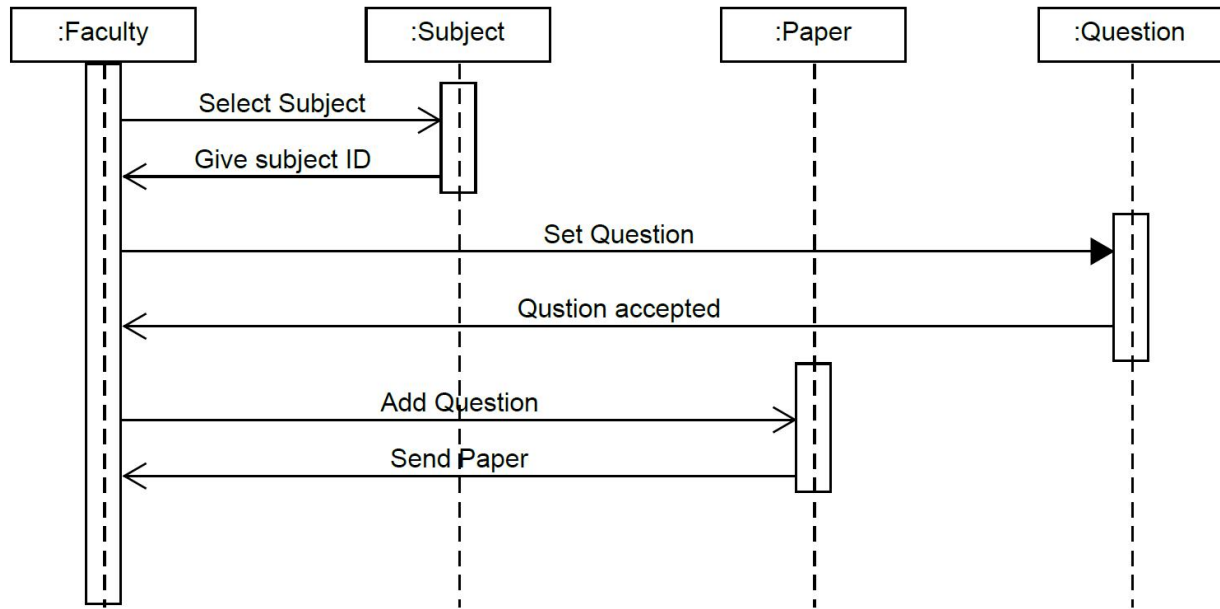
### **4.3 Sequence Diagram**



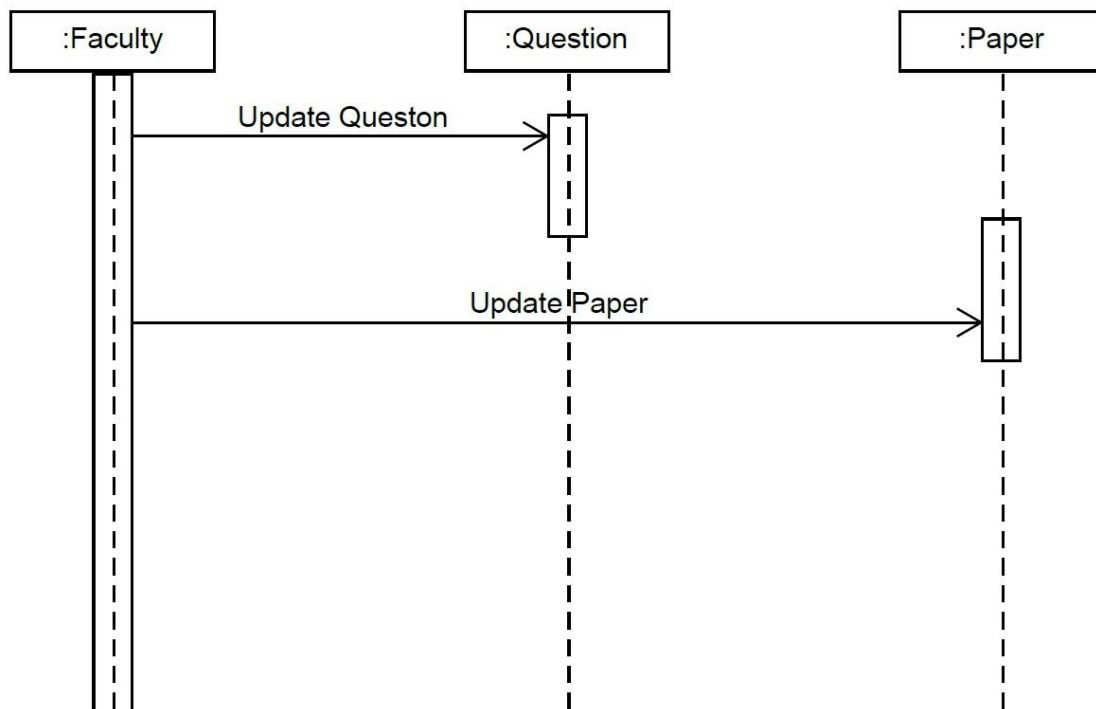
**Fig No:1. Sequence dia for 1.Result generation for student 2.Hall ticket generation for student 3.Exam Schedule for student**



**Fig No:2. Sequence between Student, Exam, Paper and Answer Object in Project**

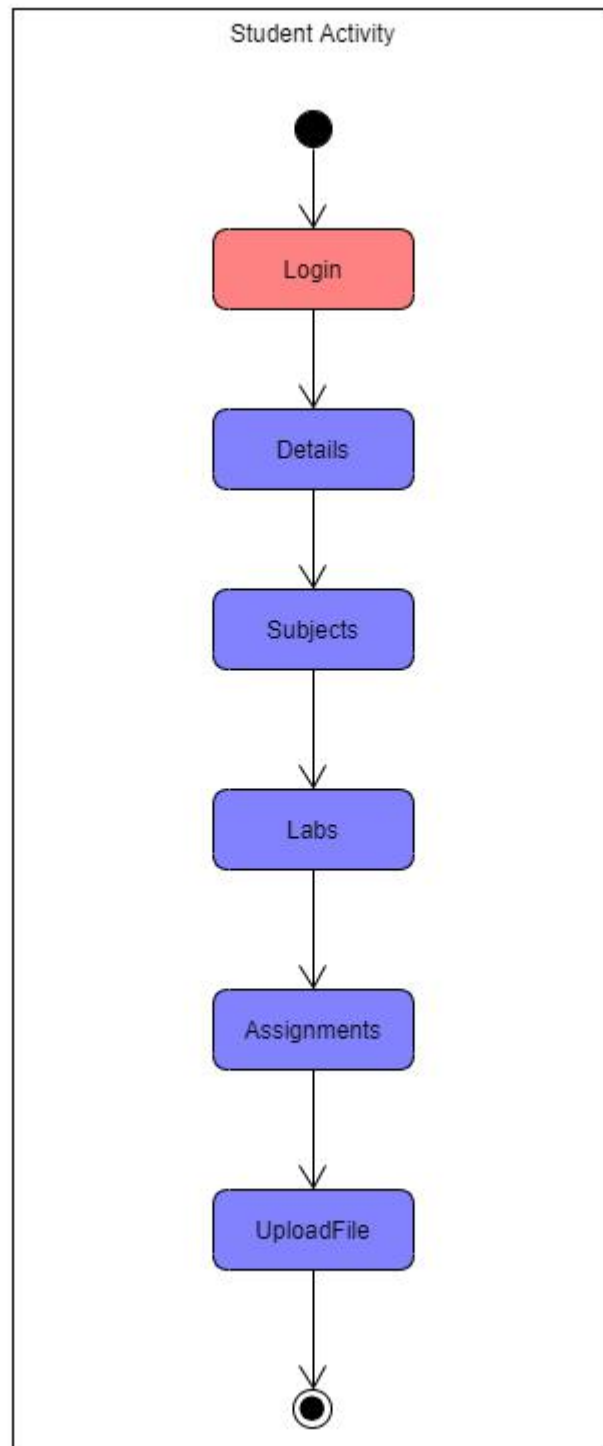


**Fig No:3. Sequence between Faculty, Subject, Paper and Question in Project**

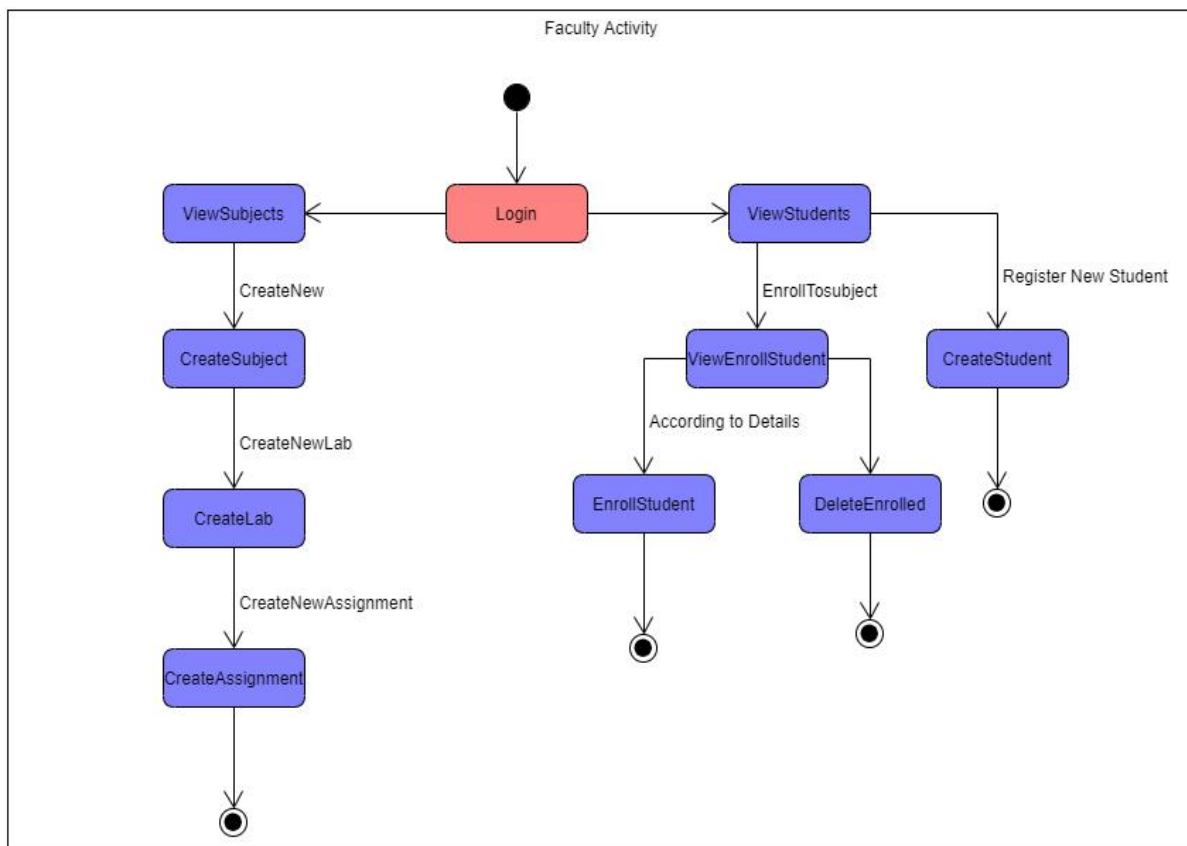


#### 4.4 Activity Diagram

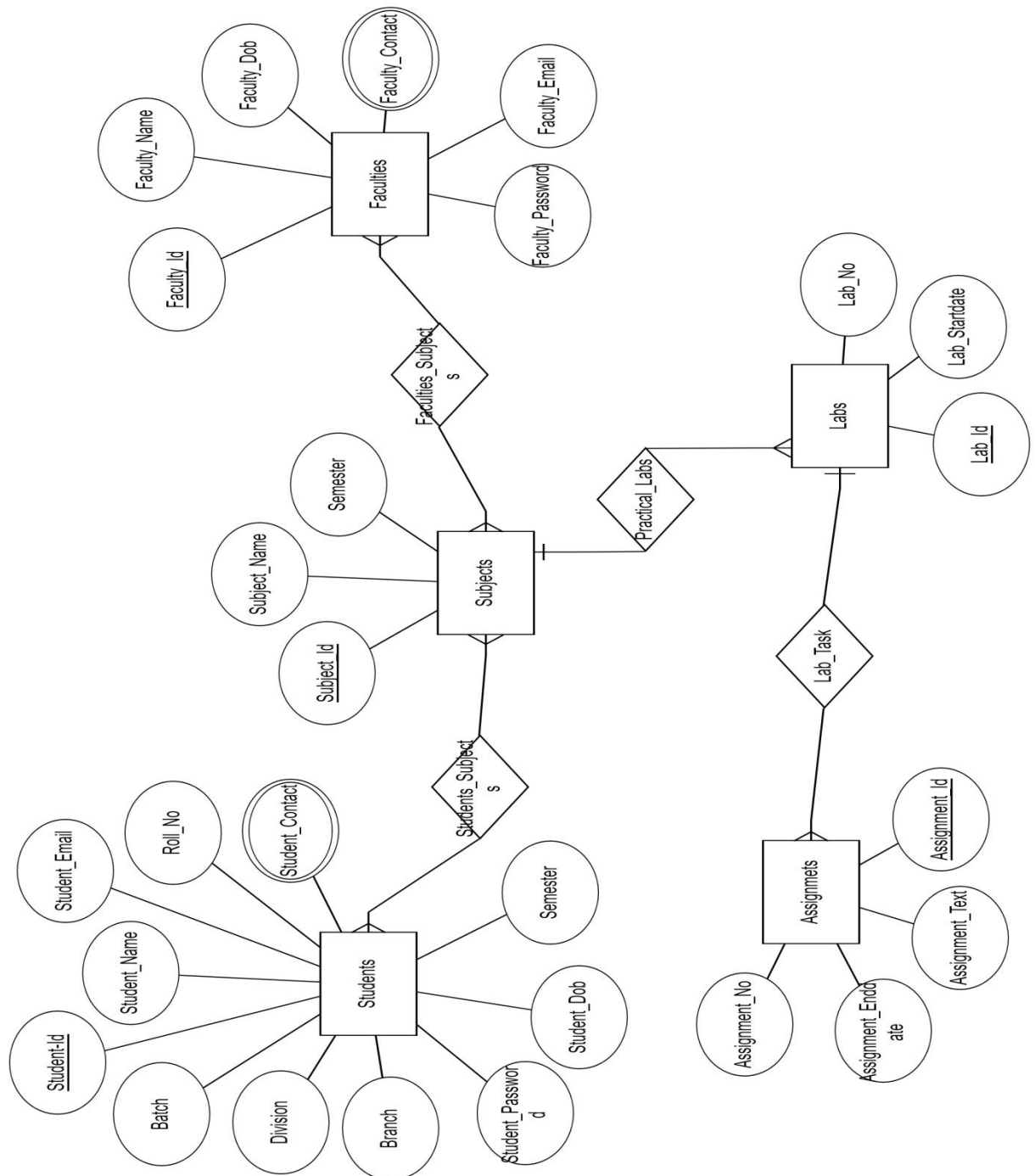
##### **STUDENT ACTIVITY:**



## FACULTY ACTIVITY:



## 4.5 E-R Diagram



## 4.6 Data Dictionary

### Data Dictionary Format For Online Term Work Submission

<b>Student</b>								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Referenced Table	Description
1	Student_Id	Int	10	Yes	Yes	PK	-	User id
2	Student_Name	Varchar	20	Yes	No	-	-	Full name
3	Student_Email	Varchar	30	Yes	Yes	-	-	Email id
4	Student_Password	Numeric	8	Yes	No	-	-	Phone number
5	Student_Dob	Date	10	Yes	No	-	-	Date of birth
6	Student_Contact	Numeric	10	Yes	No			Phone no.
7	Branch	Varchar	10	Yes	No			Branch Name
8	Semester	int	2	Yes	No			Semester No
9	Division	Varchar	5	Yes	No			Division Name
10	Batch	Varchar	5	Yes	No			Batch Name
11	Roll_No	Int	5	Yes	Yes			Roll Number

<b>Faculty</b>								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Faculty_Id	Int	10	Yes	Yes	PK	-	Faculty Id
2	Faculty_Name	Varchar	30	Yes	Yes	-	-	Faculty Name
3	Faculty_Email	Varchar	30	Yes	Yes	-	-	Faculty Email
4	Faculty_Password	Numeric	8	Yes	Yes	-	-	Password
5	Faculty_Contact	Numeric	10	No	Yes	-	-	Phone No.
6	Faculty_Dob	Date	-	Yes	No	-	-	Date Of Birth

<b>Subject</b>								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Subject_Id	Int	10	Yes	Yes	PK	-	Subject Id
2	Subject_Name	Varchar	30	Yes	Yes	-	-	Subject Name
3	Semester	Int	2	YES	No	-	-	Semester No.



Lab								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Lab_Id	Int	10	Yes	Yes	PK	-	Lab Id
2	Lab_No	Int	2	Yes	Yes	-	-	Lab No.
3	Lab_Startdate	Date	-	Yes	Yes	-	-	Lab Starting date
4	Subject_Id	Int	10	Yes	Yes	Fk	Subject	Subject Id

Assignment								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Assignment_Id	Int	10	Yes	Yes	PK	-	Assignment Id
2	Assignment_No	Int	2	Yes	Yes	-	-	Assignment No
3	Assignment_Text	Varchar	200	Yes	No	-	-	Description
4	Assignment_Enddate	Date	-	Yes	No	-	-	Last Date of submission
5	Lab_Id	Int	10	Yes	Yes	Fk	Lab	Lab Id

StudentSubjects								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Subject_Id	Int	10	Yes	Yes	FK	Subject	Subject Id
2	Student_Id	Int	10	Yes	Yes	Fk	Student	Student Id

FacultySubjects								
Sr. No	Field Name	Data Type	Width	Required	Unique	PK/FK	Reference Table	Description
1	Subject_Id	Int	10	Yes	Yes	FK	Subject	Subject Id
2	Faculty_Id	Int	10	Yes	Yes	Fk	Faculty	Faculty Id



## 5. Implementation Details

---

### **Main User Module:**

This module gives the main home page of the system which provides basic information about the website. It contains in page navigation to go to different parts of the pages. This module doesn't need authentication. It contains button for sign in and login.

**Input:** User Selection

**Output:** Corresponding response

### **Registration Module:**

The module is used to store user's data to the database and enables the use to login to the system. All the fields in this module contains required validations. User can also navigate to login page if he/she has already registered. This

**Input:** User's Information

**Output:** User Registered and redirect to login page

**Processing:** Validating user's data and then storing them to database

### **Login Module:**

This module takes users credentials and then verifies it with registered users, if user is not registered the invalid credentials error will be shown else if they match with database then user will be logged in.

**Input:** User Credentials

**Output:** Logging User.

**Processing:** Verifying user credentials with database.

### **User Home Module:**

This module is accessed only by authenticated user. It is user's home page. Users can manage their account. It provides user to select subject so that he/she can submit to their accordance.

**Input:** User Selection

**Output:** Corresponding Response

## 6. Testing

---

### 6.1: Testing Methods:

We have performed Black-box testing for the testing purpose.

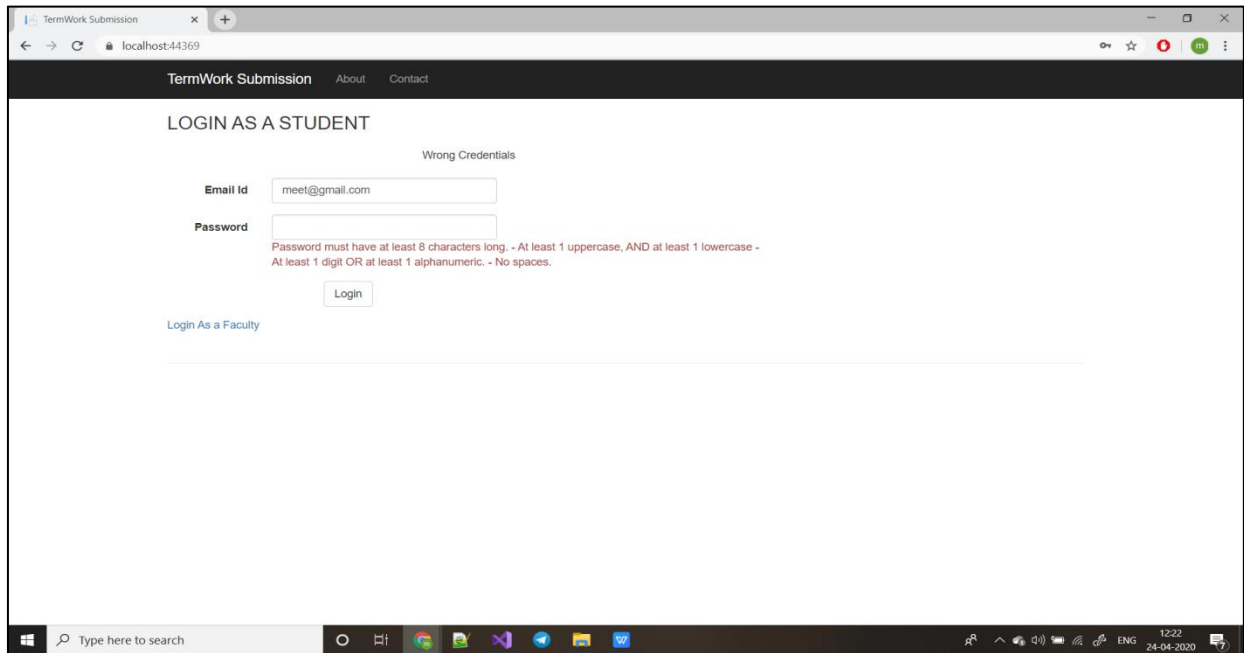
### 6.2: Test Cases:

NO.	Test Scenario	Expected Result	Actual Result	Pass/Fail
1	Login module is testing credentials if any invalidation is there then error message will be shown.	Wrong Credentials and valid password field message.	User is notified with wrong credential message and proper field message.	PASS
2	At the time of registration proper data-type and field validation testing, if any wrong message is there proper validation will be shown.	Proper Validation message will be shown.	User is notified with valid field data-type below field.	PASS
3	Testing on File Upload module for checking of deadline of assignment. If deadline is already passed than error message will be shown.	Sorry,You are out of deadline message will be shown.	User is notified with "sorry, You are out of deadline!!".	PASS
4	Automated Storage testing of assignment files	Files should be properly saved in the according folder of according file	File is saved properly.	PASS

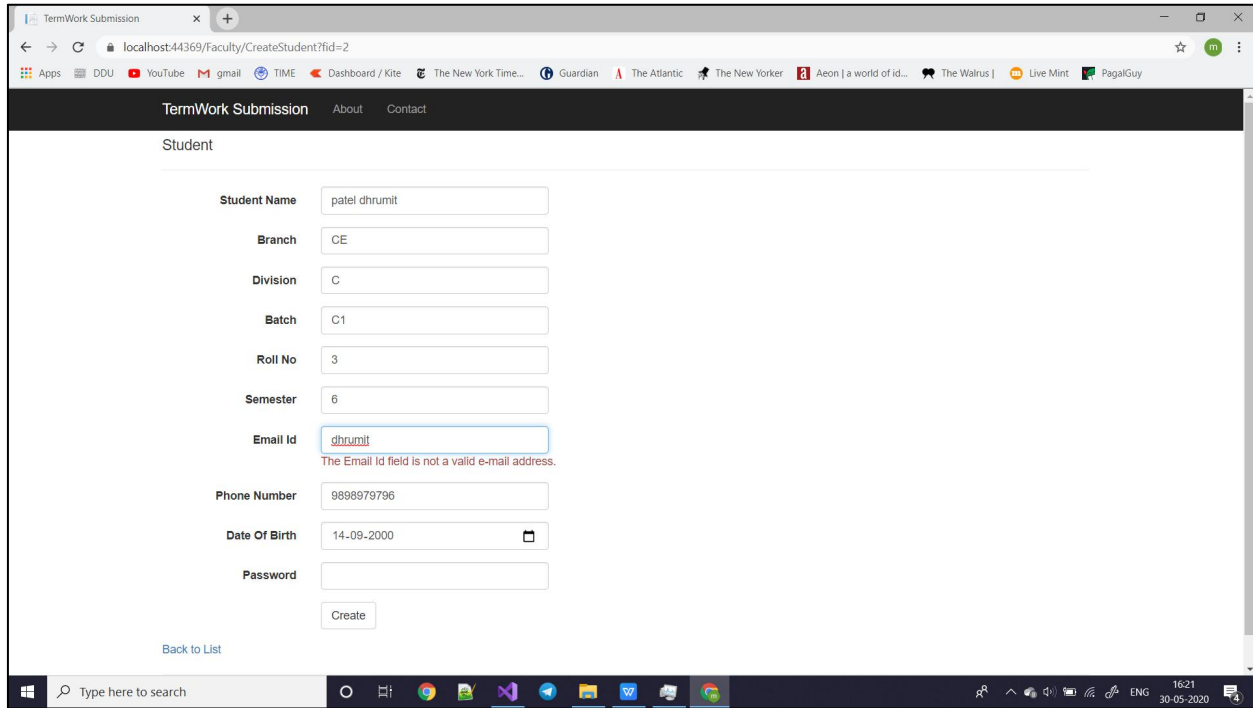
For Login:

There are validations such as every field is required and respective validations for every field, suppose Contact number is less than 10 digits then error comes.

**LOGIN:** If credentials are invalid then error message is shown.



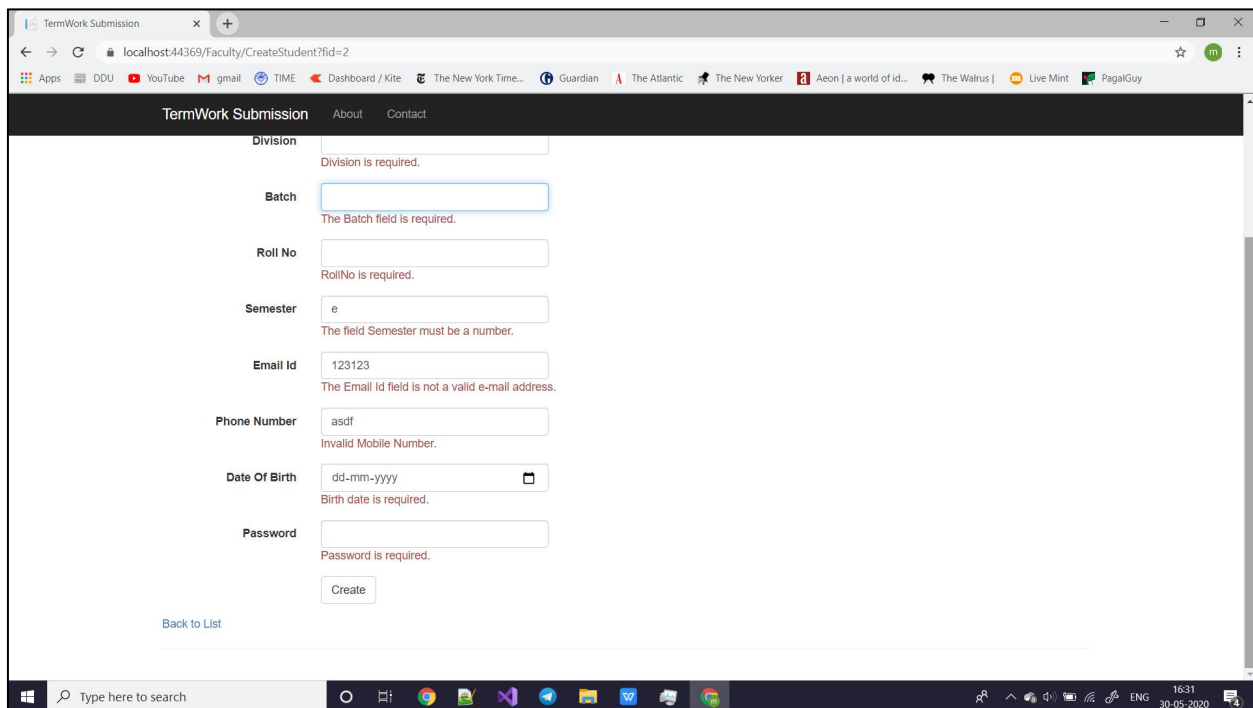
**REGISTER:** We have checked all field validation at the time of registration. If any data is not valid according to field, that field will show error message below field.



The screenshot shows a web browser window with the URL `localhost:44369/Faculty/CreateStudent?fid=2`. The page title is "TermWork Submission" and it has links for "About" and "Contact". The form is titled "Student" and contains the following fields:

- Student Name:
- Branch:
- Division:
- Batch:
- Roll No:
- Semester:
- Email Id:  (highlighted with a blue border)
- Phone Number:
- Date Of Birth:
- Password:

Below the Email Id field, there is a red error message: "The Email Id field is not a valid e-mail address." At the bottom of the form, there is a "Create" button and a "Back to List" link.



The screenshot shows the same web browser window, but with different data entered in the form fields, resulting in several validation errors:

- Division:  (empty field)
- Batch:  (empty field)
- Roll No:  (empty field)
- Semester:
- Email Id:
- Phone Number:
- Date Of Birth:
- Password:

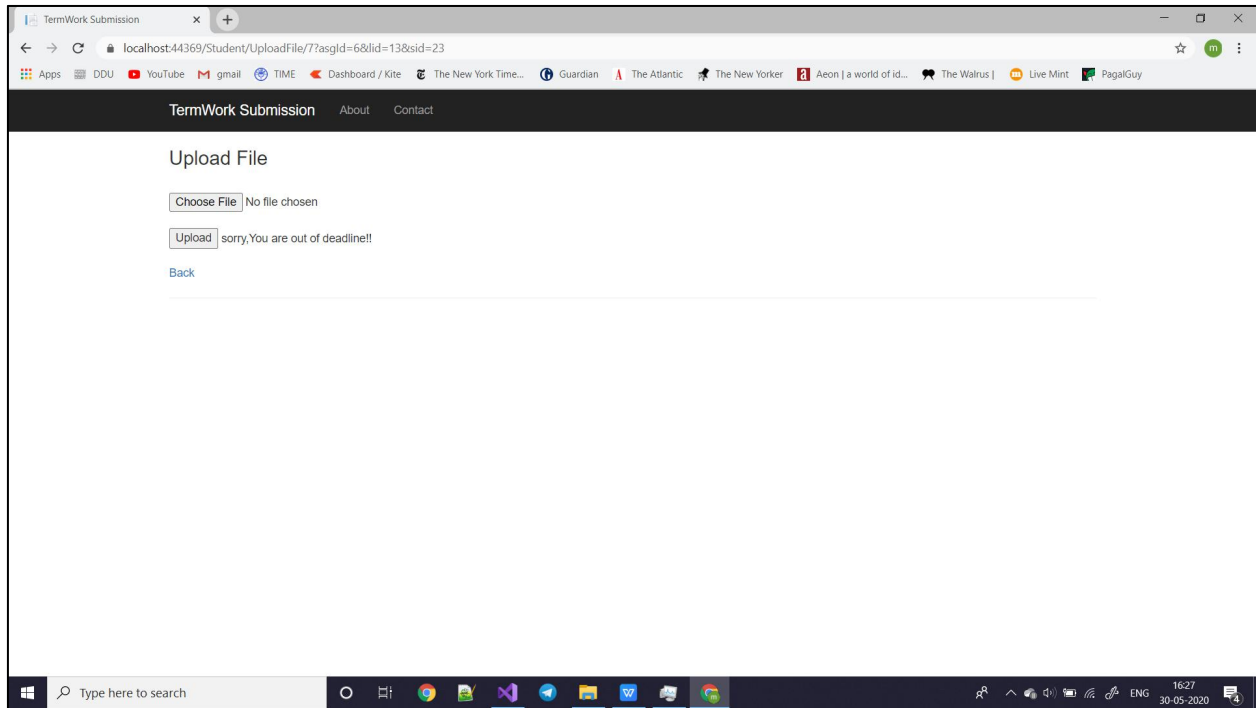
Red error messages are displayed below each field:

- Division: "Division is required."
- Batch: "The Batch field is required."
- Roll No: "RollNo is required."
- Semester: "The field Semester must be a number."
- Email Id: "The Email Id field is not a valid e-mail address."
- Phone Number: "Invalid Mobile Number."
- Date Of Birth: "Birth date is required."
- Password: "Password is required."

At the bottom of the form, there is a "Create" button and a "Back to List" link.

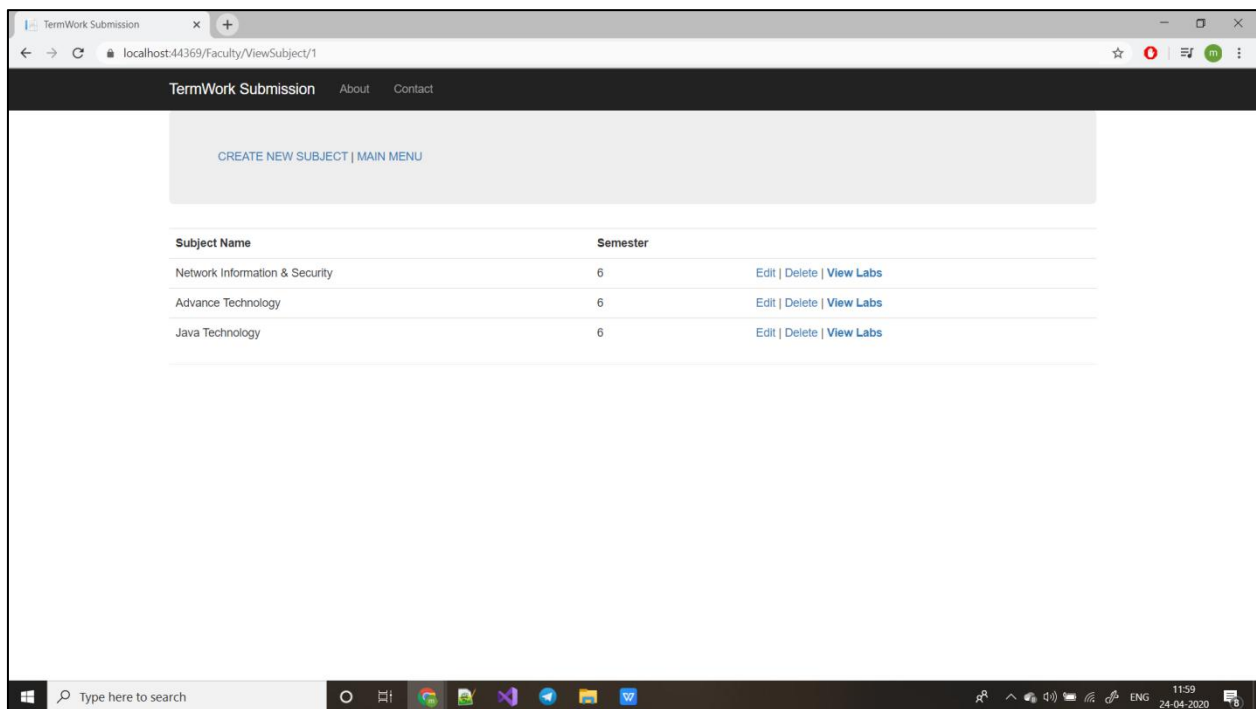
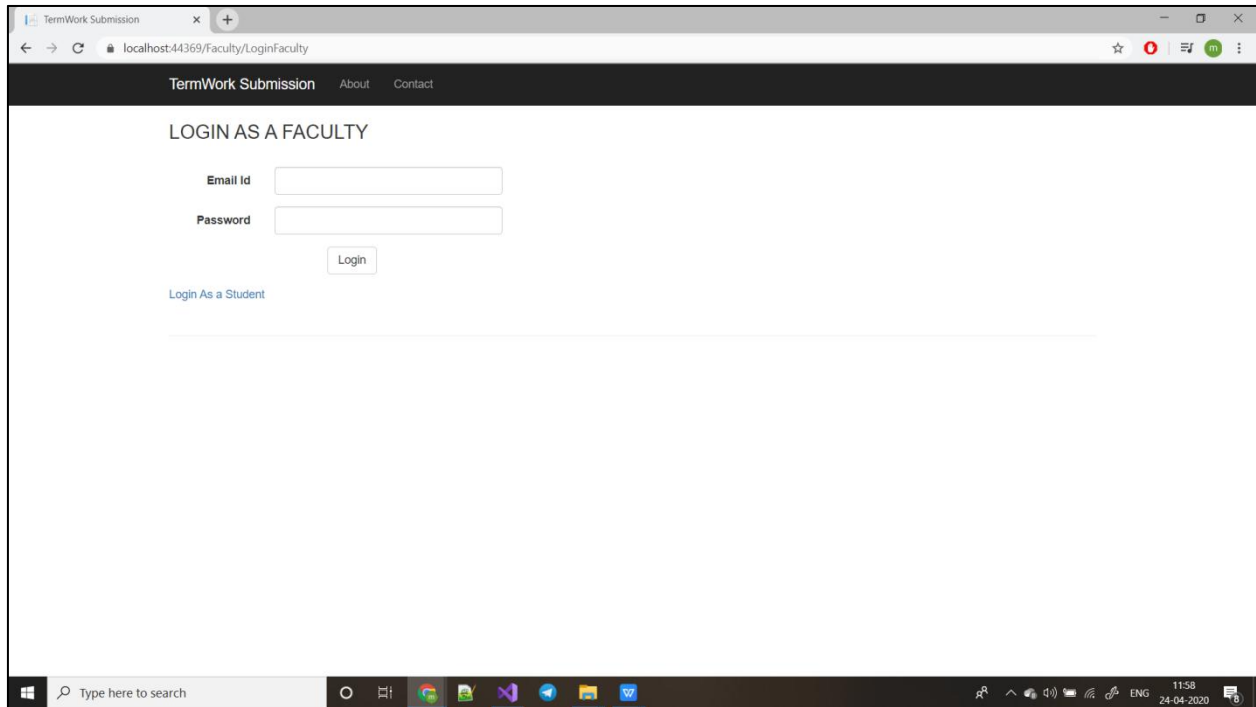
## **UploadModule:**

Whenever student upload file, we have created mechanism to check whether it is before deadline or not? If student is uploading file after the deadline than message will be shown as per below screenshot.



# 7. Screen Shot

## 1. Faculty Module



TermWork Submission

CREATE NEW STUDENT | SELECT STUDENTS TO ENROLL SUBJECTS | MAIN MENU

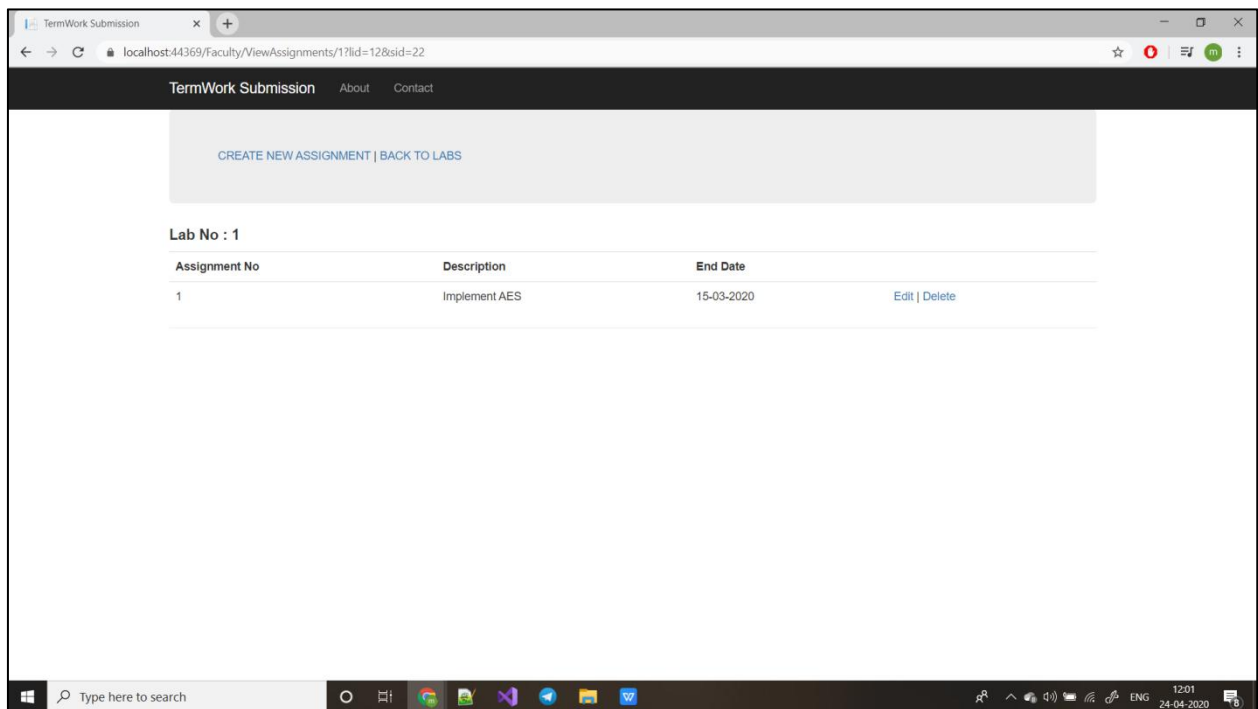
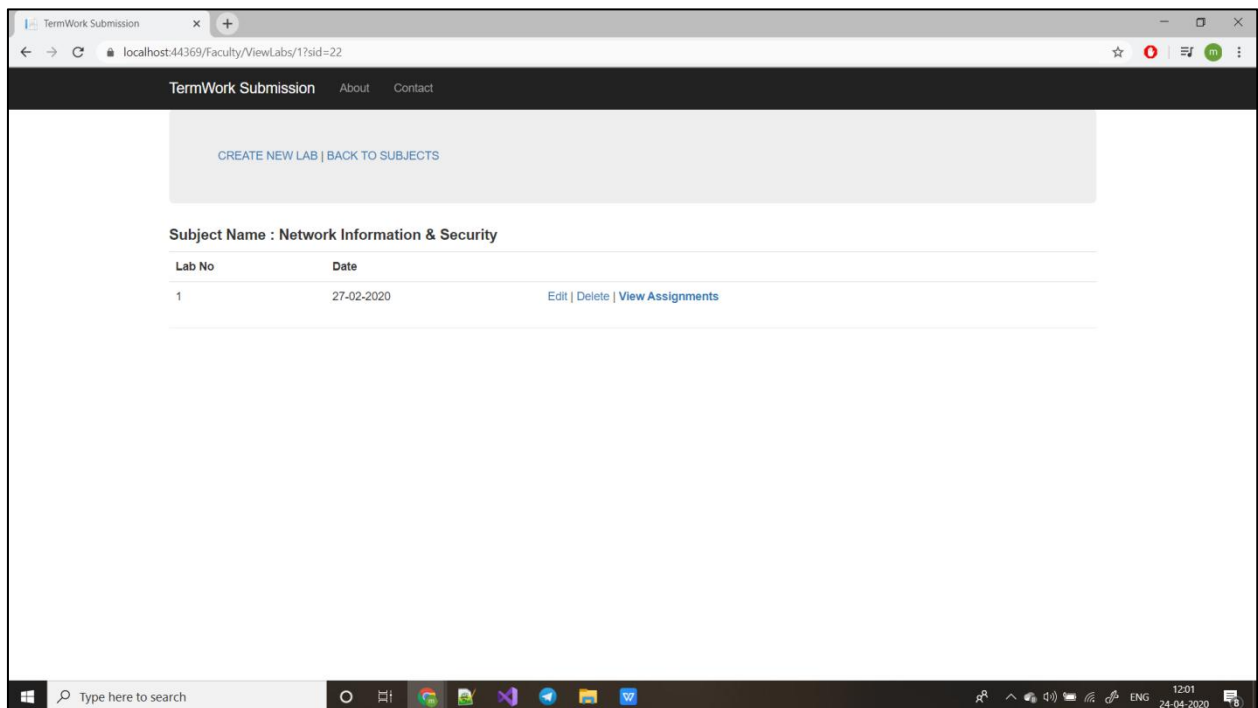
Roll No	Student Name	Branch	Semester	Division	Batch	Email Id	Phone Number	Date Of Birth	Password
1	Chintal Jain	CE	6	C	C1	chintu@gmail.com	9639639632	14-04-2020	Chintu@123 <a href="#">Edit</a>   <a href="#">Delete</a>
3	Aatish Caudhari	CE	6	C	C1	aatishc@gmail.com	9696969696	14-04-2020	Aatish@123 <a href="#">Edit</a>   <a href="#">Delete</a>
12	Parth Baudhanwala	CE	6	C	C1	parthbaudhanwala45@gmail.com	9879879879	08-10-1999	Parth@123 <a href="#">Edit</a>   <a href="#">Delete</a>
17	Meet Charola	CE	6	C	C2	meet@gmail.com	9898989898	09-03-2020	Meet@123 <a href="#">Edit</a>   <a href="#">Delete</a>
18	Jalmin Chauhan	CE	6	C	C2	jalmin@gmail.com	9870062130	04-03-2020	Jalmin@123 <a href="#">Edit</a>   <a href="#">Delete</a>
22	Harsh Kelawala	CE	6	C	C3	harsh@gmail.com	9870062130	29-02-2020	Harsh@123 <a href="#">Edit</a>   <a href="#">Delete</a>
25	Monil Modi	CE	6	C	C3	monil@gmail.com	9876543210	01-03-2020	Monil@123 <a href="#">Edit</a>   <a href="#">Delete</a>

TermWork Submission

BACK TO STUDENTS LIST

Subject Name	Semester	
Network Information & Security	6	<a href="#">View Enrolled Students</a>
Advance Technology	6	<a href="#">View Enrolled Students</a>
Java Technology	6	<a href="#">View Enrolled Students</a>





## 2. Student Module

The screenshot shows a web browser window with the title 'TermWork Submission'. The address bar shows 'localhost:44369'. The page has a dark header with 'TermWork Submission', 'About', and 'Contact' links. The main content area is titled 'LOGIN AS A STUDENT'. It contains two input fields: 'Email Id' and 'Password'. Below the 'Password' field is a 'Login' button. There is also a link 'Login As a Faculty' below the login fields. The Windows taskbar is visible at the bottom with the search bar and various application icons.

TermWork Submission About Contact

### LOGIN AS A STUDENT

Email Id

Password

Login

[Login As a Faculty](#)

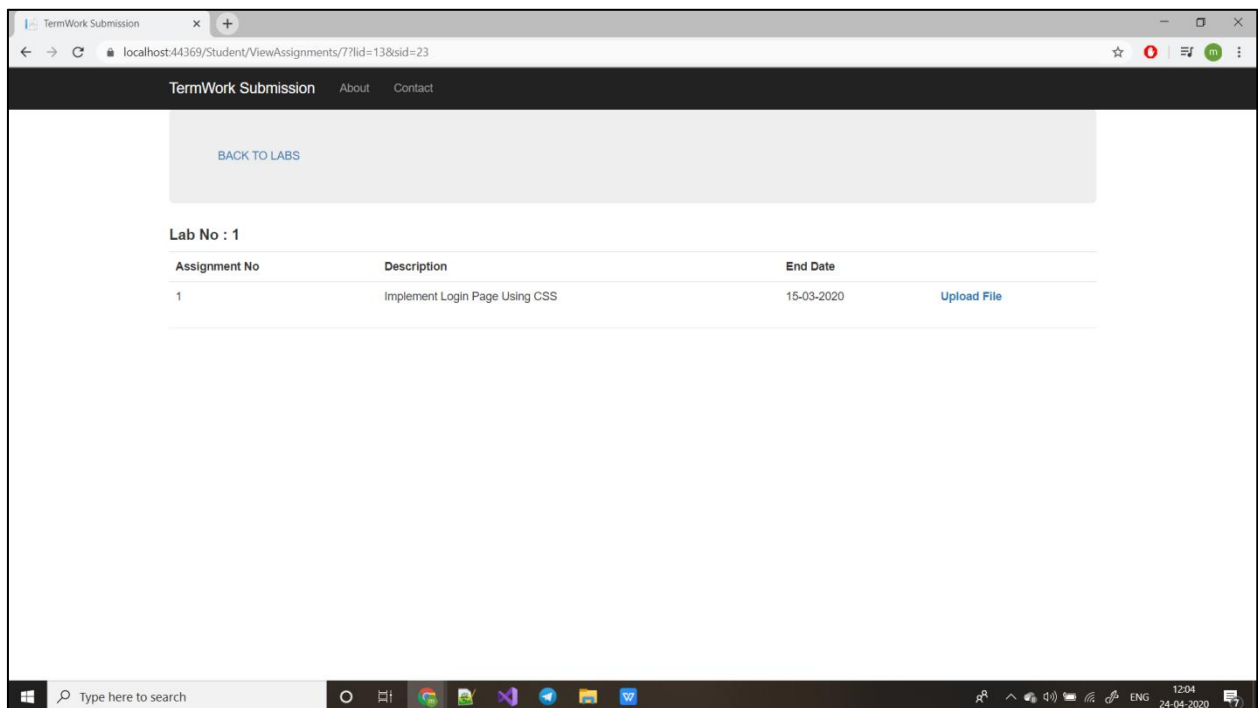
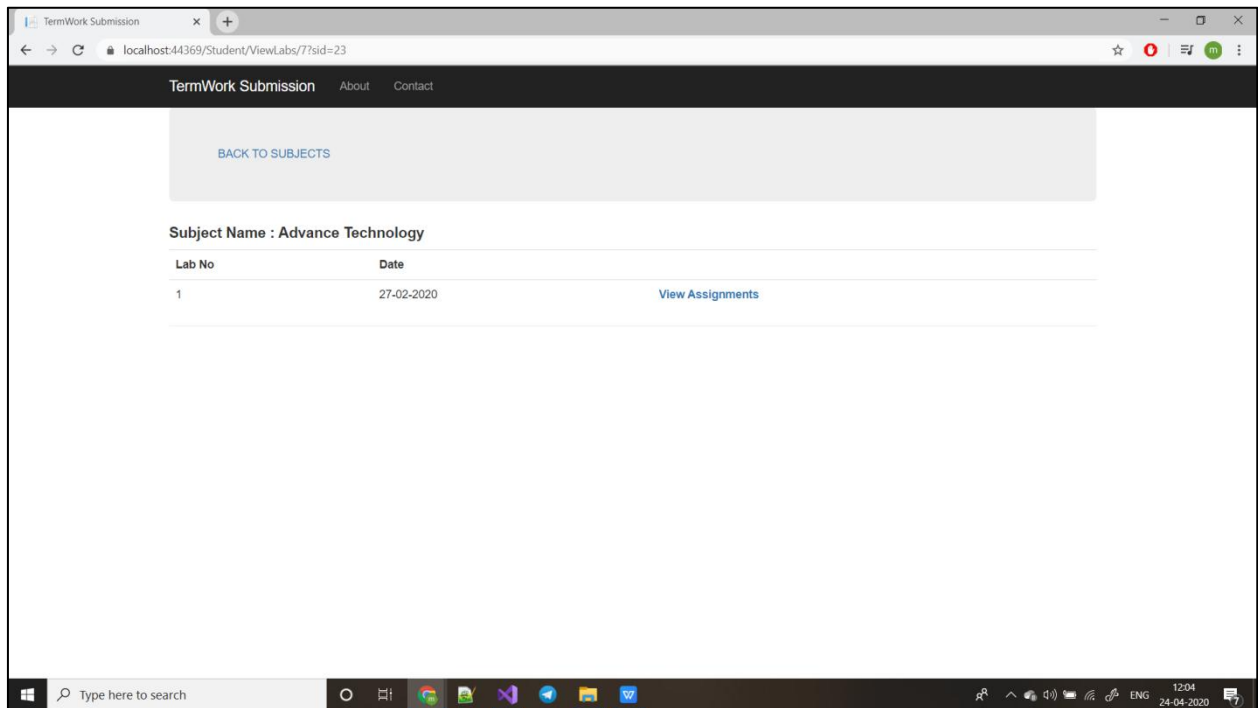
The screenshot shows a web browser window with the title 'TermWork Submission'. The address bar shows 'localhost:44369/Student/ViewSubject/7'. The page has a dark header with 'TermWork Submission', 'About', and 'Contact' links. The main content area has a 'BACK TO DETAILS' button at the top. Below it is a table with two columns: 'Subject Name' and 'Semester'. The table contains one row with 'Advance Technology' and '6'. To the right of the table is a 'View Labs' link. The Windows taskbar is visible at the bottom.

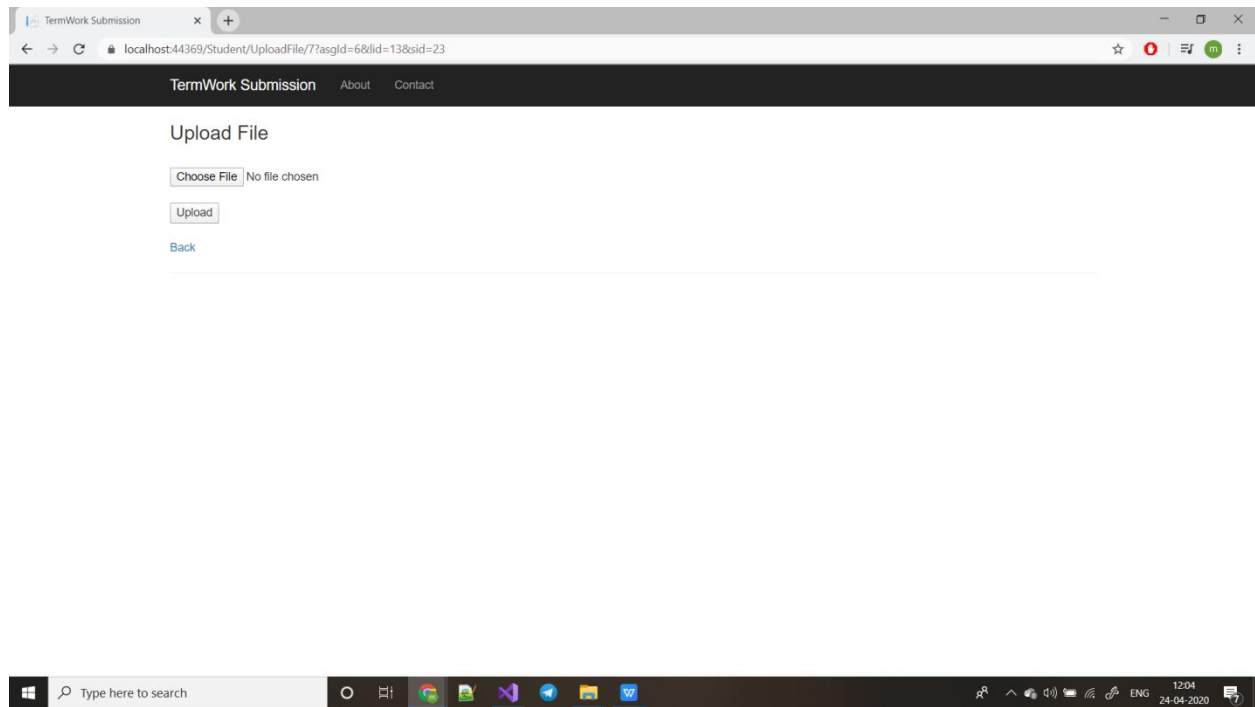
TermWork Submission About Contact

[BACK TO DETAILS](#)

Subject Name	Semester
Advance Technology	6

[View Labs](#)





## 8. Conclusion

---

The functionalities implemented in system after understanding all the system modules according to the requirements. Functionalities That are successfully implemented in the system are:

User login containing all the necessary validation on field

Logout

User Authentication

User Authorization

Upload File

Create, edit & delete student

Enrol students & delete enrolled students related to subjects

Create, edit & delete subject

Create, edit & delete lab

Create, edit & delete assignment

## 9. Limitations and Future Enhancements

---

The System has adequate scope for modification in future if it is necessary.

Development and launching of Mobile app and refining existing services and adding more service, System security, data security and reliability are the main feature.

In the existing system maximum 4MB(Because of by default setting from microsoft visual studio credibility purpose.) size of file can be uploaded, so it will be extended with the any size of file can be uploaded.

The API for the automated assessment can be added.

In the existing system there are only some selected branches & batches, so it will be extended to more branches & batches.

## 10. Reference / Bibliography

---

Following links and websites were referred during the development of this project.

<https://egov.ddit.ac.in>

<https://www.tutorialsteacher.com>

<https://www.c-sharpcorner.com>

<https://stackoverflow.com>

<https://docs.microsoft.com/en-us/dotnet/csharp>

<https://docs.microsoft.com/en-us/aspnet/mvc>