



C.T.E.S.'s
N. G. Acharya & D. K. Marathe College
Of Arts, Science & Commerce
NAAC ACCREDITED “A” GRADE

PROJECT REPORT ON
“ASSIGMENTS AND EXAM PORTAL”

SUBMITTED BY
Mr. DNYANESHWAR VITTHAL SHEKADE

UNDER THE GUIDANCE OF
Mrs. SHRADDHA D. DHAWAN

TO
UNIVERSITY OF MUMBAI
IN PARTIAL FULLFILMENT OF
T.Y.B.Sc.(COMPUTER SCIENCE)

SEM-VI
ACADEMIC YEAR 2020-2021



C.T.E.S.'s
N. G. Acharya & D. K. Marathe College
Of Arts, Science & Commerce
NAAC ACCREDITED “A” GRADE

CERTIFICATE

This is to certify that “**Mr. DNYANESHWAR VITTHAL SHEKADE**” Examination Seat No. _____ Student of N. G. ACHARYA & D. K. MARATHE COLLEGE has successfully completed the project work entitled “**ASSIGMENTS AND EXAM PORTAL**” during the academic year 2020-2021. In partial fulfilment of the requirement for the award of Degree in Bachelor of Computer Science under University Of Mumbai.

It is further certified that student completed all required phases of project.

PLACE: Mumbai

DATE: _____

PROJECT GUIDE
(Ms. Shraddha D. Dhawan)

HEAD OF DEPARTMENT
(Ms. Archana Jadhav)

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

ACKNOWLEDGMENT

The presented project, as a part of the curriculum was a first of its kind experience for me. I had looked upon this project not merely as an syllabus to be completed but as an aim to know, study, develop and experience the commercial software technologies.

I undersigned, have great pleasure in giving my sincere thanks to those who have contributed their valuable time in helping us to achieve the success in my project work. I am glad to say that, I have satisfactorily reached my aims and intentions, to make this project a success.

Foremost regards to my guide **Mrs. SHRADDHA D. DHAWAN** who was the guiding lights throughout the project work.

I would like to thank our CS staff for providing us sufficient information, which helped us to complete our project successfully.

I expressed sincere thanks to all faculty members of B. Sc.(CS.) department of Acharya College who have fine tuned my knowledge in the field of Information technology I also wish to mention the unsaid support of my parents who, as always helped me in every possible way to make this work of mine, a success.

Sincere thank from,
(Shekade Dnyaneshwar)

SR NO.	TOPIC	PAGE NO.
1)	Preliminary Investigation	6
1.1	Introduction	7
1.2	Overview of the technology	9
1.3	Proposed system and its Objectives & scope	12
1.4	Feasibility study	13
1.5	Project Synopsis	14
1.6	Gantt Chart	15
2)	SYSTEM ANALYSIS	17
2.1	Event Table	18
2.2	ERD	19
2.3	Class Diagram	21
2.4	Use Case Diagram	23
2.5	Activity Diagram	25
2.6	Sequence Diagram	27
2.7	Data Flow Diagram	27
2.8	SDLC Model	28
3)	System Design	29
3.1	Deployment Diagram	30

4)	System Layout and System Coding	31
4.1	System Layout	33
4.2	System Coding	51
5)	Implementation	123
5.1	Testing	
5.2	System Implementation	
6)	Advantages	125
7)	Future Scope	126
8)	Conclusion	127
9)	References	128

Phase 1:-

PRELIMINARY INVESTIGATION

1. INTRODUCTION

An online assignment handling is a system contained within the Module virtual learning environment. The functionality of the standard assignment handling module has been extended to cater for all the Department's needs in terms of receiving assignments from students, making them available to tutors to mark, returning grades, comments and marked work to students and keeping Registry and course administrators informed at all stages of the process. Extension requests are an integral part of the system.

Universities, Polytechnics and colleges of education are considered the main provider of knowledge in various fields. Various courses of studies are taught in institutions, covering several fields including applied Sciences, Math, Computer, Human Resource, and Accounting. Most courses at universities consist of theoretical as well as practical subject matter. To evaluate the level of understanding and degree of comprehension among students, assignments are often given.

Assignments are submitted by students either individually or in groups. Assignment management involves collecting, marking, and redistributing to students. Tregobov (1998) breaks the process down into four stages: submission, recording, marking, and return. Online assignment submission and management (OASM) involves the use of the World-Wide Web, the Internet and computers to aid this process (Jones, 2003). With traditional assignment submission system, lots of problems arise especially when the students have to

submit the answers of the assignment to the lecturer. There may be problems due to distance, time, or format of the assignment (written or printed). Also every learning process requires administrative support. Much of this administrative support is to some degree transparent but if the latter is not well organized it could disrupt the flow of learning between students and the staff.

As the educational world is moving faster and becoming more competitive, almost every university started to use an online submission system, or newer technologies to facilitate their task, to have more time, and to be in pace with this fast moving it world.

1.2 Overview of the technology

Front end :-HTML, CSS, JavaScript, Bootstrap.

1.HTML: Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

2. CSS: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and

3. JAVASCRIPT: JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

4. BOOTSTRAP: Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is the seventh-most-starred

project on GitHub, with more than 142,000 stars, behind freeCodeCamp (almost 312,000 stars) and marginally behind Vue.js framework.

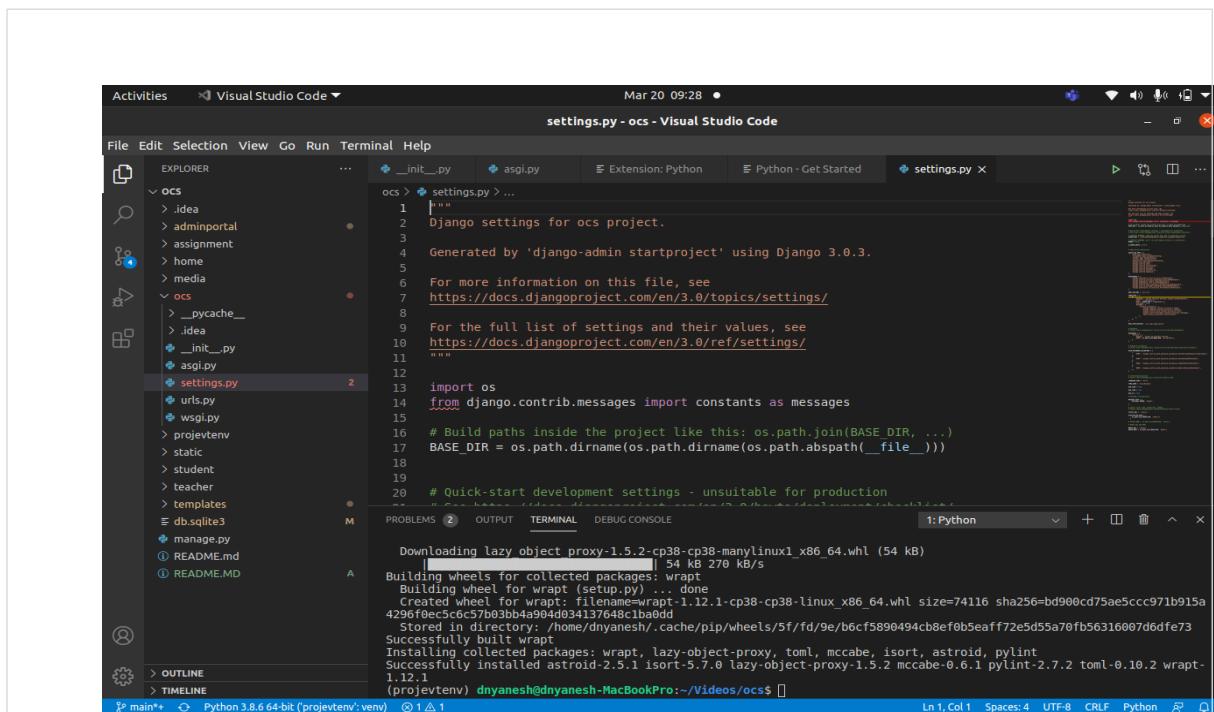
Back end :-Python, SQLite, Django Framework.

- Python: Python is user friendly and easy to learn application which is freely available or open source.
- Django Framework: Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Because Django was developed in a fast-paced newsroom environment, it was designed to make common Web-development tasks fast and easy. It's free and open source
- SQLite: The SQLite file format is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way throughout the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems and as a long-term archival format for data. There are over 1 trillion SQLite databases in active user.

IDE (Integrated Development Environment).

VS Code or Visual Studio Code: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Fig 1.1 : VS Code



1.3 Objectives and Scope

The objectives of the system are-

- To reduce manual paperwork.
- Reduced sharing and distribution time.
- Increased reliability.
- Increased operational efficiency.
- Data security.

This Online Assignment System can be readily used by non-programming personal avoiding human handled chance of error. Students can create their accounts and start solving Assignment shared by Teachers. Administrator is must be an authorized user who will keep track of all the uploaded Assignment and manage users as well through the admin panel.

Scope

This project has a large scope as it has the following features which help in making it easy to use, understand and modify it:

- Making Distance Learning Interactive.
 - No Need to do Paper Work.
 - To save the environment by using paper free work.
 - To increase the efficiency of the notes.
 - Management of all notes.
 - Easy and secure user log in and registration.
 - Department based Assignment management.
-

1.4 Feasibility study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility.
- Operational Feasibility.

1.5 Project Synopsis

1.5.1 Title of Project

Title : Assignments and exam portal

1.5.2 Category of the Project

Category: Website

1.5.3 Tools/Platform Requirements (Runtime Environment Specification)

Hardware

Laptop

Processor	:	i5 2GHz speed
Ram	:	4GB
HDD	:	128 GB SSD

Software

Operating System	:	Linux, Windows, MacOS X
Front End	:	HTML,CSS
Coding language	:	python,php
Backend	:	djyango

1.6 Gantt Chart

MONTH PHASES	DECEMBER	JANUARY	FEBRUARY	MARCH
	1	2	3	4
REQUIREMENT	1	2	3	4
PALNNING & ANALYSIS	1	2	3	4
DESIGN	1	2	3	4
CODING	1	2	3	4
TESTING	1	2	3	4

Phase 2:-

SYSTEM ANALYSIS

Software Requirement Specification (SRS)

Purpose:

The main objective of this applications to make distance learning interactive using Online Assignment System. We need maintain hundreds of thousands of records. Also searching should be very faster so we can find required details instantly.

Scope:

This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

DEVELOPERS RESPONSIBILITIES OVERVIEW

The developer is responsible for:

Developing the system, which meets the SRS and solving all the requirements of the system?

Demonstrating the system and installing the system at client's location after the acceptance testing is successful.

Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.

2.1 Event Table

2.2 ERD

In an effective system, data is divided into discrete entities. An entity relationship model is an illustration of various entities in a business and the relationship between them. An ER model is derived from the business specifications or narratives and builds during the phase of system development life cycle. ER models separate the information required for a business from the activities performed.

Benefits of ER modeling

Documents information for the organization in a clear and precise format. Provides a clear picture of the scope of information requirements. Provides an easily understood pictorial map for the database design

KEY COMPONENTS

ENTITY:

A thing of significance about which information needs to be known. For example customer, cart etc.

ATTRIBUTE:

Something that describes an entity

RELATIONSHIP:

A named association between entries showing optionality and degree.

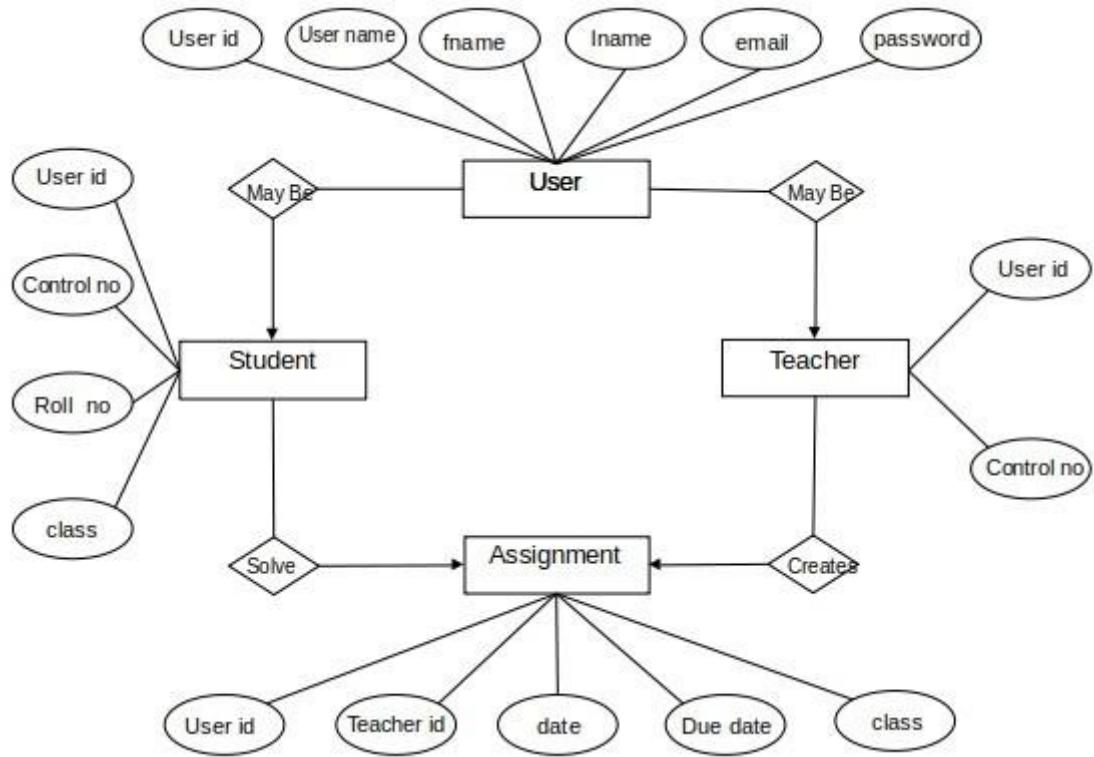


Fig 2.1 : ER Diagram

2.3 Class Diagram

Class diagrams express, in a general way, the static structure of a system, in Class terms of classes and relationships between those classes. Just as a class describes a set of objects, an association describes a set of links; objects are class instances and links are association instances.

It consists of various components which are as follows:

- Class
- Attribute
- Operation
- Association

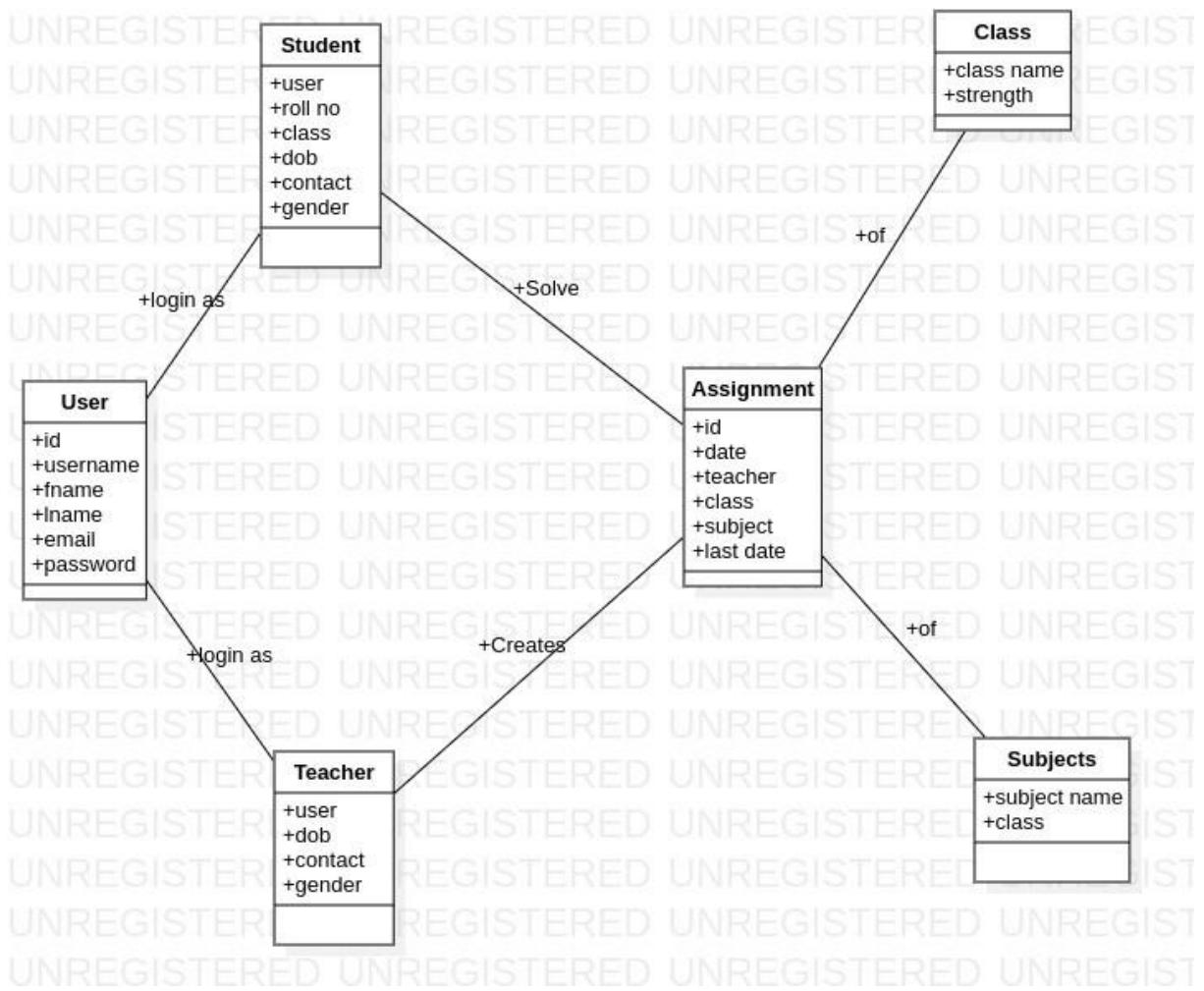


Fig 2.1 : Class Diagram

2.4 Use Case Diagram

Use case diagrams represent use cases, actors and relationships between use cases and actors. A use case diagram is a type of behavioral diagram defined by and created from a test case analysis. It represents a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. It also shows what system functions are performed for which actor. It represents typical sets of scenarios that help to structure, relate and understand essential requirements. It captures the dynamic aspect of a system.

It consists of various components which are as follows:

- Actor
- Use case
- Association
- Relationship
- Boundary

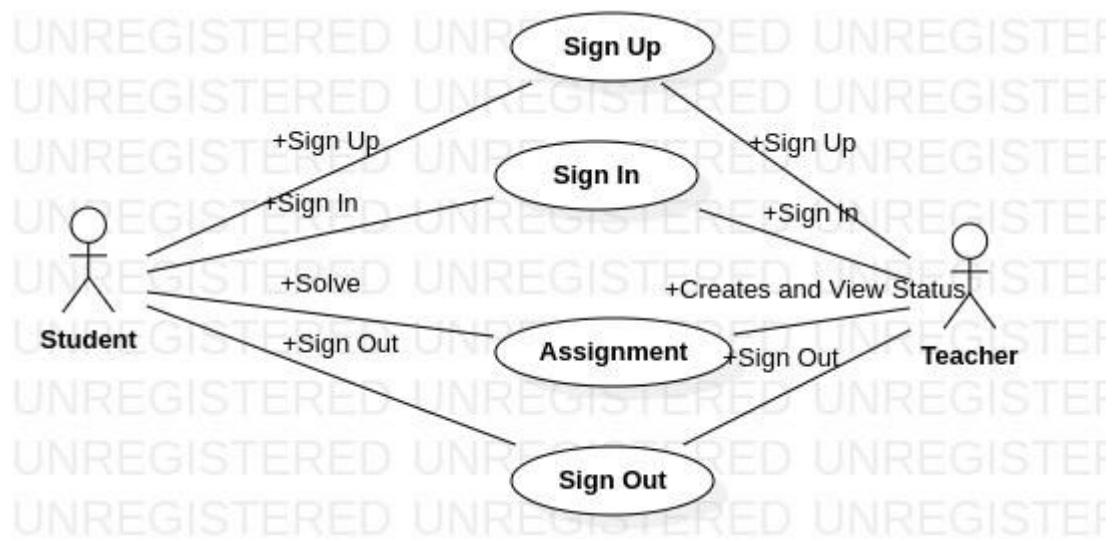


Fig 2.3 : Use Case Diagram

2.5 Activity Diagram

An activity diagram is a variant of state chart diagrams organized according to actions, and mainly targeted towards representing the internal behavior of a method or a use case. Activity Diagram is a graphical representation of workflows of step wise activities and actions with support of choice, iteration and concurrency. It describes the dynamic aspects of the

system. It shows the overall flow of control. It can be described as an

operation of the system. It specifies the flow which can be sequential

branched or concurrent. It consists of various components which are

as follows

1. Activity
2. Action
3. Fork
4. Join
5. Edge
6. Node

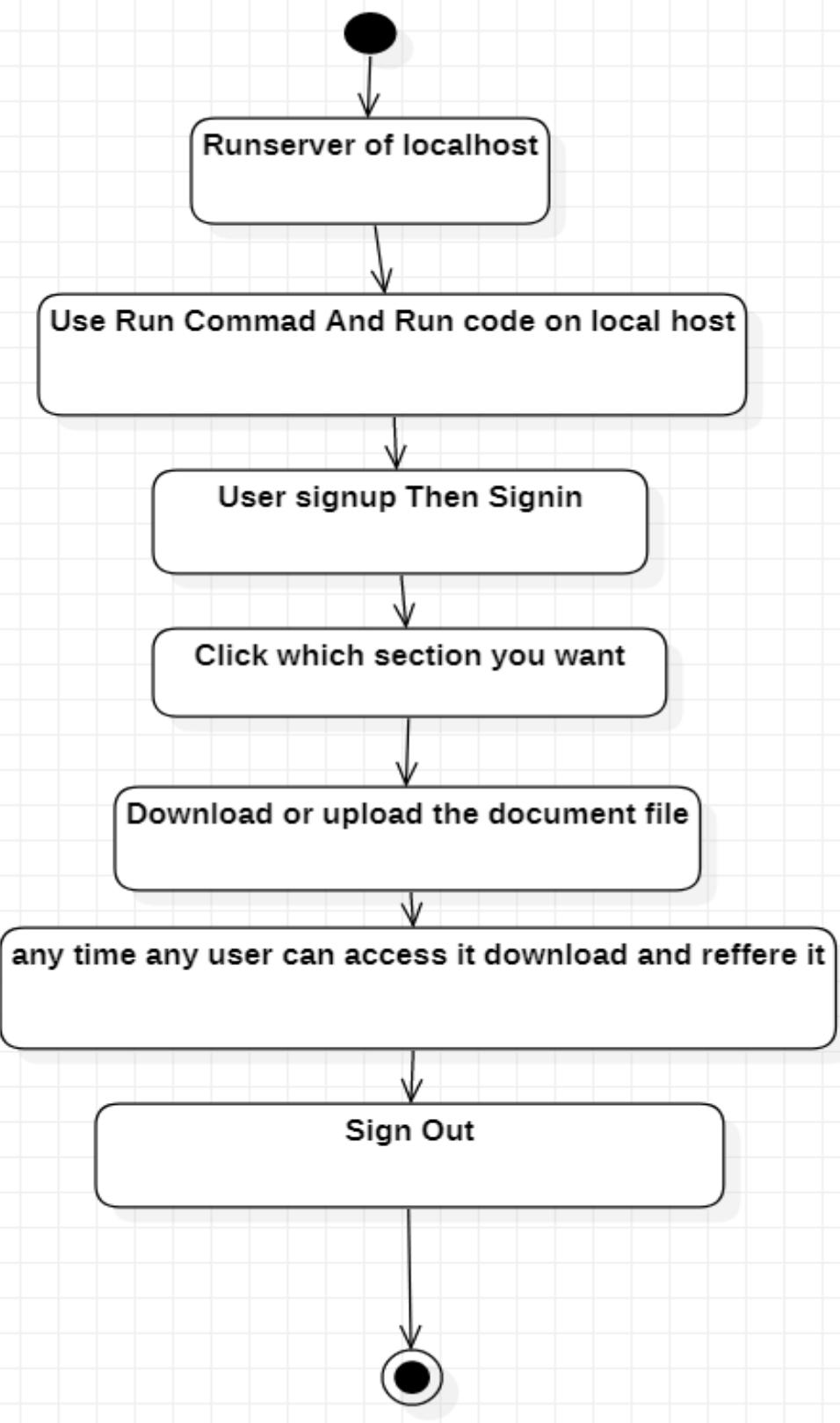


Fig 2.4 : Activity Diagram

2.6 Sequence Diagram

Sequence diagrams display interactions between objects from a temporal standpoint. Unlike collaboration diagrams, the context of the objects is not represented explicitly. The representation focuses on expressing interactions.

2.7 Data Flow Diagrams:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system.

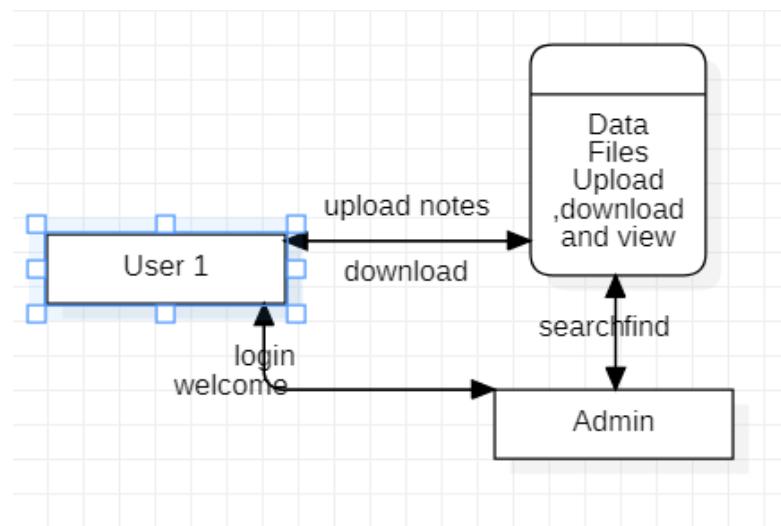


Fig 2.5 : Data Flow Diagram

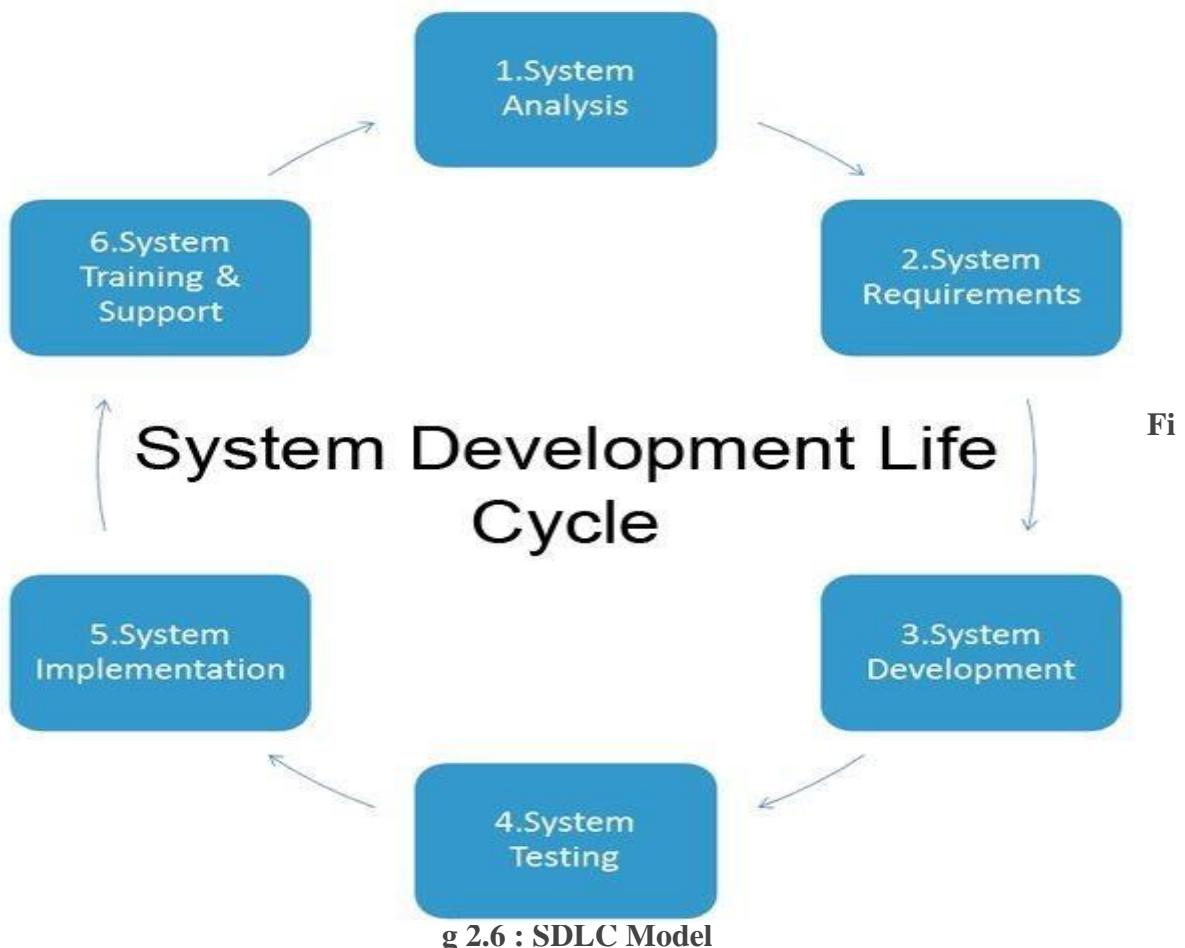
These are the central tool and the basis from which the other components are developed. The transformation of data from input to output through processed, may be described logically and independently of physical components associated with the system.

These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between

peopledepartments and workstations. A full description of a system actually consists of a set of data flow diagrams.

The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called a “*context diagram*”.

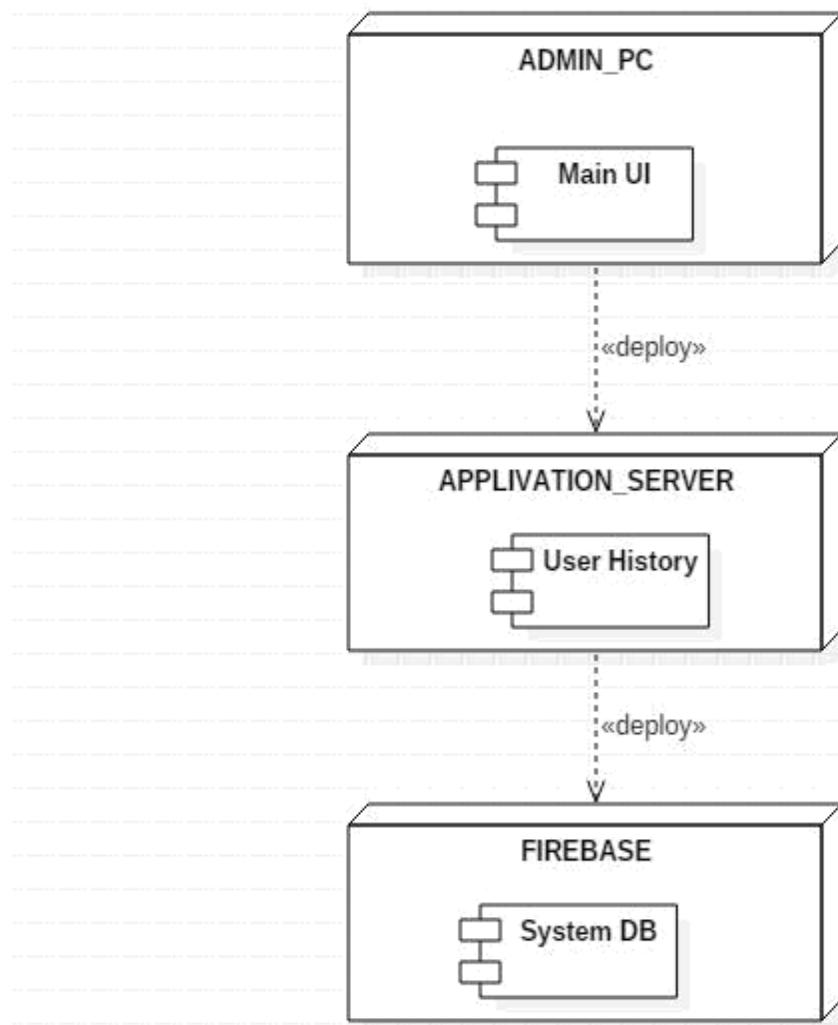
2.8 SDLC Model:



Phase 3:-

SYSTEM DESIGN

3.1 DEPLOYMENT DIAGRAM:



Phase 4:

SYSTEM CODING

PROGRAM DESCRIPTION

Implementation Approaches:

Planning is an essential aspect of any successful project. Implementation planning can reduce the chance of failure by helping turn strategy into action.

A *project implementation plan* is the plan that we create to successfully move our project plan into action for that.....

Our group consist of two members, and we need to know exactly what we're going to do (the standards that we are going to follow) to meet the project's objectives. The implementation planning process is the very first thing we do to manage our scope. Project scope planning is concerned with the definition of all the work needed to successfully meet the project objectives. The whole idea here is that when we start the implementation of the project, we need to have a clear picture of all the work that needs to happen on our project, and as the project progresses, we need to keep that scope up to date.

Our major focus is on risk assessment and on minimizing project risks by breaking a project into smaller segments and providing more ease of change during development process, as well as providing the opportunity to evaluate risks and weight consideration of project continuation throughout the life cycle.

SCREEN LAYOUT

First Page:

The screenshot shows a Google Chrome browser window with the following details:

- Address Bar:** Mar 17 08:13 | 127.0.0.1:8000
- Open Tabs:** WhatsApp, Home, Explain Sub Functions of..., Settings - Passwords
- Bookmarks Bar:** Apps, Stack Overflow, Google, tutorialpoint, WhatsApp, Trending Torr..., W3Schools On..., Other bookmarks
- Page Content:**
 - Header:** Assignments And Exam Portal, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT, Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College
 - Image:** A group photo of approximately 15 people, mostly men in formal attire (suits and ties), standing outdoors in front of a building.
 - Text:** Developed By DNYANESHWAR PVT LTD

SIGN-UP PAGE:

STUDENT

Activities Google Chrome Mar 17 08:14

(25) WhatsApp Register as Student Explain Sub Functions of Settings – Passwords

127.0.0.1:8000/stud_register

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College Contact us

 Assignments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT 

Home Log in Sign up

Register As Student

First Name <input type="text" value="First Name"/>	Last Name <input type="text" value="Last Name"/>
Control Number <input type="text" value="Control Number"/>	Contact Number <input type="text" value="Contact Number"/>
Email <input type="text" value="dryanesh"/>	Contact Number <input type="text" value="Contact Number"/>
Date of Birth <input type="text" value="dd/mm/yyyy"/>	Gender: <input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Class <input type="text" value="select"/>	Roll Number <input type="text" value="Roll Number"/>
Password <input type="text" value="*****"/>	Confirm Password <input type="text" value=""/>

Submit **Clear**

Developed By DNYANE SHWAR PVT LTD 

127.0.0.1:8000

TEACHER

Activities Google Chrome Mar 17 08:14

(25) WhatsApp Register as Student Explain Sub Functions of Settings – Passwords

127.0.0.1:8000/teacher_register

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College Contact us

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home Log in Sign up

Register As Teacher

First Name	Last Name
First Name	Last Name
Control Number	
Control Number	
Email	Contact Number
dryanesh	Contact Number
Date of Birth	Gender:
dd/mm/yyyy	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Password	Confirm Password

Submit Clear

Developed By DNYANESHWAR PVT LTD



127.0.0.1:8000

LOG-IN PAGE:

S

T

U

D

E

N

T

127.0.0.1:8000

The screenshot shows a Google Chrome browser window with the address bar set to 127.0.0.1:8000/stud_login. The main content area displays the 'Assignments And Exam Portal' for the 'COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT'. It features a logo of the college, a red circular logo with 'CET', and navigation links for Home, Log in, and Sign up. A login form is present with fields for username ('dnyanesh') and password ('.....'), and a 'Log in' button. Below the form, it says 'Developed By DNYANESHWAR PVT LTD' and includes a small logo.

TE
AC
HE
R

127.0.0.1:8000

The screenshot shows a Google Chrome browser window with the address bar set to 127.0.0.1:8000/teacher_login. The main content area displays the 'Assignments And Exam Portal' for the 'COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT'. It features a logo of the college, a red circular logo with 'CET', and navigation links for Home, Log in, and Sign up. A login form is present with fields for username ('manojkuamj') and password ('.....'), and a 'Log in' button. Below the form, it says 'Developed By DNYANESHWAR PVT LTD' and includes a small logo.

STUDENT HOME PAGE:

Activities Google Chrome ▾ Mar 17 08:21

(26) WhatsApp DnyaneshAdmin PDF to Word Converter - 1 Settings – Passwords

127.0.0.1:8000/student/ Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College Contact us f t G i

Assignments And Exam Portal

 COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT 

Home DnyaneshAdmin Notifications

Welcome DnyaneshAdmin

DnyaneshAdmin

Class TYCS

Roll No. 17

Assignments

#	Upload	Subject	Teacher	Due Date	Type
1	7 hours ago	IOT	manoj kumar	March 21, 2021	MCQ

Developed By DNYANESHWAR PVT LTD 

127.0.0.1:8000

TEACHER HOME PAGE:

Activities Google Chrome Mar 17 08:20

(26) WhatsApp manoj kumar PDF to Word Converter - 1 Settings – Passwords

127.0.0.1:8000/teacher/ Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College Contact us f t G i

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

manoj kumar Home Notifications

Assignments

#	Upload	Class	Subject	Due Date	Type	Operations
1	6 hours ago	TYCS	IOT	21/03/2021	MCQ	Save Edit Status Delete

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

ADD ASSIGNMENT PAGE:

Activities Google Chrome Mar 17 08:23

(26) WhatsApp Add Assignment PDF to Word Converter - 1 Settings - Passwords

127.0.0.1:8000/teacher/add_assignment Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College Contact us f t G i

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home manoj kumar Notifications

Add Assignment

Class: TYCS Subject: IOT

Last Date: dd/mm/yyyy Assignment Type: --select--

Submit

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

ADDING QUESTIONS:

MCQ

A screenshot of a web browser window titled "Assignment". The URL is 127.0.0.1:8000/teacher/assignment/TYCS-IOT-MCQ2021-03-17%2008:24:59.912844. The page content is an assignment portal for the Computer Science and Information Technology Department. It features a logo of the college, a navigation bar with links like Home, manoj kumar, Notifications, and Contact us, and social media icons. Below the header is a section for "MCQs" with a table and a TINY editor for creating questions. The TINY editor has a toolbar with various formatting options and a large text area below it.

Activities Google Chrome Mar 17 08:32

(27) WhatsApp Assignment PDF to Word Converter - IOT Multiple Choice Ques... +

127.0.0.1:8000/teacher/assignment/TYCS-IOT-MCQ2021-03-17%2008:30:31.421372

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Home manoj kumar Notifications

Topic : IOT
Class : TYCS
Date : March 17, 2021, 8:30 a.m.

MCQs

#	Questions	Answer	Operation
---	-----------	--------	-----------

Question

My Favorites File Edit View Insert Format Tools Table Help

MQTT stands for _____

P 3 WORDS POWERED BY TINY

A MQ Telemetry things B MQ trasport telemetry
C MQ Transport things D mq telemetry Transport

Ans mq telemetry Transport

Submit

Developed By DNYANESHWAR PVT LTD

BRIEF QUESTIONS

Activities Google Chrome Mar 17 08:44

(27) WhatsApp 127.0.0.1:8000/student/assig... IOT Multiple Choice Ques... +

127.0.0.1:8000/student/assignment/TYCS-IOT-File%20Upad2021-03-17%2008:41:42.586224

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marath College Contact us f t G S

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home Dnyaneshwar Shekade Notifications

Teacher : manoj kumar
Subject : IOT
Last Date : March 29, 2021

Upload Files

1 what is a iot explain in bref.
Choose file Browse

2 usages of iot
Choose file Browse

3 explain raspberry pi
Choose file Browse

4 what is esp and its components

MCQ TYPE ASSIGNMENT QUESTIONS

Activities Google Chrome Mar 17 08:38

(27) WhatsApp 127.0.0.1:8000/student/assignment/TYCS-IOT-MCQ2021-03-17%2008:30:31.421372 IOT Multiple Choice Ques +

← → C ⌛ 127.0.0.1:8000/student/assignment/TYCS-IOT-MCQ2021-03-17%2008:30:31.421372 Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College Contact us f t G i

Assignments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home DnyaneshAdmin Notifications

Teacher : manoj kumar
Subject : IOT
Last Date : March 18, 2021

Multiple Choice Questions

1. MQTT stands for _____

MQ Telemetry things MQ trasport telemetry
 MQ Transport things mq telemetry Transport

Submit

Developed By DNYANESHWAR PVT LTD

GitHub icon

127.0.0.1:8000

MCQ RESULT

Activities Google Chrome Mar 17 08:39

(27) WhatsApp Result for MCQs IOT Multiple Choice Quest +
127.0.0.1:8000/student/mcq_result/TYCS-IOT-MCQ2021-03-17%2008:30:31.421372

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College Contact us f t G i

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home DnyaneshAdmin Notifications

Result

Total Questions : 1
Attemp : 1
Correct : 1

#	Question	Your Answer	Correct Answer
1	MQTT stands for _	mq telemetry Transport	mq telemetry Transport

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

Activities Google Chrome Mar 17 12:15

Result for MCQs (26) WhatsApp Settings – Passwords

127.0.0.1:8000/student/mcq_result/TYCS-IOT-MCQ2021-03-17%2008:30:31.421372

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College Contact us f t G @

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home DnyaneshAdmin Notifications

Result

Total Questions : 1 0%

Attemp : 1

Correct : 0

#	Question	Your Answer	Correct Answer
1	MQTT stands for _	MQ Telemetry things	mq telemetry Transport

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

BRIEF QUESTIONS SUBMIT PAGE

Activities Google Chrome Mar 17 08:44

(27) WhatsApp 127.0.0.1:8000/student/assignment/TYCS-IOT-File%20Uplad2021-03-17%2008:41:42.586224 IOT Multiple Choice Quest +

← → C Home 127.0.0.1:8000/student/assignment/TYCS-IOT-File%20Uplad2021-03-17%2008:41:42.586224

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Contact us Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K.Marathe College

Assignments And Exam Portal
COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home Dnyaneshwar Shekade Notifications

Teacher : manoj kumar
Subject : IOT
Last Date : March 29, 2021

Upload Files

1 what is a iot explain in bref.
Choose file Browse

2 usages of iot
Choose file Browse

3 explain raspberry pi
Choose file Browse

127.0.0.1:8000

Activities Google Chrome Mar 17 08:45

Cancel Open File

Recent Home Desktop Documents Downloads Music Pictures Videos + Other Locations

dnyanesh Downloads

Name	Size	Type	Modified
IMG_20200225_090751.jpg	3.2 MB	Image	14 Jan
IMG_31082019_125721.jpg	1.1 MB	Image	19 Jan
iTunes64Setup.exe	200.5 MB	Program	4 Feb
ocs.zip	1.4 MB	Archive	00:04
PixelExperience_Plus_dreamlte-10.0-20200421-0839-UNOFFICIAL.zip	1.2 GB	Archive	12 Feb
PixelExperience_Plus_X00TD-10.0-20201116-2121-OFFICIAL.zip	1.1 GB	Archive	20 Jan
Prac 01 - Reconnaissance.docx	46.0 kB	Document	21 Jan
Prac 06 - ping, ifconfig, netstat commands.docx	360.5 kB	Document	13 Feb
pract 2 writeups.txt	1.2 kB	Text	13 Feb
Project.zip	53.1 MB	Archive	Mon
projectblackbook-tyit-170121122010.pdf	9.6 MB	Document	Yesterday
Project_Soft_Copy.docx	10.2 kB	Document	9 Feb
pycharm-professional-2020.3.2.tar.gz	607.6 MB	Archive	11 Jan
Roll_no_51_Blackbook_CNC_Machine.docx	13.7 MB	Document	9 Feb
Roll_no_51_Sem_5_Project_Blackbook_CNC_Machine.pdf	17.5 MB	Document	9 Feb
schoolmanagement.zip	2.9 MB	Archive	Mon
SetupCrypTool_1.4.41_en.exe	80.5 MB	Program	9 Feb
teams_1.3.00.30857_amd64.deb	76.3 MB	Archive	11 Jan
TeamSkeetXEvaElifie.21.02.14.Eva.Elifie.Cum.Hearts.XXX.2160p.MP4-WRB-[rarbg.to].torrent	14.4 kB	BitTorrent seed file	14 Feb
teamviewer_15.15.2_amd64.deb	14.7 MB	Archive	14 Feb
twrp-3.2.1-0-ysl.Img	32.3 MB	Raw disk image	11 Jan
TYCS login id details updated.docx	17.1 kB	Document	21 Jan
ui.zip	1.1 MB	Archive	12 Feb
UNIT-I.pptx	1.2 MB	Presentation	13 Jan
Untitled_2021_02_04_09_56_12.mp4	16.3 MB	Video	5 Feb
VID_20160915_210811.mp4	23.9 MB	Video	21 Feb
WhatsApp Audio 2021-02-04 at 11.36.01 PM.mpeg	493.0 kB	Video	5 Feb
zoom-1_amd64.deb	46.3 MB	Archive	23 Jan

CHECK STATUS OF ASSIGNMENT PAGE

SOLVED

The screenshot shows a Google Chrome browser window with the following details:

- Activity Bar:** Activities → Google Chrome Mar 17 08:47
- Open Tabs:** WhatsApp, TYCS-March 17, 2021, 8:41, IOT Multiple Choice Ques
- Address Bar:** 127.0.0.1:8000/teacher/assignment_status/TYCS-IOT-File%20Uplpad2021-03-17%2008:41:42.586224
- Bookmarks Bar:** Apps, WhatsApp, Stack Overflow, Google, tutorialpoint, W3Schools On..., Other bookmarks
- Page Content:**
 - Logo:** Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College
 - Title:** Assigments And Exam Portal
COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT
 - Progress Bar:** Total Students : 2
Solved Students : 1
50%
 - Buttons:** Total, Solved, Unsolved
 - Table:** Student Name, Control No, Submit On
 - Row:** 1 Dnyaneshwar Shekade, 1806094, a minute ago, View, Close
 - Footer:** Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

UNSOLVED

Activities Google Chrome Mar 17 08:48

(27) WhatsApp -TYCS-March 17, 2021, 8:41 IOT Multiple Choice Ques... | +

127.0.0.1:8000/teacher/assignment_status/TYCS-IOT-File%20Uplpad2021-03-17%2008:41:42.586224

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... | Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College Contact us f G i

Assignments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home manoj kumar Notifications

Subject:IOT Total Students : 2
Class:TYCS Solved Students : 1
Date:March 17, 2021, 8:41 a.m.
Type:File Upplad 50%

Total Solved Unsolved

#	Student Name	Roll No.	Class	Control Number
1	DnyaneshAdmin	17	TYCS	dnyanesh

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

TOTAL

Activities Google Chrome Mar 17 08:48

(27) WhatsApp -TYCS-March 17, 2021, 8:41 IOT Multiple Choice Ques... | +

127.0.0.1:8000/teacher/assignment_status/TYCS-IOT-File%20Uplpad2021-03-17%2008:41:42.586224

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... | Other bookmarks

Chembur Trombay Education Society's N.G. Acharya And D.K. Marathe College Contact us f G i

Assignments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home manoj kumar Notifications

Subject:IOT Total Students : 2
Class:TYCS Solved Students : 1
Date:March 17, 2021, 8:41 a.m.
Type:File Upplad 50%

Total Solved Unsolved

#	Student Name	Roll No.	Class	Control Number
1	DnyaneshAdmin	17	TYCS	dnyanesh
2	Dnyaneshwar Shekade	51	TYCS	1806094

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

ADMIN LOGIN PAGE

The screenshot shows a Google Chrome window with three tabs: WhatsApp, Admin Login, and IOT Multiple Choice Ques. The Admin Login tab is active. A modal window titled "Admin Login" is open, prompting for "Control No." with the value "dryanesh" and "Password" with the value "*****". Below the password field are three icons: "Screen", "Login", and "Selection". There are also "Show Pointer" and "Delay in Seconds" settings.

ADMIN PORTAL

The screenshot shows a Google Chrome window with three tabs: WhatsApp, Admin Portal, and IOT Multiple Choice Ques. The Admin Portal tab is active. The header includes "View Site" and "Logout" links. Below the header, there are navigation links for "Students", "Teachers", "Classes", and "Subjects". The main content area displays a welcome message "Welcome to Admin Portal".

Students

#	First Name	Last Name	Roll No.	Class	Control No.	Date Of Birth	Contact No.	Email
1	Dryanesh	Admin	17	TYCS	dryanesh	21/04/2001	9324141259	
2	Dryaneshwar	Shekade	51	TYCS	1806094	12/01/2000	7039997894	dryaneshwar@gmail.com

Teachers

#	First Name	Last Name	Control No.	Date Of Birth	Contact No.	Email	Gender
1	manoj	kumar	123	12/01/1985	9405488894	manojkumar@gmail.com	Male

Activities Google Chrome Mar 17 08:57

(27) WhatsApp Admin Portal IOT Multiple Choice Ques + 127.0.0.1:8000/adminportal/ Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Classes

#	Class	Capacity	Operation
1	FYCS	60	<button>Save</button> <button>Delete</button>
2	FYIT	60	<button>Save</button> <button>Delete</button>
3	SYCS	60	<button>Save</button> <button>Delete</button>
4	SYIT	60	<button>Save</button> <button>Delete</button>
5	TYCS	40	<button>Save</button> <button>Delete</button>
6	TYIT	40	<button>Save</button> <button>Delete</button>

[+](#)

Subjects

#	Subject	Class	Operation
#	IOT	TYCS	<button>Save</button> <button>Delete</button>

[+](#)

Activities Google Chrome Mar 17 08:57

(27) WhatsApp Admin Portal IOT Multiple Choice Ques + 127.0.0.1:8000/adminportal/ Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Teachers

#	First Name	Last Name	Control No.	Date Of Birth	Contact No.	Email	Gender	Operation
1	manoj	kumar	123	12/01/1985	9405488894	manojkumar@gmail.com	Male	<button>Save</button>

[+](#)

Classes

#	Class	Capacity	Operation
1	FYCS	60	<button>Save</button> <button>Delete</button>
2	FYIT	60	<button>Save</button> <button>Delete</button>
3	SYCS	60	<button>Save</button> <button>Delete</button>
4	SYIT	60	<button>Save</button> <button>Delete</button>
5	TYCS	40	<button>Save</button> <button>Delete</button>
6	TYIT	40	<button>Save</button> <button>Delete</button>

IF USER CREATED BY ADMIN

TO UPDATE STUDENTS DETAILS

Activities Google Chrome ▼ Mar 17 09:11 ●

(27) WhatsApp Admin Portal Change user | Django site Update your Data +

127.0.0.1:8000/student/update_data

Apps Stack Overflow Google tutorialpoint WhatsApp Trending Torr... W3Schools On... Other bookmarks

Assigments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home sakshi chavan Notifications

Welcome sakshi chavan

Update Your Information

First Name	Last Name	
sakshi	chavan	
Control Number		
1234		
Class	Roll Number	
TYCS	15	
Email	Contact Number	
1234	Contact Number	
Date of Birth	Gender:	
dd/mm/yyyy	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
Current Password	New Password	Confirm New Password

Submit Clear

127.0.0.1:8000 Developed By DNYANESHWAR PVT LTD

TO UPDATE TEACHER DETAILS

Activities Google Chrome Mar 17 09:15

(27) WhatsApp Admin Portal Change user | Django site Update your Data

127.0.0.1:8000/teacher/update_data

Assignments And Exam Portal COMPUTER SCIENCE AND INFORMATION TECHNOLOGY DEPARTMENT

Home rajiv bhatt Notifications

Update Your Information

First Name	Last Name	
rajiv	bhatt	
Control Number		
12345		
Email	Contact Number	
12345	Contact Number	
Date of Birth	Gender:	
dd/mm/yyyy	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other	
Current Password	New Password	Confirm New Password

Submit Clear

Developed By DNYANESHWAR PVT LTD

127.0.0.1:8000

4.2)CODING:

Project Directory Tree

1.Root Directory

```
dnyanesh@dnyanesh-MacBookPro: ~/Documents/ocs
|   |-- tests.py
|   |-- urls.py
|   '-- views.py
`-- templates
    '-- base.html

31 directories, 44 files
dnyanesh@dnyanesh-MacBookPro:~/Documents/ocs$ tree -L 1
.
|-- adminportal
|-- assignment
|-- db.sqlite3
|-- home
|-- manage.py
|-- media
|-- ocs
|-- projevtenv
|-- static
|-- student
|-- teacher
`-- templates

10 directories, 2 files
dnyanesh@dnyanesh-MacBookPro:~/Documents/ocs$
```

2.Apps Directories

```
dnyanesh@dnyanesh-MacBookPro:~/Documents/ocs$ cd Documents
dnyanesh@dnyanesh-MacBookPro:~/Documents$ cd ocs/
bash: cd: ocs/: No such file or directory
dnyanesh@dnyanesh-MacBookPro:~/Documents$ cd ocs/
dnyanesh@dnyanesh-MacBookPro:~/Documents/ocs$ tree -L 2
.
|-- adminportal
|   |-- __init__.py
|   |-- __pycache__
|   |-- admin.py
|   |-- apps.py
|   |-- migrations
|   |-- models.py
|   |-- templates
|   |-- tests.py
|   |-- urls.py
|   '-- views.py
|-- assignment
|   |-- __init__.py
|   |-- __pycache__
|   |-- admin.py
|   |-- apps.py
|   |-- migrations
|   '-- models.py
```

```
.
|   '-- tests.py
|   '-- views.py
|-- db.sqlite3
-- home
|   |-- __init__.py
|   |-- __pycache__
|   |-- admin.py
|   |-- apps.py
|   |-- context_processors.py
|   |-- migrations
|   |-- models.py
|   |-- templates
|   |-- templatetags
|   |-- tests.py
|   '-- urls.py
|   '-- views.py
-- manage.py
-- media
    '-- assignments
-- ocs
    |-- __init__.py
    |-- __pycache__
    |-- asgi.py
    '-- settings.py
```

```
dnyanesh@dnyanesh-MacBookPro: ~/Documents/ocs
```

```
|- |-- urls.py
|- |-- wsgi.py
|-- projenv
|   |-- bin
|   |-- lib
|   `-- pyvenv.cfg
|-- static
|   '-- Images
|-- student
|   |-- __init__.py
|   |-- __pycache__
|   |-- admin.py
|   |-- apps.py
|   |-- migrations
|   |-- models.py
|   |-- templates
|   |-- tests
|   |-- urls.py
|   '-- views.py
|-- teacher
|   |-- __init__.py
|   |-- __pycache__
|   |-- admin.py
|   |-- apps.py
|   |-- migrations
|   |-- models.py
|   |-- templates
|   |-- templatetags
|   |-- tests.py
|   |-- urls.py
|   '-- views.py
`-- templates
    '-- base.html

31 directories, 44 files
```

```
dnyanesh@dnyanesh-MacBookPro: ~/Documents/ocs$ tree -L 1
```


CODING

PYTHON FILES

ocs/settings.py

"""

Django settings for ocs project.

Generated by 'django-admin startproject' using Django 3.0.3.

*For more information on this file, see
<https://docs.djangoproject.com/en/3.0/topics/settings/>*

*For the full list of settings and their values, see
<https://docs.djangoproject.com/en/3.0/ref/settings/>*

"""

```
import os
from django.contrib.messages import constants as messages

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '@=h5)qwi59%&%&7ba3%3^x%h@(22lw!vycdy62t!qf52ny!$f5'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]

# Application definition

INSTALLED_APPS = [
    'home.apps.HomeConfig',
    'assignment.apps.AssignmentConfig',
    'student.apps.StudentConfig',
    'teacher.apps.TeacherConfig',
    'adminportal.apps.AdminportalConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.humanize',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
```

```

'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'ocs.urls'

TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': ['templates'],
    'DIRS': [BASE_DIR + "/templates"],
    'APP_DIRS': True,
    'OPTIONS': {
        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
            'home.context_processors.notifications',
        ],
    },
},
]

WSGI_APPLICATION = 'ocs.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-passwordValidators

AUTH_PASSWORD_VALIDATORS = [
{
    'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]

```

```
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'Asia/Kolkata'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# messages configuration

MESSAGE_TAGS = {
    messages.ERROR: 'danger',
}

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'static')
]

# STATIC_ROOT = os.path.join(BASE_DIR, 'static')

# MEDIA URL AND ROOT

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

OCS/url.py

"""college_proj URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/3.0/topics/http/urls/

Examples:

Function views

- 1. Add an import: from my_app import views*
- 2. Add a URL to urlpatterns: path('', views.home, name='home')*

Class-based views

- 1. Add an import: from other_app.views import Home*
- 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')*

Including another URLconf

- 1. Import the include() function: from django.urls import include, path*
- 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))*

"""

```
from django.contrib import admin
from django.urls import path,include
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("",include('home.urls')),
    path('student/',include('student.urls')),
    path('teacher/',include('teacher.urls')),
    path('adminportal/',include('adminportal.urls')),
```

```
]+static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

home/models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.utils import timezone

# Create your models here.
class Class(models.Model):
    class_name = models.CharField(max_length=50)
    strength = models.IntegerField()

    def __str__(self):
        return str(self.class_name)

class Subject(models.Model):
    class_name = models.ForeignKey(Class, on_delete=models.CASCADE)
    subject_name = models.CharField(max_length=50)

    def __str__(self):
        return self.subject_name

class Student(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    s_roll = models.IntegerField()
    s_class = models.ForeignKey(Class, on_delete=models.CASCADE, null=True, blank=True)
    s_dob = models.DateField(null=True, blank=True)
    s_contact = models.IntegerField(null=True, blank=True)
    s_gender = models.CharField(max_length=50, null=True, blank=True)
    updated = models.BooleanField(null=True, default=True)

    def __str__(self):
        return str(self.pk)

class Teacher(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    t_dob = models.DateField(null=True, blank=True)
    t_contact = models.IntegerField(null=True)
    t_gender = models.CharField(max_length=50, null=True)
    updated = models.BooleanField(null=True, default=True)

    def __str__(self):
        return str(self.pk)
```

home/view.py

```
from django.shortcuts import render,redirect
from django.contrib.auth.models import User,Group
from django.contrib.auth import authenticate,login,logout
from .templatetags import extras
from django.contrib import messages
from django.db.models.signals import post_delete
from django.dispatch import receiver
from .models import Student,Teacher,Class
from django.db import IntegrityError
import sys

# Check Exceptions
class PasswordNotMatch(Exception):
    pass

class EmptyControlNumber(Exception):
    pass

class WeakPassword(Exception):
    pass

class EmptyPassword(Exception):
    pass

# Create your views here.
def index(request):

    return render(request,'home/index.html')

def studRegister(request):
    """docstring for studRegister"""
    group = Group.objects.get(name='Student')
    if request.method == 'POST':
        try:
            fname = request.POST['fname']
            lname = request.POST['lname']
            control = request.POST['control']
            email = request.POST['email']
            cont = request.POST['cont']
            dob = request.POST['dob']
            gender = request.POST['gender']
            s_class = request.POST['class']
            roll = request.POST['roll']
            password = request.POST['password']
            con_password = request.POST['con_password']
            if password != con_password:
                raise PasswordNotMatch
            if len(control.replace(" ",""))==0:
                raise EmptyControlNumber
            if len(password)<=6:
                raise WeakPassword
        except IntegrityError:
            messages.error(request,"User already exists")
    else:
        messages.info(request,"Please enter valid details")
```

```

if len(password.replace(" ",""))<=6:
    raise EmptyPassword
user_created = False

stud_class = Class.objects.get(class_name=s_class)
user = User.objects.create_user(control,email,password,first_name=fname,last_name=lname)
user_created = True
user.groups.add(group)
student = Student.objects.create(
    user = user,
    s_roll = roll,
    s_class = stud_class,
    s_dob = dob,
    s_contact = cont,
    s_gender = gender,
)
except PasswordNotMatch:
    messages.error(request,'password and confirm password not match')

except EmptyControlNumber:
    messages.error(request,'Control number is Empty')

except WeakPassword:
    messages.error(request,'Password must be greater than 6 characters')

except EmptyPassword:
    messages.error(request,'Password is empty')
except IntegrityError:
    messages.error(request,'Control number already exists')
except:
    if user_created:
        user.delete()
    messages.error(request,sys.exc_info()[0])

else:
    messages.success(request,'Registration Successfull')
classes = Class.objects.all()
params = {
    "classes" : classes,
}
return render(request,'home/stud_register.html',params)

def teacherRegister(request):
    group = Group.objects.get(name='Teacher')
    if request.method == 'POST':
        try:
            fname = request.POST['fname']
            lname = request.POST['lname']
            control = request.POST['control']
            email = request.POST['email']
            cont = request.POST['cont']
            dob = request.POST['dob']
            gender = request.POST['gender']
            password = request.POST['password']
            con_password = request.POST['con_password']
            if password != con_password:

```

```

        raise PasswordNotMatch
if len(control.replace(" ",""))==0:
    raise EmptyControlNumber
if len(password)<=6:
    raise WeakPassword
if len(password.replace(" ",""))<=6:
    raise EmptyPassword
user_created = False
user = User.objects.create_user(control,email,password,first_name=fname,last_name=lname)
user_created = True
user.groups.add(group)
group.user_set.add(user)
teacher = Teacher.objects.create(
    user = user,
    t_dob = dob,
    t_contact = cont,
    t_gender = gender,
)
except PasswordNotMatch:
    messages.error(request,'password and confirm password not match')

except EmptyControlNumber:
    messages.error(request,'Control number is Empty')

except WeakPassword:
    messages.error(request,'Password must be greater than 6 characters')

except EmptyPassword:
    messages.error(request,'Password is empty')

except IntegrityError:
    messages.error(request,'Control number already exists')

except:
    if user_created:
        user.delete()
        print(sys.exc_info()[0])
        messages.error(request,sys.exc_info()[0])
    else:
        messages.success(request,'Registration Successfull')
return render(request,'home/teacher_register.html')

def studLogin(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method == "POST":
        # try:
        control = request.POST['control']
        password = request.POST['password']
        user = authenticate(username=control,password=password)
        if user is not None and extras.has_group(user,'Student'):
            login(request,user)
            messages.success(request,'Welcome '+user.first_name+' '+user.last_name)
            return redirect('student/')
            messages.success(request,'Welcome '+user.first_name+' '+user.last_name)
        else:

```

```

        messages.error(request,'Invalid Credentials')
        return render(request,'home/stud_login.html')

def teacherLogin(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method == "POST":
        # try:
        control = request.POST['control']
        password = request.POST['password']
        user = authenticate(username=control,password=password)
        if user is not None and extras.has_group(user,'Teacher'):
            login(request,user)
            return redirect('teacher/')
            messages.success(request,'Welcome '+user.first_name+' '+user.last_name)
        else:
            messages.error(request,'Invalid Credentials')
        return render(request,'home/teacher_login.html')

def demo(request):
    return render(request,'home/demo.html')

# APIs
def userLogout(request):
    logout(request)
    return redirect('/')

# signals(Triggers)
@receiver(post_delete, sender=Student)
def deleteStudentUser(sender, instance, *args, **kwargs):
    user = User.objects.get(username=instance.user.username).delete()

@receiver(post_delete, sender=Teacher)
def deleteTeacherUser(sender, instance, *args, **kwargs):
    user = User.objects.get(username=instance.user.username).delete()

```

home/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
    path("",views.index,name='index'),
    path('stud_register',views.studRegister,name='stud_register'),
    path('teacher_register',views.teacherRegister,name='teacher_register'),
    path('stud_login',views.studLogin,name='stud_login'),
    path('teacher_login',views.teacherLogin,name='teacher_login'),
    path('logout',views.userLogout,name='logout'),
    path('demo',views.demo)
]
```

student/models.py

```
from django.db import models
from django.contrib.auth.models import User
from assignment.models import Assignment,MCQ
from home.models import Class
from django.utils import timezone
# Create your models here.

class MCQResult(models.Model):
    assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
    student = models.ForeignKey(User,on_delete=models.CASCADE)
    total_que = models.IntegerField()
    attemp = models.IntegerField()
    correct = models.IntegerField()
    marks = models.IntegerField()
    def __str__(self):
        return str(self.pk)

class AttempQuestion(models.Model):
    student = models.ForeignKey(User,on_delete=models.CASCADE)
    #assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
    mcq = models.ForeignKey(MCQ,on_delete=models.CASCADE)
    s_ans = models.CharField(max_length=30)
    def __str__(self):
        return str(self.pk)

class File(models.Model):
```

```

student = models.ForeignKey(User,on_delete=models.CASCADE)
assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
question = models.TextField(max_length=300,null=True)
def student_name(obj, filename):
    return str("assignments/"+obj.assignment.subject.subject_name+"/"+obj.student.username+"/"+filename)
file = models.FileField(upload_to=student_name,null=True)

class NotificationForStudent(models.Model):
    date_filled = models.DateTimeField(default=timezone.now)
    assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
    notify_class = models.ForeignKey(Class,on_delete=models.CASCADE,null=True)
    def __str__(self):
        return str(self.pk)

```

student/view.py

```

from django.shortcuts import render,redirect
from home.models import Student,Class
from assignment.models import Assignment,MCQ,FileUpload
from teacher.models import AssignmentsLog
from .models import MCQResult,AttempQuestion,File
from django.contrib.auth.models import User
from home.templatetags import extras
from django.contrib import messages
from django.db.models.signals import post_save
from django.dispatch import receiver
from datetime import date
from django.utils import timezone
from home.views import PasswordNotMatch
# Create your views here.

```

```

def studIndex(request):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Student'):
            assignment = []
            student = Student.objects.get(user=request.user)
            if student.updated:
                ass = Assignment.objects.filter(ass_class=student.s_class).order_by("-date")
                for i in ass:
                    if i.last_date>date.today():
                        assignment.append(i)

            params = {
                'student':student,
                'assignments':assignment
            }
            else:
                return redirect('/student/update_data')
            else:
                return redirect('/')
        else:

```

```

    return redirect('/')
    return render(request,'student/index.html',params)

def updateData(request):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Student'):
            student = Student.objects.get(user=request.user)
            if not student.updated:
                classes = Class.objects.all().order_by("class_name")
                if request.method == "POST":
                    try:
                        first_name = request.POST['fname']
                        last_name = request.POST['lname']
                        control = request.POST.get('control')
                        class_name = Class.objects.get(class_name=request.POST['class'])
                        roll_no = request.POST['roll']
                        email = request.POST['email']
                        contact = request.POST['cont']
                        dob = request.POST['dob']
                        gender = request.POST['gender']
                        cur_password = request.POST['current-password']
                        new_password = request.POST['password']
                        confirm_password = request.POST['con_password']
                        if new_password.replace(" ","") != confirm_password.replace(" ",""):
                            raise PasswordNotMatch
                        if request.user.check_password(cur_password):
                            user = User.objects.get(username=request.user)
                            user.first_name = first_name
                            user.last_name = last_name
                            user.email = email
                            user.set_password(new_password)
                            student.s_class = class_name
                            student.s_roll = roll_no
                            student.s_contact = contact
                            student.s_dob = dob
                            student.s_gender = gender
                            student.updated = True
                            user.save()
                            student.save()
                            messages.success(request,"Successfully data updated")
                        else:
                            messages.error(request,"Invalid Current Password")
                    except PasswordNotMatch:
                        messages.error(request,'Password and Confirm Password should be same!')
                    params = {
                        'student' : student,
                        'classes' : classes,
                    }
                    return render(request,'student/update_data.html',params)
                else:
                    return redirect('/student')
            else:
                return redirect('/')
        else:
            return redirect('/')
    def assignment(request,slug):
        if request.user.is_authenticated:
            if extras.has_group(request.user,'Student'):
                assignment = Assignment.objects.get(slug = slug)
                mcqs = MCQ.objects.filter(assignment=assignment)

```

```

file_ass = FileUpload.objects.filter(assignment=assignment)
params = {
    'assignment':assignment,
    'mcqs':mcqs,
    'file_ass':file_ass,
}
else:
    return redirect('/')
else:
    return redirect('/')
return render(request,'student/assignment.html',params)

def mcqResult(request,slug):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Student'):
            correct = 0
            attemp = 0
            assignment = Assignment.objects.get(slug=slug)
            mcq = MCQ.objects.filter(assignment=assignment)
            total_quetions = len(mcq)
            AttempQuestion.objects.all().delete()
            if request.method == 'POST':
                for i in mcq:
                    ans = request.POST.get("ans"+str(i.pk))
                    if ans != None:
                        attemp += 1
                        if str(ans).replace(" ","") == str(i.ans).replace(" ",""):
                            correct += 1
                        attempt_que = AttempQuestion.objects.create(
                            student = request.user,
                            mcq = i,
                            s_ans = ans,
                        )
                    else:
                        continue
            try:
                res = MCQResult.objects.get(student=request.user,assignment=assignment)
                res.attemp = attemp
                res.total_que = total_quetions
                res.correct = correct
                res.marks = correct
                res.save()
            except MCQResult.DoesNotExist:
                res = MCQResult.objects.create(
                    assignment = assignment,
                    student = request.user,
                    total_que = total_quetions,
                    attemp = attemp,
                    correct = correct,
                    marks = correct,
                )
            attemp_questions = AttempQuestion.objects.filter(student=request.user)

            params = {
                "assignment" : assignment,
                "attemp_questions" : attemp_questions,
                "res" : res,
            }

```

```

        return render(request,"student/mcq_result.html",params)
    else:
        return redirect("/")
else:
    return redirect("/")

def fileSubmit(request,slug):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Student'):
            assignment = Assignment.objects.get(slug=slug)
            questions = FileUpload.objects.filter(assignment = assignment)
            try:
                log = AssignmentsLog.objects.get(student=request.user,assignment=assignment)
                log.submit_on = timezone.now()
            except AssignmentsLog.DoesNotExist:
                log = AssignmentsLog.objects.create(
                    student=request.user,
                    assignment=assignment,
                )
            if request.method == "POST":
                for i in questions:
                    try:
                        file = request.FILES.get("file"+str(i.pk))
                        if file is not None:
                            file_upload = File.objects.get(
                                student=request.user,
                                assignment=assignment,
                                question=i.desc,
                            )
                            file_upload.file.delete(save=True)
                            file_upload.file = file
                            file_upload.save()
                        else:
                            continue
                    except File.DoesNotExist:
                        if file is not None:
                            file_upload = File.objects.create(
                                student = request.user,
                                assignment=assignment,
                                question = i.desc,
                                file = file,
                            )
                        else:
                            continue
                    else:
                        messages.success(request,"File uploaded!")
                return redirect("/student")
            else:
                return redirect("/")
        else:
            return redirect("/")

```

```

# signals (Triggers)
@receiver(post_save, sender=MCQResult)
def addLog(sender, instance, created, **kwargs):
    if created:
        log = AssignmentsLog.objects.create(
            assignment = instance.assignment,
            student = instance.student,
            result = instance,
        )
    else:
        try:
            log = AssignmentsLog.objects.get(assignment=instance.assignment,student=instance.student)
            log.submit_on = timezone.now()
            log.result = instance
            log.save()
        except AssignmentsLog.DoesNotExist:
            log = AssignmentsLog.objects.create(
                assignment = instance.assignment,
                student = instance.student,
                result = instance,
            )

```

student/urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path('',views.studIndex,name='stud_home'),
    path('assignment/<str:slug>',views.assignment,name='assignment'),
    path('mcq_result/<str:slug>',views.mcqResult,name="mcq_result"),
    path('file_upload/<str:slug>',views.fileSubmit,name="file_upload"),
    path('update_data',views.updateData,name='update_data'),
]

```

teacher/models.py

```

from django.db import models
from django.contrib.auth.models import User
from assignment.models import Assignment
from student.models import MCQResult
from home.models import Teacher
from django.utils import timezone

# Create your models here.
class AssignmentsLog(models.Model):
    assignment = models.ForeignKey(Assignment, on_delete=models.CASCADE)
    student = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
    submit_on = models.DateTimeField(default=timezone.now)
    result = models.ForeignKey(MCQResult, on_delete=models.CASCADE, null=True, blank=True)
    def __str__(self):

```

```

    return str(self.pk)

class NotificationsForTeacher(models.Model):
    date_filled = models.DateTimeField(default=timezone.now)
    assignment = models.ForeignKey(Assignment, on_delete=models.CASCADE)
    student = models.ForeignKey(User, on_delete=models.CASCADE)
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE, null=True, blank=True)
    def __str__(self):
        return str(self.pk)

```

teacher/view.py

```

from django.shortcuts import render, redirect
from assignment.models import Assignment, MCQ, FileUpload
from .models import AssignmentsLog, NotificationsForTeacher
from home.models import Class, Subject
from student.models import MCQResult, File, NotificationForStudent
from home.models import Student, Teacher
from django.contrib.auth.models import User
from django.contrib import messages
from home.templatetags import extras
from .templatetags import app_tags
from datetime import date, timedelta, datetime
from django.conf import settings
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.utils import timezone
from home.views import PasswordNotMatch
import sys
# Create your views here.
def teacherIndex(request):
    if request.user.is_authenticated:
        if extras.has_group(request.user, 'Teacher'):
            assignment = []
            teacher = Teacher.objects.get(user=request.user)
            if teacher.updated:
                ass = Assignment.objects.filter(teacher=request.user).order_by("-date")
                for i in ass:
                    # if i.last_date < date.today() + timedelta(days=5):
                    assignment.append(i)
                params = {
                    'assignments': assignment
                }
            else:
                return redirect('/teacher/update_data')
        else:
            return redirect('/')
    else:
        return redirect('/')
    return render(request, 'teacher/index.html', params)

def updateData(request):
    if request.user.is_authenticated:
        if extras.has_group(request.user, 'Teacher'):
            teacher = Teacher.objects.get(user=request.user)

```

```

if not teacher.updated:
    if request.method == "POST":
        try:
            first_name = request.POST['fname']
            last_name = request.POST['lname']
            control = request.POST.get('control')
            email = request.POST['email']
            contact = request.POST['cont']
            dob = request.POST['dob']
            gender = request.POST['gender']
            cur_password = request.POST['current-password']
            new_password = request.POST['password']
            confirm_password = request.POST['con_password']
            if new_password.replace(" ","") != confirm_password:
                raise PasswordNotMatch
            if request.user.check_password(cur_password):
                user = User.objects.get(username=request.user)
                user.first_name = first_name
                user.last_name = last_name
                user.email = email
                user.set_password(new_password)
                teacher.t_dob = dob
                teacher.t_contact = contact
                teacher.t_gender = gender
                teacher.updated = True
                user.save()
                teacher.save()
                messages.success(request,"Successfully data updated")
            else:
                messages.error(request,"Invalid Current Password")
        except PasswordNotMatch:
            messages.error(request,'Password and Confirm Password should be same!')
        params = {
            'teacher':teacher
        }
        return render(request,'teacher/update_data.html',params)
    else:
        return redirect('/teacher')
    else:
        return redirect('/')
else:
    return redirect('/')

def addAssignment(request):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Teacher'):
            if request.method=="POST":
                ass_class = request.POST['class']
                sub = request.POST['subject']
                due_date = request.POST['last_date']
                ass_type = request.POST['type']
                slug = ass_class+"-"+sub+"-"+ass_type+str(datetime.now())
                assignment_class = Class.objects.get(class_name=ass_class)
                subject = Subject.objects.get(subject_name=sub,class_name=assignment_class)
                assignment = Assignment.objects.create(
                    teacher = request.user,
                    ass_class = assignment_class,
                    subject = subject,
                    last_date = due_date,
                    ass_type = ass_type,

```

```

        slug = slug
    )
    return redirect('/teacher/assignment/'+str(assignment.slug))
classes = Class.objects.all()
subjects = Subject.objects.all()
params = {
    "classes" : classes,
    "subjects" : subjects,
}
return render(request,'teacher/add_assignment.html',params)
else:
    return redirect('/')
else:
    return redirect('/')

def updateDueDate(request,id):
if request.user.is_authenticated:
    if extras.has_group(request.user,'Teacher'):
        if request.method == "POST":
            assignment = Assignment.objects.get(pk=id)
            new_date = request.POST['due-date']
            assignment.last_date = new_date
            assignment.save()
            messages.success(request,"date updated succesfully!")
            return redirect('/teacher')
        else:
            return redirect('/')
    else:
        return redirect('/')

def assignment(request,slug):
if request.user.is_authenticated:
    if extras.has_group(request.user,'Teacher'):
        assignment = Assignment.objects.get(slug=slug)
        mcqs = MCQ.objects.filter(assignment=assignment)
        file_ass = FileUpload.objects.filter(assignment=assignment)
        if assignment.ass_type == "MCQ":
            if request.method == "POST":
                que = request.POST['que']
                opt_a = request.POST['opt_a']
                opt_b = request.POST['opt_b']
                opt_c = request.POST['opt_c']
                opt_d = request.POST['opt_d']
                ans = request.POST['ans']
                mcq = MCQ.objects.create(
                    assignment = assignment,
                    que = que,
                    opt1 = opt_a,
                    opt2 = opt_b,
                    opt3 = opt_c,
                    opt4 = opt_d,
                    ans = ans
                )
                return redirect('/teacher/assignment/'+str(assignment.slug))
            else:
                if request.method == "POST":
                    que = request.POST['que']
                    f_ass = FileUpload.objects.create(
                        assignment = assignment,
                        desc = que,

```

```

        }
    params = {
        'assignment':assignment,
        'mcqs':mcqs,
        'file_ass':file_ass,
    }
else:
    return redirect('/')

return render(request,'teacher/assignment.html',params)

def assignmentStatus(request,slug):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Teacher'):
            solved = []
            assignment = Assignment.objects.get(slug=slug)
            student = Student.objects.filter(s_class=assignment.ass_class)
            log = AssignmentsLog.objects.filter(assignment=assignment).order_by("-submit_on")
            solved_log = AssignmentsLog.objects.filter(assignment=assignment).values_list('student')
            unsolved = Student.objects.filter(s_class=assignment.ass_class).exclude(user_in=solved_log)
            file = File.objects.filter(assignment=assignment)
            count_student = len(student)
            solved_student = len(log)
            params = {
                'assignment':assignment,
                'log':log,
                'file':file,
                'range':len(file),
                'media':settings.MEDIA_URL,
                'count_student':count_student,
                'solved_student':solved_student,
                'unsolved':unsolved,
                'students':student,
            }
            return render(request,'teacher/assignment_status.html',params)
        else:
            return redirect('/')
    else:
        return redirect('/')
# API FOR TEACHER

def deleteAssignment(request,slug):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Teacher'):
            ass = Assignment.objects.get(slug=slug)
            ass.delete()
            return redirect('/teacher')
        else:
            return redirect('/')
    else:
        return redirect('/')

def deleteQuetion(request,id):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Teacher'):
            que = MCQ.objects.get(pk=id)
            ass = Assignment.objects.get(pk=que.assignment.pk)
            que.delete()
            return redirect('/teacher/assignment/'+ass.slug)

```

```

else:
    return redirect('/')
else:
    return redirect('/')

def deleteFileQuestion(request,id):
    if request.user.is_authenticated:
        if extras.has_group(request.user,'Teacher'):
            que = FileUpload.objects.get(pk=id)
            ass = Assignment.objects.get(pk=que.assignment.pk)
            que.delete()
            return redirect("/teacher/assignment/"+ass.slug)
        else:
            return redirect('/')
    else:
        return redirect('/')

# signals (Triggers)
@receiver(post_save, sender=Assignment)
def addNotification(sender,instance,created,**kwargs):
    if created:
        NotificationForStudent.objects.create(
            assignment = instance,
            notify_class = instance.ass_class,
        )

    @receiver(post_save, sender=AssignmentsLog)
    def addNotificationForTeacher(sender,instance,created,**kwargs):
        teacher = Teacher.objects.get(user=instance.assignment.teacher)
        if created:
            notification = NotificationsForTeacher.objects.create(
                assignment = instance.assignment,
                student = instance.student,
                teacher = teacher,
            )
        else:
            try:
                notification = NotificationsForTeacher.objects.get(assignment=instance.assignment,student=instance.student)
                notification.date_filled = timezone.now()
                notification.assignment = instance.assignment
                notification.student = instance.student
                notification.teacher = teacher
                notification.save()
            except NotificationsForTeacher.DoesNotExist:
                notification = NotificationsForTeacher.objects.create(
                    assignment = instance.assignment,
                    student = instance.student,
                    teacher = teacher
                )

```

assigments/model.py

```

from django.db import models
from django.utils import timezone
from django.contrib.auth.models import User
from home.models import Class,Subject

# Create your models here.

```

```

class Assignment(models.Model):
    date = models.DateTimeField(default=timezone.now)
    teacher = models.ForeignKey(User,on_delete=models.CASCADE)
    ass_class = models.ForeignKey(Class,on_delete=models.CASCADE,null=True)
    subject = models.ForeignKey(Subject,on_delete=models.CASCADE,null=True)
    last_date = models.DateField(null=True)
    ass_type = models.CharField(max_length=30)
    slug = models.CharField(max_length=80,null=True)

    def __str__(self):
        return str(self.pk)

class MCQ(models.Model):
    assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
    que = models.TextField(max_length=3000)
    opt1 = models.CharField(max_length=30)
    opt2 = models.CharField(max_length=30)
    opt3 = models.CharField(max_length=30)
    opt4 = models.CharField(max_length=30)
    ans = models.CharField(max_length=30)

    def __str__(self):
        return str(self.pk)

class FileUpload(models.Model):
    assignment = models.ForeignKey(Assignment,on_delete=models.CASCADE)
    desc = models.TextField(max_length=3000)

    def __str__(self):
        return str(self.pk)

```

adminportal/view.py

```

from django.shortcuts import render,redirect
from django.contrib.auth import authenticate,login,logout
from django.contrib import messages
from django.db.models import Q
from django.contrib.auth.models import User,Group
from home.models import Student,Teacher,Class,Subject
import sys

# Create your views here.
def adminIndex(request):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            students = Student.objects.all().order_by("s_class")
            teachers = Teacher.objects.all().order_by("user")
            classes = Class.objects.all().order_by("class_name")
            subjects = Subject.objects.all().order_by("class_name")
            if request.GET.get('filter-students') is not None and request.GET.get('filter-students')!="All":
                class_name = request.GET.get('filter-students')
                s_class = Class.objects.get(class_name=class_name)
                students = Student.objects.filter(s_class = s_class).order_by("user")
            if request.GET.get('filter-subjects') is not None and request.GET.get('filter-subjects') != "All":
                76 | Page

```

```

class_name = request.GET.get('filter-subjects')
s_class = Class.objects.get(class_name=class_name)
subjects = Subject.objects.filter(class_name=s_class).order_by("subject_name")

if request.GET.get('search-student') is not None:
    querry = request.GET.get('search-student')
    users = User.objects.filter(
        Q(first_name_icontains=querry) | Q(last_name_icontains=querry) | Q(username_icontains=querry)
    )
    students = Student.objects.filter(user_in=users)

if request.GET.get('search-teacher') is not None:
    querry = request.GET.get('search-teacher')
    users = User.objects.filter(
        Q(first_name_icontains=querry) | Q(last_name_icontains=querry) | Q(username_icontains=querry)
    )
    teachers = Teacher.objects.filter(user_in=users)

if request.GET.get('search-class') is not None:
    querry = request.GET.get('search-class')
    classes = Class.objects.filter(
        Q(class_name_icontains=querry) | Q(strength_icontains=querry)
    )

if request.GET.get('search-subject') is not None:
    querry = request.GET.get('search-subject')
    subjects = Subject.objects.filter(subject_name_icontains=querry)

params = {
    'students':students,
    'teachers':teachers,
    'classes':classes,
    'subjects':subjects,
}
return render(request,'adminportal/index.html',params)
else:
    return redirect("/adminportal/admin_login")
else:
    return redirect("/adminportal/admin_login")

def saveStudent(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                student = Student.objects.get(pk=id)
                user = User.objects.get(username=student.user.username)
                class_name = Class.objects.get(class_name=request.POST['class']+str(student.pk))
                user.first_name = request.POST['first-name']
                user.last_name = request.POST['last-name']
                student.s_roll = request.POST['roll-no']
                student.s_class = class_name
                user.username = request.POST['control']
                student.s_dob = request.POST['dob']
                student.s_contact = request.POST['contact']
                user.email = request.POST['email']
                student.s_gender = request.POST['gender'+str(student.pk)]
                student.save()
                user.save()
                messages.success(request,"Save Successfully")

```

```

        return redirect('/adminportal')
    else:
        return redirect("/adminportal/admin_login")
else:
    return redirect("/adminportal/admin_login")

def saveTeacher(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                teacher = Teacher.objects.get(pk=id)
                user = User.objects.get(username=teacher.user.username)
                user.first_name = request.POST['first-name']
                user.last_name = request.POST['last-name']
                user.username = request.POST['control']
                teacher.t_dob = request.POST['dob']
                teacher.t_contact = request.POST['contact']
                user.email = request.POST['email']
                teacher.t_gender = request.POST['gender'+str(teacher.pk)]
                teacher.save()
                user.save()
                messages.success(request,"save succesfully")
                return redirect('/adminportal')
            else:
                return redirect("/adminportal/admin_login")
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

def saveClass(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                cls = Class.objects.get(pk=id)
                cls.class_name = request.POST['class-name']
                cls.strength = request.POST['strength']
                cls.save()
                messages.success(request,"Save Successfully")
                return redirect('/adminportal')
            else:
                return redirect("/adminportal/admin_login")
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

def saveSubject(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                subject = Subject.objects.get(pk=id)
                class_name = Class.objects.get(class_name=request.POST['class'+str(subject.pk)])
                Subject.subject_name = request.POST['subject-name']
                subject.class_name = class_name
                subject.save()
                messages.success(request,"save succesfully")
                return redirect('/adminportal')
            else:
                return redirect("/adminportal/admin_login")
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

```

```

def adminLogin(request):
    if request.method == "POST":
        control = request.POST['control']
        password = request.POST['password']
        user = authenticate(username=control,password=password)
        if user is not None:
            if user.is_superuser or user.username == "T-502480":
                login(request,user)
                return redirect('/adminportal')
            else:
                messages.error(request,"Not Admin User")
        else:
            messages.error(request,"Invalid Credentials")
    return render(request,"adminportal/login.html")

# APIs for admin
def adminLogout(request):
    logout(request)
    return redirect('/adminportal/admin_login')

def addStudent(request):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                first_name = request.POST['first-name'].replace(" ","")
                last_name = request.POST['last-name'].replace(" ","")
                roll_no = request.POST['roll-no'].replace(" ","")
                class_name = Class.objects.get(class_name=request.POST['class'])
                control = request.POST['control'].replace(" ","")
                password = first_name.capitalize()+"@"+control
                try:
                    user_created = False
                    group = Group.objects.get(name = "Student")
                    user = User.objects.create_user(
                        username = control,
                        password = password,
                        first_name = first_name,
                        last_name = last_name
                    )
                    user_created = True
                    user.groups.add(group)
                    student = Student.objects.create(
                        user = user,
                        s_roll = roll_no,
                        s_class = class_name,
                        updated = False
                    )
                except:
                    if user_created:
                        user.delete()
                    messages.error(request,sys.exc_info()[0])
                    messages.success(request,"Add Student successfully")
                return redirect('/adminportal')
            else:
                return redirect("/adminportal/admin_login")
        else:
            return redirect("/adminportal/admin_login")

def addTeacher(request):
    if request.user.is_authenticated:

```

```

if request.user.username == "T-502480" or request.user.is_superuser:
    if request.method == "POST":
        first_name = request.POST['first-name'].replace(" ","")
        last_name = request.POST['last-name'].replace(" ","")
        control = request.POST['control'].replace(" ","")
        password = first_name.capitalize()+"@"+control
        try:
            user_created = False
            group = Group.objects.get(name = "Teacher")
            user = User.objects.create_user(
                username = control,
                password = password,
                first_name = first_name,
                last_name = last_name
            )
            user_created = True
            user.groups.add(group)
            teacher = Teacher.objects.create(
                user = user,
                updated = False
            )
        except:
            if user_created:
                user.delete()
                messages.error(request,sys.exc_info()[0])
                messages.success(request,"Add Student successfully")
            return redirect('/adminportal')
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")
def addClass(request):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                class_name = request.POST['class-name']
                strength = request.POST['strength']
                Class.objects.create(
                    class_name = class_name,
                    strength = strength
                )
                messages.success(request,"Add Class Succesfull")
                return redirect('/adminportal')
            else:
                return redirect("/adminportal/admin_login")
        else:
            return redirect("/adminportal/admin_login")
def addSubject(request):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            if request.method == "POST":
                subject_name = request.POST['subject-name']
                class_name = Class.objects.get(class_name=request.POST['class'])
                Subject.objects.create(
                    subject_name = subject_name,
                    class_name = class_name
                )
                messages.success(request,"Add Class Succesfull")

```

```

        return redirect('/adminportal')
    else:
        return redirect("/adminportal/admin_login")
else:
    return redirect("/adminportal/admin_login")

def deleteStudent(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            student = Student.objects.get(pk=id)
            user = User.objects.get(username=student.user.username).delete()
            messages.success(request,"Delete Succesfully")
            return redirect('/adminportal')
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

def deleteTeacher(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            teacher = Teacher.objects.get(pk=id)
            user = User.objects.get(username=teacher.user.username).delete()
            messages.success(request,"Delete Succesfully")
            return redirect('/adminportal')
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

def deleteClass(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            class_name = Class.objects.get(pk=id).delete()
            messages.success(request,"Delete Succesfully")
            return redirect('/adminportal')
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

def deleteSubject(request,id):
    if request.user.is_authenticated:
        if request.user.username == "T-502480" or request.user.is_superuser:
            subject = Subject.objects.get(pk=id).delete()
            messages.success(request,"Delete Succesfully")
            return redirect('/adminportal')
        else:
            return redirect("/adminportal/admin_login")
    else:
        return redirect("/adminportal/admin_login")

```

adminportal/url.py

```
from django.urls import path
from . import views

urlpatterns = [
    path("",views.adminIndex,name='adminportal'),
    path('admin_login',views.adminLogin,name='admin_login'),
    path('save_student/<int:id>',views.saveStudent,name='save_student'),
    path('save_teacher/<int:id>',views.saveTeacher,name='save_teacher'),
    path('save_class/<int:id>',views.saveClass,name='save_class'),
    path('save_subject/<int:id>',views.saveSubject,name='save_subject'),
    # APIs
    path('admin_logout',views.adminLogout,name='admin_logout'),
    path('delete_student/<int:id>',views.deleteStudent,name='delete_student'),
    path('delete_teacher/<int:id>',views.deleteTeacher,name='delete_teacher'),
    path('delete_class/<int:id>',views.deleteClass,name='delete_class'),
    path('delete_subject/<int:id>',views.deleteSubject,name='delete_subject'),
    path('add_student',views.addStudent,name='add_student'),
    path('add_teacher',views.addTeacher,name='add_teacher'),
    path('add_class',views.addClass,name='add_class'),
    path('add_subject',views.addSubject,name='add_subject'),
```

HTML PAGES

templates/base.html

```
<!DOCTYPE html>
<html>

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>{% block title %}{% endblock %}</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
      integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
      crossorigin="anonymous">

<!-- font awesome link -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<!--summernote -->
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
      integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
      crossorigin="anonymous"></script>
<link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.js"></script>

<!-- custom stylesheet -->
{% block css %}

{% endblock css %}
<style>
.top-header {
  height: 50px;
  background-image: -webkit-linear-gradient(-45deg, #405de6, #e1306c 50%, #ffdc80) !important;
  color: white;
}

.top-header a {
  font-size: 12px;
  transition: 0.6s;
}

.top-header i {
  color: white;
  opacity: 0.5;
}

```

```
.top-header a:hover {  
    text-decoration: none;  
    opacity: 0.5;  
}
```

```
.icon {  
    margin: 0 2px;  
    height: 40px;  
    width: 40px;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}
```

```
.icon a {  
    font-size: 25px;  
    height: 100%;  
}
```

```
.icon i {  
    opacity: 1;  
}
```

```
.brand-header a {  
    color: black;  
}
```

```
.brand-header a:hover {  
    text-decoration: none;  
}
```

```
.brand-header a h4 {  
    color: #0d2d62;  
}
```

```
.nav-active a {  
    color: #b80924;  
}
```

```
.nav-link {  
    color: black;  
    text-decoration: none;  
    font-weight: bold;  
    transition: 0.5s;  
}
```

```
.nav-link:hover {  
    color: #b80924;  
}
```

```
.notifications {  
    right: 0;  
    left: auto;
```

```
width: max-content;
max-width: 90vw;
height: max-content;
max-height: 80vh;
overflow-y: auto;
overflow-x: hidden;
}

.toggle-btn:hover {
color: #b80924;
}

@keyframes notify {
0% {
transform: scale(0);
opacity: 0.7;
}

100% {
transform: scale(1);
opacity: 1;
}
}

@media (max-width: 576px) {

.top-header,
.brand-header {
display: none;
}
}

@media (max-width: 768px) {

.top-header,
.brand-header {
display: none !important;
}
}

@media (min-width: 992px) {
.brand {
display: none;
}
}

@media (min-width: 1200px) {
.brand {
display: none;
}
}
```

```

        }
    </style>
    <!-- custom javascript -->
</head>

<body>
    {% load extras %}

    <header>
        <div class="top-header container-fluid">
            <div class="row h-100">
                <div class="col-6 d-flex align-items-center">
                    <i class="fa fa-map-marker"></i>
                    <a href="http://www.acharyamarathecollege.in/" class="text-white ml-3">Chembur Trombay Education
Society's N.G Acharya And D.K Marathe College </a>
                </div>
                <div class="col-6 ">
                    <div class="ml-auto h-100 d-flex align-items-center justify-content-end">
                        <i class="fa fa-sign-in"></i>
                        <a href="http://www.acharyamarathecollege.in/contact_us" class="text-white ml-1 mr-3">Contact us</a>
                        <div class="icon rounded-circle" style="background-color: #395b98;">
                            <a href="https://www.facebook.com/AMCollegeChemburOfficial/" target="_blank">
                                <i class="fa fa-facebook"></i>
                            </a>
                        </div>
                        <div class="icon rounded-circle" style="background-color: #2ca7e0;">
                            <a href="https://twitter.com/AcharyaMarathe" target="_blank" rel="noopener noreferrer">
                                <i class="fa fa-twitter"></i>
                            </a>
                        </div>
                        <div class="icon rounded-circle" style="background-color: #d95139;">
                            <a href="https://www.youtube.com/channel/UCWsdNzMJRMVl7prby6IPWxg" target="_blank"
rel="noopener noreferrer">
                                <i class="fa fa-google"></i>
                            </a>
                        </div>
                        <div class="icon rounded-circle" style="background-image: linear-gradient(-45deg,#ffdc80,#e1306c 50%,#405de6);">
                            <a href="https://www.instagram.com/ngacharyadkmarathecollege/?hl=en" target="_blank"
rel="noopener noreferrer">
                                <i class="fa fa-instagram"></i>
                            </a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <div class="brand-header bg-light">
            <a href="{% url 'index' %}">
                <div class="container py-3">
                    <div class="row h-100">
                        <div class="col-2 my-auto px-0 text-right">
                            
                        </div>
                        <div class="col-8 text-center">
                            <h4> Exam Portal</h4>
                            <h2><strong>Computer Science And information Technology </strong></h2>
                        </div>

```

```

<div class="col-2 px-0 text-left">
    
</div>
</div>
</div>
</a>
</div>
<nav class="navbar navbar-expand-lg bg-light">
    {% load static %}
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"
        style="outline: none;">
        <i class="fa fa-bars toggle-btn"></i>
    </button>
    <a class="navbar-brand text-dark brand text-left mr-auto" href="#">N.G. Acharya College</a>
    {% if request.user|has_group:"Student" %}
        <div class="dropdown d-lg-none show">
            <a class="dropdown-toggle nav-link" style="width: fit-content;" href="#" role="button"
                id="dropdownMenuLink"
                data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-bell"></i>
            </a>

            <div class="dropdown-menu notifications" aria-labelledby="dropdownMenuLink">
                {% if notifications %}

                    {% for notification in notifications %}
                        <div class="row">
                            <div class="col-12">
                                <h5><a href="/student/assignment/{{notification.assignment.slug}}" class="nav-link">New Assignment
                    for
                        {{notification.assignment.subject.subject_name}} by {{notification.assignment.teacher.first_name}}
                        {{notification.assignment.teacher.last_name}}</a></h5>
                    {% load humanize %}
                        <div class="col py-0 my-0 text-right"><small>{{notification.date_filled|naturaltime}}</small></div>
                    </div>
                </div>
                {% endfor %}
                {% else %}
                    <h5 class="text-center">No Notifications</h5>
                {% endif %}
            </div>
        </div>
    {% elif request.user|has_group:"Teacher" %}
        <div class="d-lg-none">
            <a href="{% url 'add_assignment'%}" class="nav-link m-0 p-0" style="width: 0;"><i class="fa fa-plus"></i></a>
        </div>
        <div class="dropdown d-lg-none show">
            <a class="dropdown-toggle nav-link ml-2 pr-0" style="width: fit-content;" href="#" role="button"
                id="dropdownMenuLink"
                data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                <i class="fa fa-bell"></i>
            </a>

            <div class="dropdown-menu notifications" aria-labelledby="dropdownMenuLink">
                {% if notifications %}
```

```

{% for notification in notifications %}
<div class="row">
  <div class="col-12">
    <h5><a href="/teacher/assignment_status/{{notification.assignment.slug}}" class="nav-link">{{notification.student.first_name}} {{notification.student.last_name}} submitted {{notification.assignment.subject.subject_name}} assignment of {{notification.assignment.ass_class}}</a></h5>
    {% load humanize %}
    <div class="col py-0 my-0 text-right"><small>{{notification.date_filled|naturaltime}}</small></div>
  </div>
</div>
{% endfor %}
{% else %}
<h5 class="text-center">No Notifications</h5>
{% endif %}
</div>
</div>
{% endif %}

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mx-auto">
    <li class="nav-item {% if request.get_full_path == '/' %}nav-active{% endif %}">
      <a class="nav-link" href="{% url 'index'%}">Home <span class="sr-only">(current)</span></a>
    </li>

    {% if request.user.is_authenticated %}
    <li
      class="nav-item dropdown {% if request.get_full_path == '/student/' or request.get_full_path == '/teacher/' %}nav-active{% endif %}">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false">
        {{request.user.first_name}} {{request.user.last_name}}
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">

        {% if request.user|has_group:"Student" %}
          <a class="dropdown-item text-dark" href="#">Assignments</a>
          <a class="dropdown-item text-dark" href="{% url 'stud_home'%}">{{request.user|group_name}}</a>
          <a class="dropdown-item text-dark" href="#" data-toggle="modal" data-target="#logout-confirm">Logout</a>
        {% else %}
          <a class="dropdown-item text-dark" href="{% url 'add_assignment'%}">Add Assignments</a>
          <a class="dropdown-item text-dark" href="{% url 'teacher_home'%}">{{request.user|group_name}}</a>
          <a class="dropdown-item text-dark" href="#" data-toggle="modal" data-target="#logout-confirm">Logout</a>
        {% endif %}
      </div>
    </li>
    {% else %}
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="loginDropdown" role="button" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="false">Log in</a>
      <div class="dropdown-menu" aria-labelledby="loginDropdown">
        <a href="{% url 'stud_login' %}" class="dropdown-item">Student Login</a>
        <a href="{% url 'teacher_login'%}" class="dropdown-item">Teacher Login</a>
      </div>
    </li>
    {% endif %}
  </ul>
</div>

```

```

</li>
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="signupDropdown" role="button" aria-haspopup="true"
    data-toggle="dropdown" aria-expanded="false">Sign up</a>
  <div class="dropdown-menu" aria-labelledby="singupDropdown">
    <a href="{% url 'stud_register' %}" class="dropdown-item">Student SingUp</a>
    <a href="{% url 'teacher_register' %}" class="dropdown-item">Teacher SingUp</a>

  </div>
</li>
{% endif %}
{% if request.user|has_group:"Student" %}
<li
  class="nav-item d-none d-lg-block {% if request.get_full_path == '/#notifications' %}nav-active{% endif %}"
dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
  toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">Notifications</a>
  <div class="dropdown-menu notifications" aria-labelledby="dropdownMenuLink">
    {% if notifications %}
      {% for notification in notifications %}
        <div class="row">
          <div class="col-12">
            <h5><a href="/student/assignment/{{notification.assignment.slug}}" class="nav-link">New Assignment
for
            {{notification.assignment.subject.subject_name}} by {{notification.assignment.teacher.first_name}}
            {{notification.assignment.teacher.last_name}}</a></h5>
            {% load humanize %}
            <div class="col py-0 my-0 text-right"><small>{{notification.date_filled|naturaltime}}</small></div>
          </div>
        </div>
      {% endfor %}
      {% else %}
        <h5 class="text-center">No Notifications</h5>
      {% endif %}
    </div>
  </li>

  {% elif request.user|has_group:"Teacher" %}
<li
  class="nav-item d-none d-lg-block {% if request.get_full_path == '/#notifications' %}nav-active{% endif %}"
dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-
  toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">Notifications</a>
  <div class="dropdown-menu notifications" aria-labelledby="dropdownMenuLink">
    {% if notifications %}
      {% for notification in notifications %}
        <div class="row">
          <div class="col-12">
            <h5><a href="/teacher/assignment_status/{{notification.assignment.slug}}"
              class="nav-link">{{notification.student.first_name}} {{notification.student.last_name}} submitted
              {{notification.assignment.subject.subject_name}} assignment of
              {{notification.assignment.ass_class}}</a></h5>
            {% load humanize %}
            <div class="col py-0 my-0 text-right"><small>{{notification.date_filled|naturaltime}}</small></div>
          </div>
        </div>
      {% endfor %}
    {% endfor %}
  </div>
</li>

```

```

    {% else %}
    <h5 class="text-center">No Notifications</h5>
    {% endif %}
    </div>
</li>

    {% endif %}
</ul>
<!-- <form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
</form> -->
</div>
</nav>
</header>
<!-- MODAL FRAME -->
<div class="modal fade" id="logout-confirm" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
aria-hidden="true">
<div class="modal-dialog" role="document">
    <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Confirm LogOut</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                <span aria-hidden="true">&times;</span>
            </button>
        </div>
        <div class="modal-body">
            Are You Sure want to Log out
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
            <a type="button" class="btn btn-primary" href="{% url 'logout' %}">Sure</a>
        </div>
    </div>
</div>
</div>
</div>

```

```

    {% for message in messages %}
<div class="alert alert-{{message.tags}} alert-dismissible fade show" role="alert">
    {{message}}
    <button type="button" class="close" data-dismiss="alert" aria-label="Close">
        <span aria-hidden="true">&times;</span>
    </button>
</div>
    {% endfor %}
    {% block body %}
    {% endblock %}

```

```

<section id="footer" style="position: relative;bottom: 0;left: 0" class="bg-light text-secondary">
    <h5 class="text-center">
        Developed By <strong>DNYANESHWAR PVT LTD.</strong>
    </h5>
    <div class="cont w-100 d-flex justify-content-center">
        <a href="https://github.com/dnyanesh2" class="text-secondary" target="_blank"><i
            class="fa fa-3x fa-github"></i></a>
    </div>
</section>

```

```

<script>
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function () {
  'use strict';
  window.addEventListener('load', function () {
    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.getElementsByClassName('needs-validation');
    // Loop over them and prevent submission
    var validation = Array.prototype.filter.call(forms, function (form) {
      form.addEventListener('submit', function (event) {
        if (form.checkValidity() === false) {
          event.preventDefault();
          event.stopPropagation();
        }
        form.classList.add('was-validated');
      }, false);
    });
  }, false);
})();
</script>

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIYzSFKEQXK现实中n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4lh7YwaYd1iqkjtj0Uod8GCExl30g8ifwB6"
crossorigin="anonymous"></script>

{% block script %}

{% endblock script %}
</body>

</html>

```

home/templates/home/stud_login.html

```

{% extends 'base.html' %}
{% block title %}Student Login{% endblock %}
{% block body %}
<div class="container mt-3">
  <h3 class="text-center">Student Login</h3>
  <form class="w-50 mx-auto" action="{% url 'stud_login' %}" method="post" accept-charset="utf-8">
    {% csrf_token %}
    <div class="row">
      <div class="col">
        <div class="form-group">
          <!-- <label for="control">Control Number</label> -->
          <input type="text" name="control" class="form-control" id="control" placeholder="Control Number" />
        </div>
      </div>
    </div>
  </form>
</div>

```

```

        </div>
    </div>
    <div class="row">
        <div class="col">
            <div class="form-group">
                <input type="password" name="password" id="password" class="form-control" placeholder="Password"/>
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col">
            <div class="form-group">
                <input type="submit" name="" id="" value="Log in" class="btn btn-info" />
            </div>
        </div>
    </div>

</div>
</form>
</div>

{% endblock %}

```

home/templates/home/stud_register.html

```

{% extends 'base.html' %}
{% block title %}Register as Student{% endblock %}
{% block body %}
<div class="container my-3">
    <div class="text-center">
        <h3>Register As Student</h3>
    </div>
    <form action="{% url 'stud_register' %}" method="post" accept-charset="utf-8" class="needs-validation"
novalidate >
        {% csrf_token %}
        <div class="row">
            <div class="col-sm-12 col-md-6 col-lg-6">
                <div class="form-group">
                    <label for="fname">First Name</label>
                    <input required type="text" class="form-control" id="fname" name="fname" placeholder="First Name">
                </div>
            </div>
            <div class="col-sm-12 col-md-6 col-lg-6">
                <div class="form-group">
                    <label for="lname">Last Name</label>
                    <input required type="text" class="form-control" id="lname" name="lname" placeholder="Last Name">
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col">
                <div class="form-group">
                    <label for="control">Control Number</label>

```

```

<input required type="text" class="form-control" id="control" name="control" placeholder="Control Number">
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="email">Email</label>
<input required type="email" class="form-control" id="email" name="email" placeholder="Email">
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="cont">Contact Number</label>
<input required type="text" name="cont" id="cont" class="form-control" placeholder="Contact Number"/>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="dob">Date of Birth</label>
<input required type="date" name="dob" id="dob" class="form-control"/>
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label class="col px-0">Gender: </label>
<div class="col px-0">

<div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="gender" id="male" value="male" required>
<label class="form-check-label" for="male">Male</label>
</div>
<div class="form-check form-check-inline">
<input required class="form-check-input" type="radio" name="gender" id="female" value="female">
<label class="form-check-label" for="female">Female</label>
</div>
<div class="form-check form-check-inline">
<input required class="form-check-input" type="radio" name="gender" id="other" value="other">
<label class="form-check-label" for="othet">Other</label>
</div>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="class">Class</label>
<select required class="form-control" name="class" id="class">
<option disabled selected>select</option>
{% for class in classes %}
<option>{{class.class_name}}</option>
{% endfor %}
</select>
</div>
</div>

```

```

<div class="col-sm-12 col-md-6 col-lg-6">
  <div class="form-group">
    <label for="roll">Roll Number</label>
    <input required type="text" name="roll" id="roll" placeholder="Roll Number" class="form-control"/>
  </div>
</div>
<div class="row">
  <div class="col-sm-12 col-md-6 col-lg-6">
    <div class="form-group">
      <label for="password">Password</label>
      <input required type="password" name="password" id="password" class="form-control"/>
    </div>
  </div>
  <div class="col-sm-12 col-md-6 col-lg-6">
    <div class="form-group">
      <label for="con_password">Confirm Password</label>
      <input required type="password" name="con_password" id="con_password" class="form-control"/>
    </div>
  </div>
</div>
<div class="row">
  <div class="col text-center">
    <input type="submit" value="Submit" class="btn btn-info"/>
    <input type="reset" value="Clear" class="btn btn-info"/>
  </div>
</div>
</form>
</div>

<!-- java script -->

{%
  endblock %}

```

{% endblock %}

home/templates/home/teacher_register.html

```

{% extends 'base.html' %}
{% block title %}Register as Student{% endblock %}
{% block body %}
<div class="container my-3">
  <div class="text-center">
    <h3>Register As Teacher</h3>
  </div>
  <form action="{% url 'teacher_register' %}" method="post" accept-charset="utf-8" class="needs-validation"
novalidate >
    {% csrf_token %}
    <div class="row">
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
          <label for="fname">First Name</label>
          <input required type="text" class="form-control" id="fname" name="fname" placeholder="First Name">
        </div>
      </div>
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">

```

```

<label for="lname">Last Name</label>
<input required type="text" class="form-control" id="lname" name="lname" placeholder="Last Name">
</div>
</div>
</div>
<div class="row">
<div class="col">
<div class="form-group">
<label for="control">Control Number</label>
<input required type="text" class="form-control" id="control" name="control" placeholder="Control Number">
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="email">Email</label>
<input required type="email" class="form-control" id="email" name="email" placeholder="Email">
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="cont">Contact Number</label>
<input required type="text" name="cont" id="cont" class="form-control" placeholder="Contact Number"/>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="dob">Date of Birth</label>
<input required type="date" name="dob" id="dob" class="form-control"/>
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label class="col px-0">Gender: </label>
<div class="col px-0">

<div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="gender" id="male" value="male" required>
<label class="form-check-label" for="male">Male</label>
</div>
<div class="form-check form-check-inline">
<input required class="form-check-input" type="radio" name="gender" id="female" value="female">
<label class="form-check-label" for="female">Female</label>
</div>
<div class="form-check form-check-inline">
<input required class="form-check-input" type="radio" name="gender" id="other" value="other">
<label class="form-check-label" for="othet">Other</label>
</div>
</div>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">

```

```

<label for="password">Password</label>
<input required type="password" name="password" id="password" class="form-control"/>
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="con_password">Confirm Password</label>
<input required type="password" name="con_password" id="con_password" class="form-control"/>
</div>
</div>
</div>
<div class="row">
<div class="col text-center">
<input type="submit" value="Submit" class="btn btn-info"/>
<input type="reset" value="Clear" class="btn btn-info"/>
</div>
</div>
</form>
</div>

<!-- java script -->
<script>
// Example starter JavaScript for disabling form submissions if there are invalid fields
(function() {
  'use strict';
  window.addEventListener('load', function() {
    // Fetch all the forms we want to apply custom Bootstrap validation styles to
    var forms = document.getElementsByClassName('needs-validation');
    // Loop over them and prevent submission
    var validation = Array.prototype.filter.call(forms, function(form) {
      form.addEventListener('submit', function(event) {
        if (form.checkValidity() === false) {
          event.preventDefault();
          event.stopPropagation();
        }
        form.classList.add('was-validated');
      }, false);
    });
  }, false);
})();
</script>

{% endblock %}

```

student/templates/student/assigments.html

```

{% extends 'base.html' %}
{% block title %}{{assignment.topic}}{% endblock %}
{% block body %}
<div class="container mt-3">
<h3>Teacher : {{assignment.teacher.first_name}} {{assignment.teacher.last_name}}</h3>
<h3>Subject : {{assignment.subject.subject_name}}</h3>
<h3>Last Date : {{assignment.last_date}}</h3>
{% if assignment.ass_type == 'MCQ' %}
<div id="mcq">
<div class="questions mt-5">

```

```

<h3 class="text-center">Multiple Choice Questions</h3>
{% for question in mcqs %}
<form action="/student/mcq_result/{{assignment.slug}}" method="post" accept-charset="utf-8" class="mt-3">
  {% csrf_token %}
  <div class="px-3 border border-dark my-2">
    <div class="form-row">
      <div class="col-12 col-sm-12 col-md-12">
        <div class="row">
          <div class="col-1 mx-0 px-0 text-center">
            {{forloop.counter}}.
          </div>
          <div class="col-11 mx-0 px-0 ">
            {{question.que | safe}}
          </div>
        </div>
      </div>
    </div>
    <div class="row mt-1">
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="custom-control custom-radio custom-control-inline">
          <input type="radio" name="ans{{question.pk}}" id="id_{{question.opt1}}" value="{{question.opt1}}"
        class="custom-control-input" />
          <label for="id_{{question.opt1}}" class="custom-control-label">{{question.opt1}}</label>
        </div>
      </div>
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="custom-control custom-radio custom-control-inline">
          <input type="radio" class="custom-control-input" name="ans{{question.pk}}" id="id_{{question.opt2}}"
        value="{{question.opt2}}" />
          <label for="id_{{question.opt2}}" class="custom-control-label">{{question.opt2}}</label>
        </div>
      </div>
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="custom-control custom-radio custom-control-inline">
          <input type="radio" class="custom-control-input" name="ans{{question.pk}}" id="id_{{question.opt3}}"
        value="{{question.opt3}}" />
          <label for="id_{{question.opt3}}" class="custom-control-label">{{question.opt3}}</label>
        </div>
      </div>
      <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="custom-control custom-radio custom-control-inline">
          <input type="radio" class="custom-control-input" name="ans{{question.pk}}" id="id_{{question.opt4}}"
        value="{{question.opt4}}" />
          <label for="id_{{question.opt4}}" class="custom-control-label">{{question.opt4}}</label>
        </div>
      </div>
    </div>
  {% endfor %}

  <div class="row mt-3">
    <div class="col text-center">
      <input type="submit" class="btn btn-primary" value="Submit" />
    </div>
  </div>
</form>
</div>
</div>
{% else %}

```

```

<div id="file-assignment">
    <div class="questions">
        <h3 class="text-center">Upload Files</h3>
        <form action="/student/file_upload/{{assignment.slug}}" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            {% for questions in file_ass %}
                <div class="row border border-dark my-3">
                    <div class="col">
                        <div class="row">
                            <div class="col-1 mx-0 px-0 text-center">
                                {{forloop.counter}}
                            </div>
                            <div class="col-11 mx-0 px-0">
                                {{questions.desc | safe}}
                            </div>
                        </div>
                    <div class="row my-4">
                        <div class="col">
                            <div class="custom-file w-50">
                                <input type="file" class="custom-file-input" name="file{{questions.pk}}"
onchange="dispName('{{questions.pk}}') id="file{{questions.pk}}">
                                <label class="custom-file-label" id="file-name{{questions.pk}}" for="customFile">Choose file</label>
                            </div>
                        </div>
                    </div>
                </div>
            {% endfor %}
            <div class="row">
                <div class="col text-center">
                    <input type="submit" value="Submit" class="btn btn-primary" />
                </div>
            </div>
        </form>
    </div>
</div>

{% endif %}
</div>

{% endblock %}
{% block script %}
<script>
function dispName(id){
    var fileName = document.getElementById("file"+id).files[0].name;
    document.getElementById("file-name"+id).innerText = fileName;
}
</script>
{% endblock script %}

```

student/templates/student/mcq_result.html

```
{% extends 'base.html' %}  
{% block title %}Result for MCQs{% endblock %}  
{% block css %}  
<style>  
.bar-container {  
    position: relative;  
    width: 100%;  
    height: 15px;  
    margin: 0 0;  
}  
.bar{  
    position: absolute;  
    top: 0;  
    left: 0;  
    height: 100%;  
    width: 100%;  
}  
.progress {  
    position: absolute;  
    display: none;  
    top: 0;  
    left: 0;  
    height: 100%;  
    width: 100%;  
    background: green;  
    animation: progress 1s ease-in-out forwards;  
}  
@keyframes progress {  
    0%{  
        transform: scaleX(0);  
        transform-origin: left;  
    }  
    100%{  
        transform: scaleX(1);  
        transform-origin: left;  
    }  
}  
</style>  
{% endblock css %}  
{% block body %}  


### Result



#### Total Questions :



{{res.total_que}}


```

```

        </div>
    </div>
    <div class="row">
        <div class="col">
            <h4>
                <strong>Attemp : </strong>
                {{res.attemp}}
            </h4>
        </div>
    </div>
    <div class="row">
        <div class="col">
            <h4>
                <strong>Correct : </strong>
                {{res.correct}}
            </h4>
        </div>
    </div>
    <div class="col-lg-6 col-sm-12 col-md-6">
        <div class="bar-container" style="height: 15px;">
            <div class="bar bg-light" id="bar"></div>
            <div class="progress" id="progress"></div>
        </div>
        <span id="show-per"></span>
    </div>
</div>
</div>
<% if attemp_questions %>
<div class="table-responsive">
    <table class="table table-sm table-hover">
        <thead>
            <tr>
                <th>#</th>
                <th>Question</th>
                <th class="text-center">Your Answer</th>
                <th class="text-center">Correct Answer</th>
            </tr>
        </thead>
        <tbody>
            <% for question in attemp_questions %>
            <tr>
                <th>{{forloop.counter}}</th>
                <td>{{question.mcq.que|safe|slice:"20"}}</td>
                <td class="text-center">{{question.s_ans}}</td>
                <% if question.s_ans == question.mcq.ans %>
                    <td class="bg-success text-white text-center">{{question.mcq.ans}}</td>
                <% else %>
                    <td class="bg-danger text-white text-center">{{question.mcq.ans}}</td>
                <% endif %>
            </tr>
            <% endfor %>
        </tbody>
    </table>
</div>
<% endif %>
</div>
<% endblock %>

```

```

{% block script %}
<script>
var marks = {{ res.correct }};
var total = {{ res.total_que }};
var per = marks / total * 100;
var progress = document.getElementById("progress")
progress.style.width = per+"%";
progress.style.display = "initial";
console.log(marks);
console.log(total);
console.log(per);
document.getElementById("show-per").innerHTML = per+"%";
if(per<40){
    progress.style.backgroundColor = "red";
}
</script>
{% endblock script %}

```

student/templates/student/update_data.html

```

{% extends 'base.html' %}
{% block title %}Update your Data{% endblock title %}
{% block css %}

{% endblock css %}
{% block body %}


### Update Your Information


<form action="/student/update_data" method="post" accept-charset="utf-8" class="needs-validation"
novalidate>
    {% csrf_token %}
    <div class="row">
        <div class="col-sm-12 col-md-6 col-lg-6">
            <div class="form-group">
                <label for="fname">First Name</label>
                <input required type="text" class="form-control" id="fname" name="fname" placeholder="First Name"
value="{{student.user.first_name}}">
            </div>
        </div>
        <div class="col-sm-12 col-md-6 col-lg-6">
            <div class="form-group">
                <label for="lname">Last Name</label>
                <input required type="text" class="form-control" id="lname" name="lname" placeholder="Last Name"
value="{{student.user.last_name}}">
            </div>
        </div>
    <div class="row">
        <div class="col">
            <div class="form-group">
                <label for="control">Control Number</label>


```

```

        <input required type="text" class="form-control" id="control" name="control"
               placeholder="Control Number" value="{{student.user.username}}" disabled>
    </div>
</div>
</div>
<div class="row">
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="class">Class</label>
            <select required class="form-control" name="class" id="class">
                {% for class in classes %}
                    <option {% if student.s_class.class_name == class.class_name %}selected{% endif
                %}>{{class.class_name}}</option>
                {% endfor %}
            </select>
        </div>
    </div>
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="roll">Roll Number</label>
            <input required type="text" name="roll" id="roll" placeholder="Roll Number" class="form-control"
value="{{student.s_roll}}>
        </div>
    </div>
<div class="row">
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="email">Email</label>
            <input required type="email" class="form-control" id="email" name="email" placeholder="Email">
        </div>
    </div>
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="cont">Contact Number</label>
            <input required type="text" name="cont" id="cont" class="form-control"
placeholder="Contact Number" />
        </div>
    </div>
</div>
<div class="row">
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="dob">Date of Birth</label>
            <input required type="date" name="dob" id="dob" class="form-control" />
        </div>
    </div>
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label class="col px-0">Gender: </label>
            <div class="col px-0">
                <div class="form-check form-check-inline">
                    <input class="form-check-input" type="radio" name="gender" id="male" value="male" required>
                    <label class="form-check-label" for="male">Male</label>
                </div>
                <div class="form-check form-check-inline">
                    <input required class="form-check-input" type="radio" name="gender" id="female"
value="female" required>
                    <label class="form-check-label" for="female">Female</label>
                </div>
            </div>
        </div>
    </div>

```

```

        value="female">
      <label class="form-check-label" for="female">Female</label>
    </div>
    <div class="form-check form-check-inline">
      <input required class="form-check-input" type="radio" name="gender" id="other"
        value="other">
      <label class="form-check-label" for="other">Other</label>
    </div>
  </div>
</div>

<div class="row">
  <div class="col-sm-12 col-md-4 col-lg-4">
    <div class="form-group">
      <label for="password">Current Password</label>
      <input required type="password" name="current-password" id="current-password" class="form-control" />
    </div>
  </div>
  <div class="col-sm-12 col-md-4 col-lg-4">
    <div class="form-group">
      <label for="password">New Password</label>
      <input required type="password" name="password" id="password" class="form-control" />
    </div>
  </div>
  <div class="col-sm-12 col-md-4 col-lg-4">
    <div class="form-group">
      <label for="con_password">Confirm New Password</label>
      <input required type="password" name="con_password" id="con_password" class="form-control" />
    </div>
  </div>
</div>
<div class="row">
  <div class="col text-center">
    <input type="submit" value="Submit" class="btn btn-info" />
    <input type="reset" value="Clear" class="btn btn-info" />
  </div>
</div>
</form>
</div>
{%- endblock body %}
```

teacher/template/teacher/index.html

```
{% extends 'base.html' %}  
{% block title %}{{request.user.first_name}} {{request.user.last_name}}{% endblock %}  
{% block body %}  
<div class="container mt-3">  
  <h3>{{request.user.first_name}} {{request.user.last_name}}</h3>  
  <div class="details">  
    <div class="row">  
      <div class="col-sm-2">  
        <strong>Class</strong>  
      </div>  
      <div class="col-sm-10">  
        <p>  
          {{student.s_class}}  
        </p>  
      </div>  
    </div>  
    <div class="row">  
      <div class="col-sm-2">  
        <strong>Roll No.</strong>  
      </div>  
      <div class="col-sm-10">  
        <p>  
          {{student.s_roll}}  
        </p>  
      </div>  
    </div>  
  </div>  
  <div class="assignments">  
    <h2 class="text-center">Assignments</h2>  
    {% if assignments %}  
      <div class="table-responsive">  
  
        <table class="table table-hover">  
          <thead>  
            <tr>  
              <th scope="col">#</th>  
              <th scope="col">Upload</th>  
              <th scope="col">Subject</th>  
              <th scope="col">Teacher</th>  
              <th scope="col">Due Date</th>  
              <th>Type</th>  
              <th></th>  
            </tr>  
          </thead>  
          <tbody>  
            {% for assignment in assignments %}  
              <tr>  
                <th scope="row">{{forloop.counter}}</th>  
                {% load humanize %}  
                <td>{{assignment.date | naturaltime}}</td>
```

```

<td>{{assignment.subject.subject_name}}</td>
<td>{{assignment.teacher.first_name}} {{assignment.teacher.last_name}}</td>
<td>{{assignment.last_date}}</td>
<td>{{assignment.ass_type}}</td>
<td><a href="assignment/{{assignment.slug}}" class="btn btn-sm btn-primary">Solve</a></td>
</tr>

{% endfor %}
</tbody>
</table>
</div>
{% else %}
<h4>no assignments</h4>
{% endif %}
</div>
</div>

{% endblock %}

```

teacher/template/teacher/assigment.html

```

{% extends 'base.html'%}
{% block title %}Add Assignment{% endblock %}
{% block body %}
<div class="container mt-3">
<h3 class="text-center">Add Assignment</h3>
<form action="" method="post" accept-charset="utf-8">
{% csrf_token %}
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="class">Class</label>
<select class="form-control" onchange="changeSub()" name="class" id="class">
<option disabled selected>--select--</option>
{% for class in classes %}
<option value="{{class.class_name}}>{{class.class_name}}</option>
{% endfor %}
</select>
</div>
</div>
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="subject">Subject</label>
<select class="form-control" name="subject" id="subject">
<option disabled selected>--select--</option>
</select>
</div>
</div>
</div>
<div class="row">
<div class="col-sm-12 col-md-6 col-lg-6">
<div class="form-group">
<label for="last_date">Last Date</label>
<input class="form-control" type="date" name="last_date" id="last_date" placeholder="Last Date"/>

```

```

        </div>
    </div>
    <div class="col-sm-12 col-md-6 col-lg-6">
        <div class="form-group">
            <label for="type">Assignment Type</label>
            <select name="type" id="type" class="form-control">
                <option disabled selected>--select--</option>
                <option>MCQ</option>
                <option>File Uplopad</option>
            </select>
        </div>
    </div>
    <div class="row">
        <div class="col">
            <div class="form-group text-center">
                <input class="btn btn-primary" type="submit" name="" id="submit" value="Submit" />
            </div>
        </div>
    </div>
</form>
</div>

{% endblock %}
{% block script %}
<script>
    function changeSub(){
        var class_name = document.getElementById("class").value;
        var sub_select = document.getElementById("subject");
        for(i=sub_select.options.length;i>=1;i--){
            sub_select.options[i] = null;
        }
        {% for subject in subjects %}
        if(class_name == "{{subject.class_name}}"){
            var option = new Option="{{subject.subject_name|safe}}","{{subject.subject_name|safe}}";
            sub_select.add(option,undefined);
        }
        {% endfor %}
    }
</script>
{% endblock script %}

```

teacher/templates/techer/assigment_status.html

```
{% extends 'base.html' %}  
{% block title %}  
{{assignment.topic}}-{{assignment.ass_class}}-{{assignment.date}}  
{% endblock title %}  
{% block css %}  
<style>  
.bar-container {  
    position: relative;  
    width: 100%;  
    height: 15px;  
    margin: 0 0;  
}  
  
.bar {  
    position: absolute;  
    top: 0;  
    left: 0;  
    height: 100%;  
    width: 100%;  
}  
  
.progress {  
    position: absolute;  
    display: none;  
    top: 0;  
    left: 0;  
    height: 100%;  
    width: 100%;  
    background: green;  
    animation: progress 1s ease-in-out forwards;  
}  
  
.questions {  
    position: absolute;  
    top: 56px;  
    left: 0;  
    display: none;  
    width: 100%;  
    animation: show 0.5s ease-in-out forwards;  
}  
  
#solved {  
    position: relative;  
    top: 0;  
    left: 0;  
}
```

```

#unsolved {
    position: relative;
    top: 0;
    left: 0;
    display: none;
}

#total {
    position: relative;
    top: 0;
    left: 0;
    display: none;
}

@keyframes show {
    0% {
        transform: translateY(-50px);
        opacity: 0.7;
    }

    100% {
        opacity: 1;
    }
}

@keyframes progress {
    0% {
        transform: scaleX(0);
        transform-origin: left;
    }

    100% {
        transform: scaleX(1);
        transform-origin: left;
    }
}


```

</style>

```

{%- endblock css %}

{%- block body %}



<h2 class="text-center">{{assignment.topic}}</h2>
    <div class="row mt-5">
        <div class="col-lg-6 col-md-6 col-sm-12">
            <h5><strong>Subject:</strong>{{assignment.subject.subject_name}}</h5>
            <h5><strong>Class:</strong>{{assignment.ass_class}}</h5>
            <h5><strong>Date:</strong>{{assignment.date}}</h5>
            <h5><strong>Type:</strong>{{assignment.ass_type}}</h5>
        </div>
        <div class="col-lg-6 col-md-6 col-sm-12">
            <h5><strong>Total Students :</strong>{{count_student}}</h5>
            <h5><strong>Solved Students :</strong>{{solved_student}}</h5>
            <div class="bar-container" style="height: 15px;">
                <div class="bar bg-light" id="bar"></div>
                <div class="progress" id="progress"></div>
            </div>
        </div>
    </div>


```

```

        <span id="show-per"></span>
    </div>
</div>
<div class="filter-button my-3 text-right">
    <button class="btn btn-sm btn-primary" onclick="showTotal()">Total</button>
    <button class="btn-sm btn btn-primary" onclick="showSolved()">Solved</button>
    <button class="btn btn-sm btn-primary" onclick="showUnsolved()">Unsolved</button>
</div>
<div class="students">
    <div id="solved">
        {% if assignment.ass_type == "MCQ" %}
        <div class="table-responsive my-4">
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Student Name</th>
                        <th>Control No</th>
                        <th>Attemp</th>
                        <th>Correct</th>
                        <th>Submit On</th>
                    </tr>
                </thead>
                <tbody>
                    {% for detail in log %}
                    <tr>
                        {% load humanize %}
                        <th>{{forloop.counter}}</th>
                        <td>{{detail.student.first_name}} {{detail.student.last_name}}</td>
                        <td>{{detail.student.username}}</td>
                        <td>{{detail.result.attemp}}</td>
                        <td>{{detail.result.correct}}</td>
                        <td>{{detail.submit_on | naturaltime}}</td>
                    </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
        {% else %}
        <div class="table-responsive">
            <table class="table table-hover">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Student Name</th>
                        <th>Control No</th>
                        <th>Submit On</th>
                        <th colspan="2"></th>
                    </tr>
                </thead>
                <tbody>
                    {% for detail in log %}
                    <tr>
                        <th scope="col">{{forloop.counter}}</th>
                        <td>{{detail.student.first_name}} {{detail.student.last_name}}</td>
                        <td>{{detail.student.username}}</td>
                        <td>{{detail.submit_on | naturaltime}}</td>
                        <td><button onclick="showQue({{detail.student.pk}})">View</button></td>
                        <td><button onclick="closeQue({{detail.student.pk}})">Close</button></td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>

```

```

        class="btn btn-sm btn-danger">Close</button></td>
    </tr>
    {% endfor %}
</tbody>
</table>
</div>
</div>
{% for detail in log %}
<div class="questions" id="que{{detail.student.pk}}">
    <div class="table-responsive p-5 mx-auto bg-light" style="width: 60%;">
        <table class="table table-sm table-hover table-white ">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Questions</th>
                    <th>File</th>
                </tr>
            </thead>
            <tbody>
                {% for question in file %}
                {% if question.student.pk == detail.student.pk %}
                <tr>
                    <th>{{forloop.counter}}</th>
                    <td>{{question.question | safe | slice:"30"}}</td>
                    <td><a href="{{media}}{{question.file}}" class="text-success"><i
                        class="fa fa-2x fa-file"></i></a></td>
                </tr>
                {% endif %}
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>
{% endfor %}
{% endif %}
<div id="unsolved">
    <div class="table-responsive my-4">
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Student Name</th>
                    <th>Roll No.</th>
                    <th>Class</th>
                    <th>Control Number</th>
                </tr>
            </thead>
            <tbody>
                {% for stud in unsolved %}
                <tr>
                    <th>{{forloop.counter}}</th>
                    <td>{{stud.user.first_name}} {{stud.user.last_name}}</td>
                    <td>{{stud.s_roll}}</td>
                    <td>{{stud.s_class.class_name}}</td>
                    <td>{{stud.user.username}}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>

```

```

</div>
<div id="total">
    <div class="table-responsive my-4">
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>#</th>
                    <th>Student Name</th>
                    <th>Roll No.</th>
                    <th>Class</th>
                    <th>Control Number</th>
                </tr>
            </thead>
            <tbody>
                {% for stud in students %}
                <tr>
                    <th>{{forloop.counter}}</th>
                    <td>{{stud.user.first_name}} {{stud.user.last_name}}</td>
                    <td>{{stud.s_roll}}</td>
                    <td>{{stud.s_class.class_name}}</td>
                    <td>{{stud.user.username}}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>
</div>
<% endblock body %>
<% block script %>
<script>
    var total = {{ count_student }};
    var solved = {{ solved_student }};
    var per = solved / total * 100;
    var progress = document.getElementById("progress")
    progress.style.width = per + "%";
    progress.style.display = "initial";
    document.getElementById("show-per").innerHTML = per + "%";
    if (per < 40) {
        progress.style.backgroundColor = "red";
    }
    function showQue(stud_id) {
        // document.getElementsByClassName("questions").style.display = "initial";
        document.getElementById("que" + stud_id).style.display = "initial";
    }
    function closeQue(stud_id){
        document.getElementById("que"+stud_id).style.display = "none";
        document.getElementsByClassName("questions").style.display = "none";
    }
</script>
<script>
    function showSolved() {
        document.getElementById("solved").style.display = "initial";
        document.getElementById("unsolved").style.display = "none";
        document.getElementById("total").style.display = "none";
    }
    function showUnsolved() {
        document.getElementById("unsolved").style.display = "initial";
        document.getElementById("solved").style.display = "none";
    }
</script>

```

```

        document.getElementById("total").style.display = "none";
    }
    function showTotal() {
        document.getElementById("total").style.display = "initial";
        document.getElementById("unsolved").style.display = "none";
        document.getElementById("solved").style.display = "none";
    }
</script>
{% endblock script %}

```

adminimport/adminbase.html

```

<!DOCTYPE html>
<html lang="en">

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>{% block title %}{% endblock %}</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
      integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
      crossorigin="anonymous">

<!-- font awesome link -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<!-- summernote -->
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
      integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
      crossorigin="anonymous"></script>
<link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.js"></script>

<!-- custom stylesheet -->
{% block css %}
{% endblock css %}
<style>
.nav-link {
    color: white;
    font-weight: bold;
    font-size: 1.5rem;
}

.nav-link:hover {
    color: #ffc107;
}

.nav-active a {

```

```

        color: #ffc107;
    }
</style>
</head>

<body>
<header>
<div class="brand-header container-fluid py-2 bg-info">
<div class="row">
<div class="col-6">
<h3 class="text-left text-warning m-0">Admin Portal</h3>
</div>
<div class="col-6 text-right d-flex justify-content-end align-items-start">
<a href="{% url 'index' %}" class="nav-link py-0">View Site</a>
<a href="{% url 'admin_logout' %}" onclick="return confirm('Are you sure want to logouot?');" class="nav-link py-0">Logout</a>
</div>
</div>
</div>
</div>

<nav class="navbar navbar-expand-lg bg-info">
<button class="navbar-toggler ml-auto" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<i class="fa fa-bars nav-link p-0 m-0" style="outline: none;"></i>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mx-auto">
<li class="nav-item active">
<a class="nav-link" href="#students">Students</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#teachers">Teachers</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#classes">Classes</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#subjects">Subjects</a>
</li>
</ul>
</div>
</nav>
</header>
{% for message in messages %}
<div class="alert alert-{{message.tags}} alert-dismissible fade show" role="alert">
{{message}}
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
{% endfor %}
{% block body %}
{% endblock body %}

<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>

```

```

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4lh7YwaYd1iqfktj0Uod8GCExl30g8ifwB6"
crossorigin="anonymous"></script>
{% block script %}
{% endblock script %}

</body>

</html>

```

adminportal/templates/adminportal/index.html

```

{% extends 'adminportal/adminbase.html' %}
{% block title %}Admin Portal{% endblock title %}
{% block css %}
<style>
td {
    min-width: 100px;
}

.search-bar {
    width: 50%;
}

#add-class,
#add-subject,
#add-student,
#add-teacher {
    display: none;
}

#hide-add-class,
#hide-add-subject,
#hide-add-student,
#hide-add-teacher {
    display: none;
}

section {
    padding: 30px 0;
}

@media (max-width: 576px) {
    .search-bar {
        width: 100%;
    }
}
</style>
{% endblock css %}
{% block body %}

```

```

<div class="container-fluid">
    <h2 class="text-center text-warning">Welcome to Admin Portal</h2>
</div>
<section id="students" class="my-5 bg-light">
    <div class="container">
        <div class="head d-flex justify-content-between align-items-center">
            <h3>Students</h3>
            <div class="filter">
                <form action="{% url 'adminportal' %}" method="get">
                    <div class="row">
                        <div class="col-6 py-0 pr-0">
                            <select name="filter-students" class="custom-select custom-select-sm w-100" id="filter-students">
                                <option>All</option>
                                {% for class in classes %}
                                    <option>{{class.class_name}}</option>
                                {% endfor %}
                            </select>
                        </div>
                        <div class="col-6 py-0">
                            <input type="submit" class="btn btn-sm btn-primary" value="Filter">
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
    <form action="{% url 'adminportal' %}" method="get">
        <div class="row my-3">
            <div class="col-lg-11 col-md-11 col-sm-12">
                <input type="text" class="form-control form-control-sm search-bar ml-auto" name="search-student" placeholder="Search Student by Name or Control No">
            </div>
            <div class="col-lg-1 col-md-1 col-sm-12">
                <button type="submit" class="btn btn-sm btn-info mr-auto search-btn"><i class="fa fa-search"></i></button>
            </div>
        </div>
    </form>
    <hr>
    {% if students.count == 0 %}
        <h4 class="text-center">No Data Found</h4>
    {% else %}
        <div class="table-responsive">
            <table class="table table-sm table-hover">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>Roll No.</th>
                        <th>Class</th>
                        <th>Control No.</th>
                        <th>Date Of Birth</th>
                        <th>Contact No.</th>
                        <th>Email</th>
                        <th>Gender</th>
                        <th colspan="2" class="text-center">Operation</th>
                    </tr>
                </thead>
                <tbody>

```

```

{%- for student in students %}

<tr>
    <form action="/adminportal/save_student/{{student.pk}}" method="post">
        {% csrf_token %}
        <th>{{forloop.counter}}</th>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="first-name" value="{{student.user.first_name}}"/></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="last-name" value="{{student.user.last_name}}"/></td>
        <td><input type="text" class="form-control form-control-sm" name="roll-no" value="{{student.s_roll}}"/></td>
        <td style="min-width: 100px;"><select name="class{{student.pk}}" class="custom-select custom-select-sm" id="class{{student.pk}}">
            {% for class in classes %}
                <option {% if student.s_class == class %}selected{% endif %}>{{class.class_name}}</option>
            {% endfor %}
            </select></td>
        <td><input type="text" name="control" class="form-control form-control-sm" value="{{student.user.username}}"/></td>
        <td style="min-width: 200px;"><input type="date" name="dob" class="form-control form-control-sm" value="{{student.s_dob|date:'Y-m-d'}}"/></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="contact" value="{{student.s_contact}}"/></td>
        <td style="min-width: 200px;"><input type="text" class="form-control form-control-sm" name="email" value="{{student.user.email}}"/></td>
        <td><select name="gender{{student.pk}}" class="custom-select custom-select-sm" id="s-gender{{student.pk}}">
            <option {% if student.s_gender|lower == "male" %}selected{% endif %}>Male</option>
            <option {% if student.s_gender|lower == "female" %}selected{% endif %}>Female</option>
        </select></td>
        <td><input type="submit" class="btn btn-sm btn-info" value="Save"/></td>
        <td><a href="/adminportal/delete_student/{{student.pk}}" onclick="return confirm('Are you sure To delete {{student.user.first_name}}?');" class="btn btn-sm btn-danger">Delete</a></td>
    </form>
</tr>
{% endfor %}
<tr id="add-student">
    <form action="{% url 'add_student' %}" method="post">
        {% csrf_token %}
        <th>+</th>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="first-name"/></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="last-name"/></td>
        <td><input type="text" class="form-control form-control-sm" name="roll-no"/></td>
        <td style="min-width: 100px;"><select name="class" class="custom-select custom-select-sm" id="class">
            <option selected disabled>Select</option>
            {% for class in classes %}
                <option>{{class.class_name}}</option>
            {% endfor %}
            </select></td>
        <td><input type="text" name="control" class="form-control form-control-sm"/></td>
        <td style="min-width: 200px;"><input type="date" name="dob" class="form-control form-control-sm" disabled></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="contact" disabled></td>
        <td style="min-width: 200px;"><input type="text" class="form-control form-control-sm" name="email" disabled></td>
    </form>
</tr>

```

```

name="email" disabled></td>
    <td><select name="gender{{student.pk}}" class="custom-select custom-select-sm" id="s-gender{{student.pk}}" disabled>
        <option {% if student.s_gender|lower == "male" %}selected{% endif %}>Male</option>
        <option {% if student.s_gender|lower == "female" %}selected{% endif %}>Female</option>
    </select>
    </td>
    <td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
</form>
</tr>
</tbody>
</table>
</div>
<button class="btn btn-sm btn-info my-2" onclick="addStudent()" style="width: 50px;"><i class="fa fa-plus"></i></button>
<button class="btn btn-sm btn-info my-2" id="hide-add-student" onclick="hideAddStudent()" style="width: 50px;"><i class="fa fa-angle-double-up" style="font-weight:bold;"></i></button>
{%
endif %}
</div>
</section>
<section id="teachers" class="my-5">
    <div class="container">
        <h3>Teachers</h3>
        <form action="{% url 'adminportal' %}" method="get">
            <div class="row my-3">
                <div class="col-lg-11 col-md-11 col-sm-12">
                    <input type="text" class="form-control form-control-sm search-bar ml-auto" name="search-teacher" placeholder="Search Teacher by Name or Control No">
                </div>
                <div class="col-lg-1 col-md-1 col-sm-12">
                    <button type="submit" class="btn btn-sm btn-info mr-auto search-btn"><i class="fa fa-search"></i></button>
                </div>
            </div>
            </div>
            </form>
            <hr>
{%
if teachers.count == 0 %}
<h4 class="text-center">No Data Found</h4>
{%
else %}
<div class="table-responsive">
    <table class="table table-hover table-sm">
        <thead>
            <tr>
                <th>#</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Control No.</th>
                <th>Date Of Birth</th>
                <th>Contact No.</th>
                <th>Email</th>
                <th>Gender</th>
                <th colspan="2" class="text-center">Operation</th>
            </tr>
        </thead>
        <tbody>
{%
for teacher in teachers %}
            <tr>
                <td><form action="/adminportal/save_teacher/{{teacher.pk}}" method="post">
                    {% csrf_token %}
                    <th>{{forloop.counter}}</th>

```

```

<td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="first-name" value="{{teacher.user.first_name}}"></td>
<td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="last-name" value="{{teacher.user.last_name}}"></td>
<td><input type="text" class="form-control form-control-sm" name="control" value="{{teacher.user.username}}"></td>
<td style="min-width: 120px;"><input type="date" class="form-control form-control-sm" name="dob" value="{{teacher.t_dob|date:'Y-m-d'}}"></td>
<td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="contact" value="{{teacher.t_contact}}"></td>
<td style="min-width: 200px;"><input type="text" class="form-control form-control-sm" name="email" value="{{teacher.user.email}}"></td>
<td><select name="gender{{teacher.pk}}" class="custom-select custom-select-sm" id="t-gender{{teacher.pk}}">
    <option {% if teacher.t_gender|lower == "male" %}selected{%endif%}>Male</option>
    <option {% if teacher.t_gender|lower == "female" %}selected{%endif%}>Female</option>
</select></td>
<td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
<td><a href="/adminportal/delete_teacher/{{teacher.pk}}" onclick="return confirm('Are you sure want to delete {{teacher.user.first_name}}?');" class="btn btn-sm btn-danger">Delete</a></td>
</form>
</tr>
{% endfor %}
<tr id="add-teacher">
    <form action="{% url 'add_teacher' %}" method="post">
        {% csrf_token %}
        <th>+</th>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="first-name"></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="last-name"></td>
        <td><input type="text" class="form-control form-control-sm" name="control"></td>
        <td style="min-width: 120px;"><input type="date" class="form-control form-control-sm" name="dob" disabled></td>
        <td style="min-width: 150px;"><input type="text" class="form-control form-control-sm" name="contact" disabled></td>
        <td style="min-width: 200px;"><input type="text" class="form-control form-control-sm" name="email" disabled></td>
        <td><select name="gender" class="custom-select custom-select-sm" id="t-gender" disabled>
            <option>Male</option>
            <option>Female</option>
        </select></td>
        <td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
    </form>
</tr>
</tbody>
</table>
</div>
<button class="btn btn-sm btn-info my-2" onclick="addTeacher()" style="width: 50px;"><i class="fa fa-plus"></i></button>
<button class="btn btn-sm btn-info my-2" id="hide-add-teacher" onclick="hideAddTeacher()" style="width: 50px;"><i class="fa fa-angle-double-up" style="font-weight:bold;"></i></button>
{% endif %}
</div>
</section>
<section id="classes" class="my-5 bg-light">
    <div class="container">
        <h3>Classes</h3>

```

```

<form action="{% url 'adminportal' %}" method="get">
    <div class="row my-3">
        <div class="col-lg-11 col-md-11 col-sm-12">
            <input type="text" class="form-control form-control-sm search-bar ml-auto" name="search-class"
                   placeholder="Search Class by Name">
        </div>
        <div class="col-lg-1 col-md-1 col-sm-12">
            <button type="submit" class="btn btn-sm btn-info mr-auto search-btn"><i
                class="fa fa-search"></i></button>
        </div>
    </div>
</form>
<hr>
{% if classes.count == 0 %}
<h4 class="text-center">No Data Found</h4>
{% else %}
<div class="table-responsive">
    <table class="table table-hover table-sm">
        <thead>
            <tr>
                <th>#</th>
                <th>Class</th>
                <th>Capacity</th>
                <th colspan="2" class="text-center">Operation</th>
            </tr>
        </thead>
        <tbody>
            {% for class in classes %}
            <tr>
                <form action="/adminportal/save_class/{{class.pk}}" method="post">
                    {% csrf_token %}
                    <th>{{forloop.counter}}</th>
                    <td><input type="text" class="form-control form-control-sm" name="class-name"
                               value="{{class.class_name}}"></td>
                    <td><input type="text" class="form-control form-control-sm" name="strength"
                               value="{{class.strength}}"></td>
                    <td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
                    <td><a href="/adminportal/delete_class/{{class.pk}}"
                           onclick="return confirm('Are YOu sure Want to delete {{class.class_name}}?');"
                           class="btn btn-sm btn-danger">Delete</a></td>
                </form>
            </tr>
            {% endfor %}
            <tr id="add-class">
                <form action="/adminportal/add_class" method="post">
                    {% csrf_token %}
                    <th>+</th>
                    <td><input type="text" class="form-control form-control-sm" name="class-name"></td>
                    <td><input type="text" class="form-control form-control-sm" name="strength"></td>
                    <td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
                    <td><a href="/adminportal/delete_class/{{class.pk}}"
                           onclick="return confirm('Are YOu sure Want to delete {{class.class_name}}?');"
                           class="btn btn-sm btn-danger">Delete</a></td>
                </form>
            </tr>
        </tbody>
    </table>
</div>
<button class="btn btn-sm btn-info my-2" onclick="addClass()" style="width: 50px;"><i
        class="fa fa-plus"></i></button>

```

```

<button class="btn btn-sm btn-info my-2" id="hide-add-class" onclick="hideAddClass()" style="width: 50px;"><i class="fa fa-angle-double-up" style="font-weight:bold;"></i></button>
{%
  endif %}
</div>
</section>
<section id="subjects" class="my-5">
  <div class="container">
    <div class="head d-flex justify-content-between align-items-center">
      <h3>Subjects</h3>
      <div class="filter">
        <form action="{% url 'adminportal' %}" method="get">
          <div class="row">
            <div class="col-6 py-0 pr-0">
              <select name="filter-subjects" class="custom-select custom-select-sm w-100" id="filter-subjects">
                <option>All</option>
                {% for class in classes %}
                  <option>{{class.class_name}}</option>
                {% endfor %}
              </select>
            </div>
            <div class="col-6 py-0">
              <input type="submit" class="btn btn-sm btn-primary" value="Filter">
            </div>
          </div>
        </form>
      </div>
    </div>
    <form action="{% url 'adminportal' %}" method="get">
      <div class="row my-3">
        <div class="col-lg-11 col-md-11 col-sm-12">
          <input type="text" class="form-control form-control-sm search-bar ml-auto" name="search-subject" placeholder="Search Subject by Name">
        </div>
        <div class="col-lg-1 col-md-1 col-sm-12">
          <button type="submit" class="btn btn-sm btn-info mr-auto search-btn"><i class="fa fa-search"></i></button>
        </div>
      </div>
    </form>
    <hr>
    {% if subjects.count == 0 %}
      <h4 class="text-center">No Data Found</h4>
    {% else %}
      <div class="table-responsive">
        <table class="table table-hover table-sm">
          <thead>
            <tr>
              <th>#</th>
              <th>Subject</th>
              <th>Class</th>
              <th colspan="2" class="text-center">Operation</th>
            </tr>
          </thead>
          <tbody>
            {% for subject in subjects %}
              <tr>
                <form action="/adminportal/save_subject/{{subject.pk}}" method="post">
                  {% csrf_token %}

```

```

<th>#</th>
<td><input type="text" class="form-control form-control-sm" name="subject-name"
           value="{{subject.subject_name}}"></td>
<td><select name="class{{subject.pk}}" class="custom-select custom-select-sm"
           id="class{{subject.pk}}">
           {% for class in classes %}
             <option {% if subject.class_name == class %}selected{% endif %}>{{class.class_name}}
             </option>
           {% endfor %}
         </select></td>
<td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
<td><a href="/adminportal/delete_subject/{{subject.pk}}"
           onclick="return confirm('Are you sure want to delete {{subject.subject_name}}?');"
           class="btn btn-sm btn-danger">Delete</a></td>
</form>
</tr>
{% endfor %}
<tr id="add-subject">
<form action="/adminportal/add_subject" method="post">
  {% csrf_token %}
  <th>+</th>
  <td><input type="text" class="form-control form-control-sm" name="subject-name"></td>
  <td><select name="class" class="custom-select custom-select-sm" id="class">
    {% for class in classes %}
      <option>{{class.class_name}}</option>
    {% endfor %}
  </select></td>
  <td><input type="submit" class="btn btn-sm btn-info" value="Save"></td>
  <td><a href="/adminportal/delete_class/{{class.pk}}"
           onclick="return confirm('Are YOu sure Want to delete {{class.class_name}}?');"
           class="btn btn-sm btn-danger">Delete</a></td>
  </form>
  </tr>
</tbody>
</table>
</div>
<button class="btn btn-sm btn-info my-2" onclick="addSubject()" style="width: 50px;"><i
           class="fa fa-plus"></i></button>
<button class="btn btn-sm btn-info my-2" id="hide-add-subject" onclick="hideAddSubject()"
           style="width: 50px;"><i class="fa fa-angle-double-up" style="font-weight:bold;"></i></button>
  {% endif %}
</div>
</section>
{% endblock body %}
{% block script %}
<script>
  function addStudent(){
    document.getElementById('add-student').style.display = "table-row";
    document.getElementById('hide-add-student').style.display = "initial";
  }

  function addTeacher(){
    document.getElementById('add-teacher').style.display = "table-row";
    document.getElementById('hide-add-teacher').style.display = "initial";
  }

  function addClass() {
    document.getElementById('add-class').style.display = "table-row";
    document.getElementById('hide-add-class').style.display = "initial";
  }
</script>

```

```
}

function addSubject() {
    document.getElementById('add-subject').style.display = "table-row";
    document.getElementById('hide-add-subject').style.display = "initial";
}

function hideAddStudent(){
    document.getElementById('add-student').style.display = "none";
    document.getElementById('hide-add-student').style.display = "none";
}

function hideAddTeacher(){
    document.getElementById('add-teacher').style.display = "none";
    document.getElementById('hide-add-teacher').style.display = "none";
}

function hideAddClass() {
    document.getElementById('add-class').style.display = "none";
    document.getElementById('hide-add-class').style.display = "none";
}

function hideAddSubject() {
    document.getElementById('add-subject').style.display = "none";
    document.getElementById('hide-add-subject').style.display = "none";
}
</script>
{% endblock script %}
```

HOSTING ON APACHE2 SERVER IN UBUNTU

STEP 1 : Install Apache2 server and required library in ubuntu.

```
$ sudo apt update
```

```
$ sudo apt install apache2 libapache2-mod-wsgi-py3
```

STEP 2 : Install virtualenv package of python.

```
$ pip3 install virtualenv
```

STEP 3 : Create virtual environment.

```
$ virtualenv projectenv
```

STEP 4 : Start virtual environment.

```
$ source projectenv/bin/activate
```

STEP 5 : Install django in virtual environment.

```
$ pip3 install django
```

STEP 6 : To run project on django development server.

```
$ python manage.py runserver 0.0.0.0:8000
```

STEP 7 : Deactivate virtual environment.

```
$ deactivate
```

STEP 8 : Give all Permission to Apache2 for project directory.

```
$ sudo chmod -R 777 ocs/
```

STEP 9 : Copy admin static files to project static folder

```
$ sudo cp -r /home /devil /web /projectenv /lib /python3.8 /site-packages /django /contrib /admin /static /admin /home /devil /web
```

/ocs /static

STEP 10 : Edit /etc/apache2/sites-available/000-default.conf

```
$ sudo vi 000-default.conf
```

The screenshot shows a dual-terminal session on a Linux desktop. The top terminal window is titled "devil@devil-Inspiron-3542: /etc/apache2/sites-available" and displays the Apache configuration file `000-default.conf`. The bottom terminal window is also titled "devil@devil-Inspiron-3542: /etc/apache2/sites-available" and shows the command `$ sudo vi 000-default.conf` being typed. The Vim editor is open in the top terminal, displaying the configuration file's contents. The configuration file includes directives for virtual hosts, log levels, and specific application paths like `/home/devil/web/ocs`.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /home/devil/web/

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    Alias /static /home/devil/web/ocs/static
    <Directory /home/devil/web/ocs/static>
        Require all granted
    </Directory>

    <Directory /home/devil/web/ocs>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>

    WSGIDaemonProcess ocs python-home=/home/devil/web/assignment_system python-path=/home/devil/web/ocs
    WSGIProcessGroup ocs
    WSGIScriptAlias / /home/devil/web/ocs/ocs/wsgi.py

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
```

STEP 11 : Desable and enable 000-default.conf

```
$ sudo a2dissite 000-default.conf
```

```
$ sudo a2ensite 000-default.conf
```

STEP 12 : Restart Apache2 service.

```
$ sudo service apache2 restart
```

6)Advantages

- Easy to use web based system.
- Users can register online.
- Notes are accessible from anyone to anywhere .
- Centralized control by administrator.
- Managed notes in a secure manner.

7)Future Scope

- This Application is Future Scopes is more advanceable or Secure System which is used to only authentications Place like School, College, etc.
- This Application Not Be accessible to any one behind there School or College Class Student and faculty. I plan to this project developed like is used by Three types of users
 - Students.
 - Teachers
 - Administrators
- Students and Teachers can create their accounts and start viewing notes shared by other users as well as upload their own notes.
- Students and Teachers New features can be added to the system as per requirements

8)Conclusion

Is Conclude that this web application is same as the free open source app. means that anyone can accessible , modifiable, re uploadable without any permission so that this web application is more useful not only authenticated people else any user also.

9) References

Tutorials:-

- 1 <https://www.geeksforgeeks.org/django-tutorial>
- 2 <https://www.tutorialspoint.com/django/index.htm>
- 3 www.python.org
- 4 www.w3schools.com
- 5 www.youtube.com