

Online Examination System

Project Documentation

Debayan Ghosh (536)

Subham Jha (521)

Arkaprabha Ghosh (548)

Under the Supervision of Prof. Kaushik Goswami

Content

Introduction -----	5
Brief overview-----	6
Objective & Scope -----	6
Basic model & Features -----	7
System Specifications -----	8
Software & Programming/Scripting languages used -----	9
System requirements -----	9
Design analysis -----	10
Introduction -----	11
Data flow diagrams (DFD) -----	11
Introduction -----	11
0-LEVEL DFD -----	14
1-LEVEL DFD -----	15
2-LEVEL DFD -----	16
Database and Entity-Relationship (ER) model -----	17
Introduction -----	17
Database ERD-----	19
Foreign key and many to many relationships of tables-----	20
Database Schema Description -----	21
Module Descriptions and Screenshots -----	25
Homepage (login.html)-----	26
Teacher -----	27
Teacher Dashboard (teacher_dashboard.html) -----	27
Create a new question paper or edit previously set question papers (papers_and_questions.html) -----	28
Add / Edit questions (view_paper_questions.html) -----	29
Form to add/edit question and question paper details (create.html) -----	34
For the teacher to view student marks (marks_for_teacher.html) -----	36
For the teacher to unlock paper and publish marks of students (unlock_papers_and_marks.html) -----	37
Student -----	38
Student Dashboard (student_dashboard.html) -----	38
Student's past exam results (marks_for_student.html) -----	41
Examination page (ajax_exam.html) -----	42
Change Password (change_password_html) -----	44
The code -----	45
Homepage (login.html)-----	46
Teacher -----	52

Teacher Dashboard (teacher_dashboard.html) -----	53
Create a new question paper or edit previously set question papers (papers_and_questions.html) -----	63
Add / Edit questions (view_paper_questions.html) -----	70
Form to add/edit question and question paper details (create.html) -----	83
For the teacher to view student marks (marks_for_teacher.html) -----	90
For the teacher to unlock paper and publish marks of students (unlock_papers_and_marks.html) -----	100
Student-----	108
Student Dashboard (student_dashboard.html) -----	109
Student's past exam results (marks_for_student.html) -----	117
Examination page (ajax_exam.html) -----	127
Change Password (<i>change_password.html</i>)-----	142
Backend-----	150
Manages the view of Model View Template (MVT) Architecture (Views.py) -----	151
Definition of model classes (models.py)-----	170
Definition of Django forms (forms.py)-----	173
Controls the user permission and privileges (decorators.py)-----	176
Adds the models in Django Admin panel (admin.py) -----	178
Manages all URLs of the exam application (urls.py) -----	179
Manages configuration/settings of the project (Main project (settings.py)) -----	181
To redirect every request to the exam application (Main project (urls.py)) -----	186
Testing -----	187
 Introduction -----	188
 Test Cases: -----	191
 Test Report-----	195
Security and Special Features -----	197
 Secure Browsing-----	198
Introduction -----	198
Need for Secure Browser in Online Exam -----	198
How Secure Browser for Online Exams work? -----	198
 Random Question Selector -----	200
 Autosave Feature -----	201
Future Scope and Improvements -----	202
 Remote Proctoring-----	203
 Question Pattern -----	204
 Notice Board-----	204
Bibliography-----	205

Introduction

Brief overview

Online Examination System is a technology-driven way to simplify examination activities like defining exam patterns with question banks, defining exam timer, objective question sections, conducting exams using the computer or mobile devices in a paperless manner with proper security features and monitoring technologies to ensure the best possible remote invigilation of an examination.

It is a cost-effective way to change from the traditional method of holding exams in pen and paper format to a paperless mode.

The system has been designed for a candidate to give exams from any mobile, laptop or desktop which supports the latest browsers.

It has made the assessment method entirely automated, comprehensive, accurate and instant, thereby bringing ease and comfort to the professors. It eliminates the dependency on any sort of manual workload which can at times be too much in case of an online test.

The online examination system basically comprises of two parts, **Student and Teacher**: -

- The student part deals with a student logging in and giving the examination.
- The teacher part deals with the teacher setting up the question paper for the student to answer, unlocking the papers for the student to start the paper and publishing the marks of the student.

Objective & Scope

Owing to the Covid-19 pandemic, and the imposition of the lockdown a shift has been made to digital media as never seen before. One of the prime examples of the change is of our education system, in both schools and colleges.

While classes have been for the most part being conducted successfully through various video conferencing applications, the problem lies in taking **examinations**. To conduct proper examinations the schools and colleges have basically adopted two ways. One is designing their own system and another one is using emails for the exchange of question papers. Now the later method especially does not guarantee the integrity of an examination hall and rampant cheating has been seen to be prevalent.

We, the students of St Xavier's College have hence decided to come up with an examination system for a smooth and efficient conduction of any examination, eliminating malpractices as far as possible. The system is designed to work for objective based MCQ examinations.

Basic model & Features

The online examination system basically consists of two main sections: -

- i) Teacher
- ii) Student

The teacher can get into his/her account, **create a new question paper, edit an existing question paper, remove a question paper, view marks of the student who have attended the papers, publish marks for the students to see and unlock the paper for the students to begin their examination.**

The student can get into his/her account and **can attend the question paper in a stipulated time. He/she can also see the papers which he/she has attended or missed and can also see the marks in his/her previously attended papers.**

Some of the many special **features** are as follows: -

- 1) A fixed no. of questions from a set of questions, set by the teacher will be **randomly generated** for different students.
- 2) Repeated **swapping to other applications, tabs in the same browser or exiting the full screen** at the time of examination will result in **disqualification**.
- 3) Question papers can be edited by the teachers for **unlimited no of times till the exam has started**.
- 4) A thoroughly **synchronized system with automated timer synchronization with server and continuity for system failures**

System Specifications

Software & Programming/Scripting languages used

- ✓ Google Chrome
- ✓ Django – 3.1.5
- ✓ Python - 3.7.4 (x64)
- ✓ PostgreSQL
- ✓ Bootstrap – 4.0.0 & CSS
- ✓ JavaScript
- ✓ HTML

System requirements

Minimum Hardware requirements:

Processor	x86 64-bit CPU (Intel / AMD architecture)
RAM	2GB
Hard disk	5 GB.

Software requirements

Web Browsers	Chrome, Firefox, Internet Explorer, Microsoft Edge, Opera, Safari, Android Browser and WebView (v5.0+) (Use the latest version of the browsers)
Operating Systems	Windows, Mac.

Design analysis

Introduction

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

Data flow diagrams (DFD)

Introduction

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

It is usually beginning with a context diagram as level 0 of the DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1

diagram with lower-level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to levels 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of detail for decomposing a particular function depending on the complexity that function.

DFD diagram notations:

- **External Entity:**

An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people used to draw external entities on the edge of a diagram.



- **Process:**

A process is a business activity or function where the manipulation and transformation of data take place. A process can be decomposed to a finer level of details, for representing how data is being processed within the process.



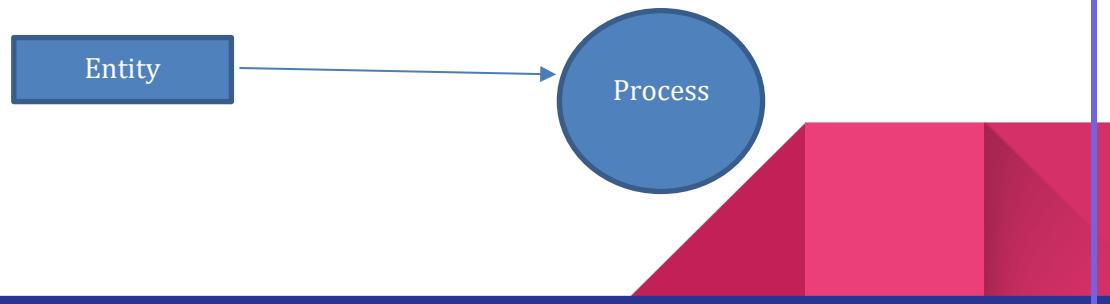
- **Datastore:**

A data store represents the storage of persistent data required and/or produced by the process. Here are some examples of data stores: membership forms, database tables, etc.



- **Dataflow:**

A data flow represents the flow of information, with its direction represented by an arrowhead that shows at the end(s) of flow connector.



Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance (like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes upto 9 and minimum 3 processes.

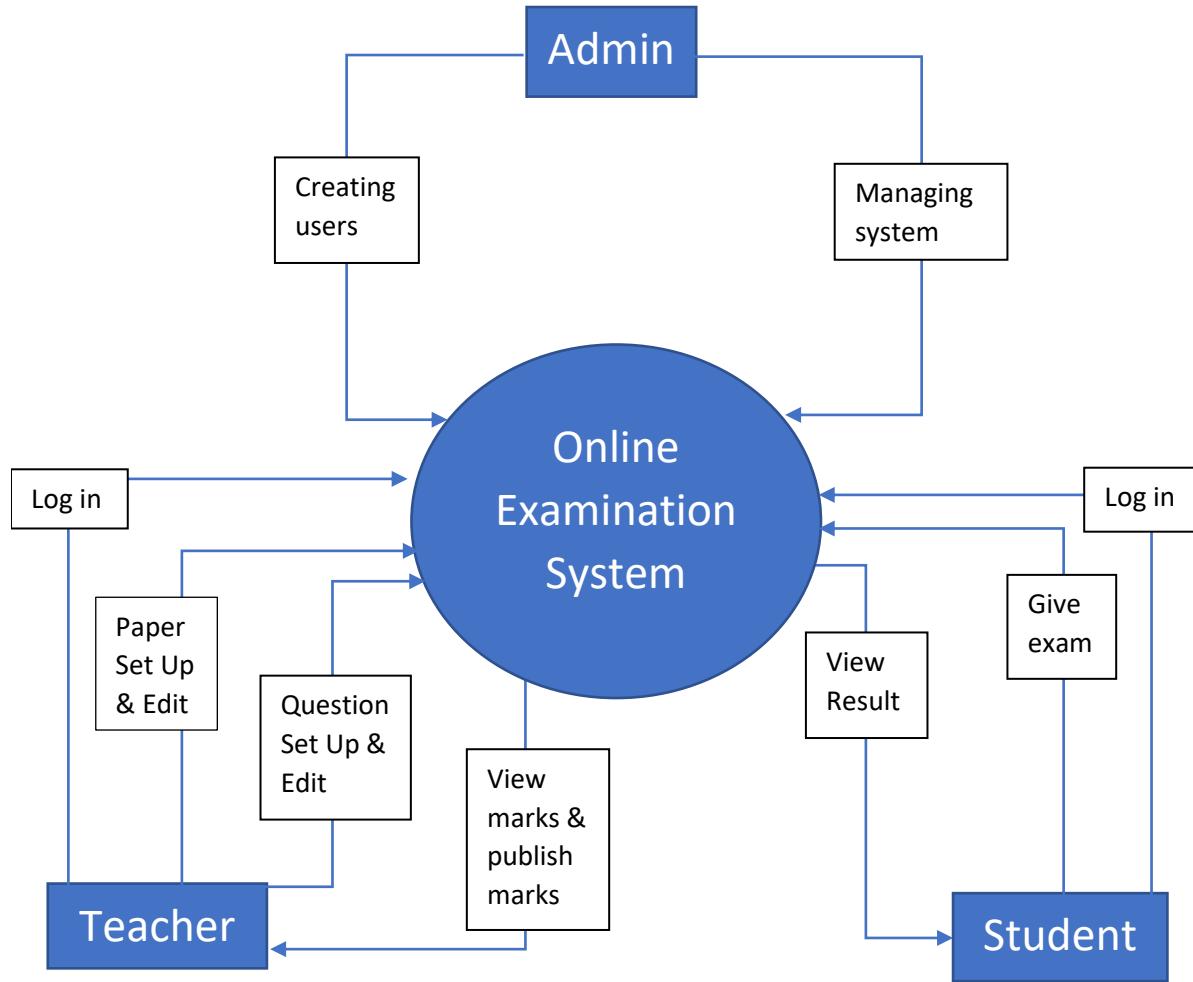
Levels of DFD:

DFD uses hierarchy to maintain transparency thus multilevel DFD's can be created. Levels of DFD are as follows:

- 0-level DFD
- 1-level DFD:
- 2-level DFD:

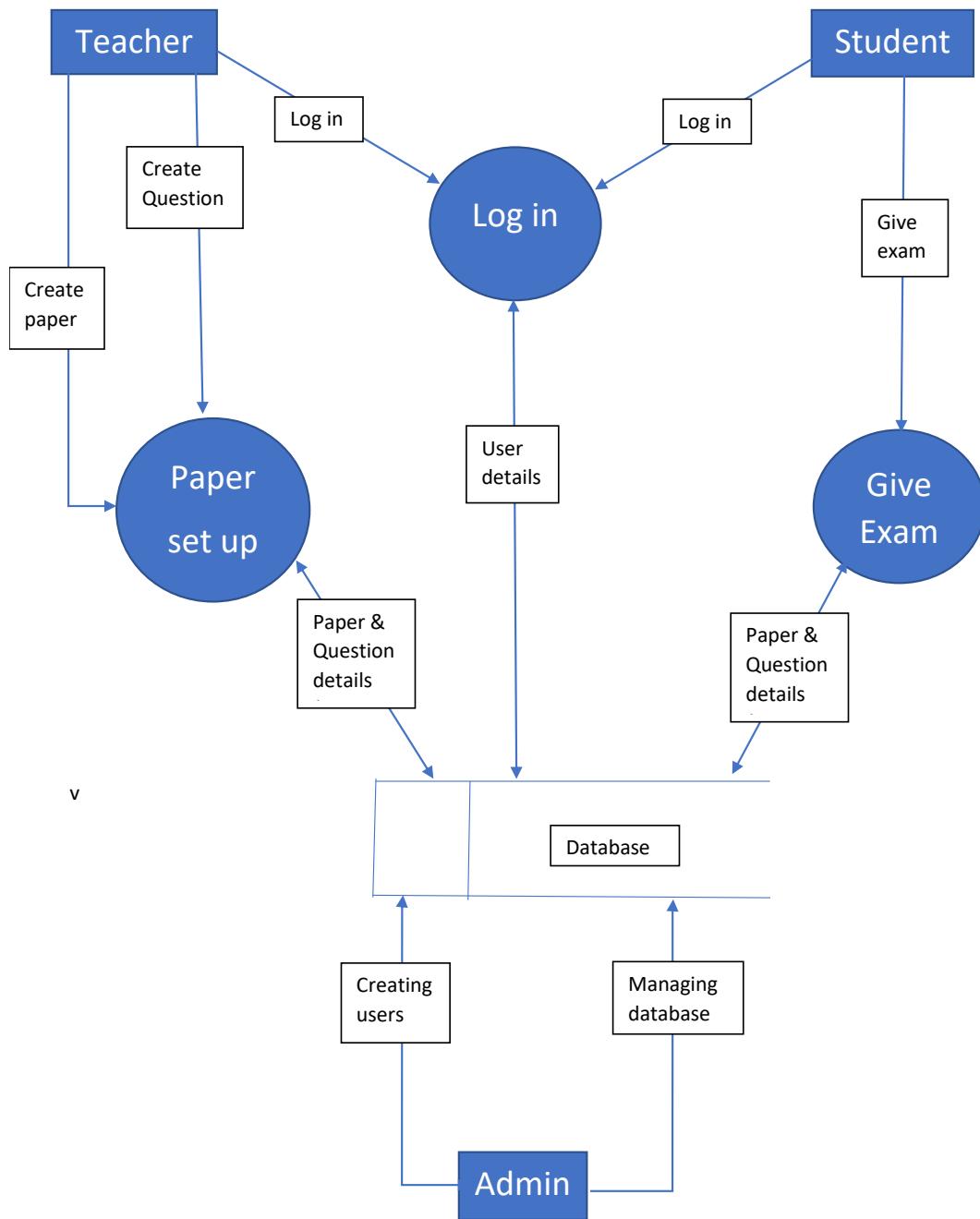
0-LEVEL DFD

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. For our examination system it is as follows: -



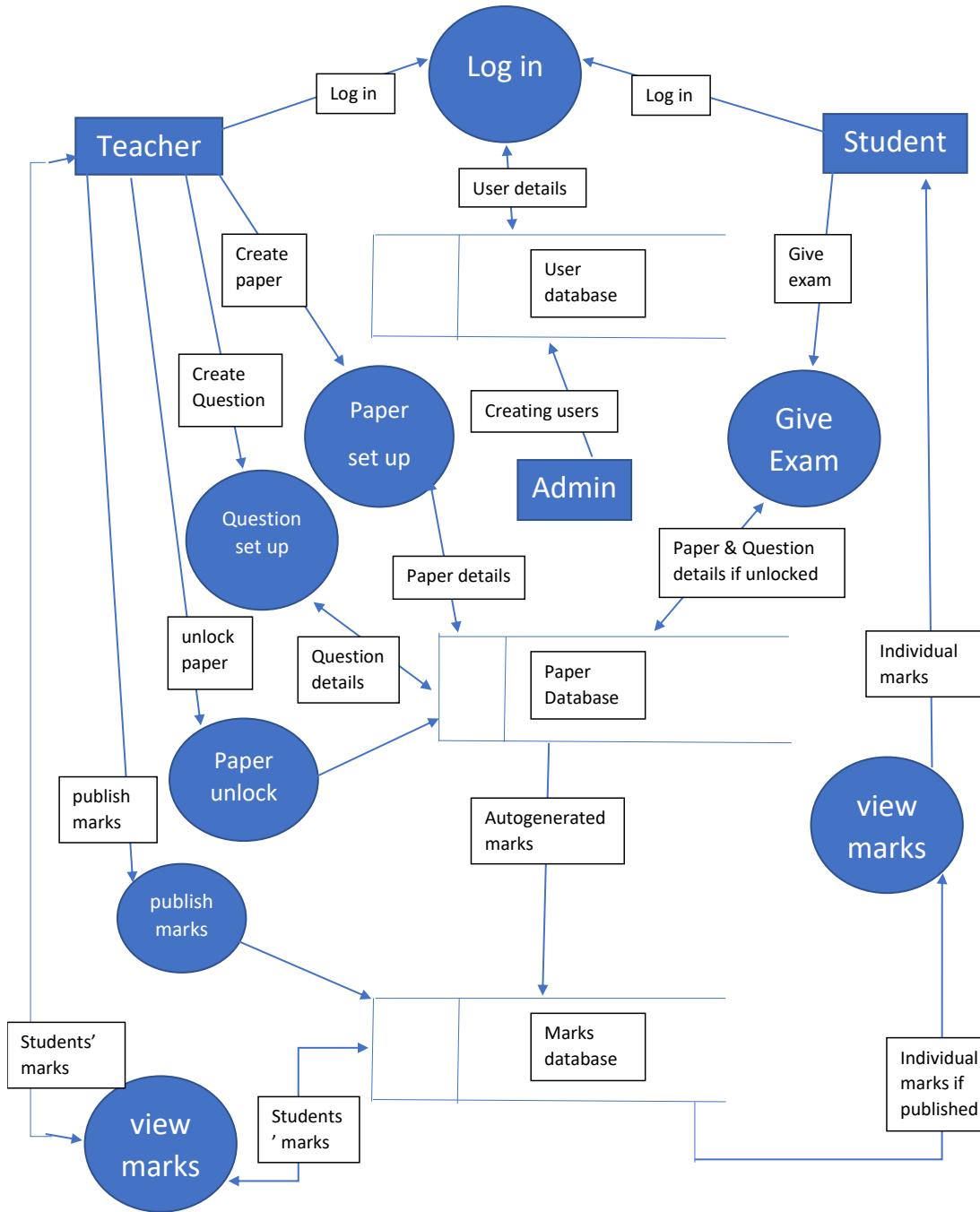
1-LEVEL DFD

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses. For our examination system it is as follows: -



2-LEVEL DFD

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning. For our examination system it is as follows: -



Database and Entity-Relationship (ER) model

Introduction

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represents data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design. For the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Connectivity and Cardinality

The basic types of connectivity for relations are one-to-one, one-to-many, and many-to-many. A one-to one (1:1) relationship is when at most one instance of a entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A one-to-many (1: N) relationships is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1: N relationships is

- a department has many employees
- each employee is assigned to one department

A many-to-many (M: N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A.

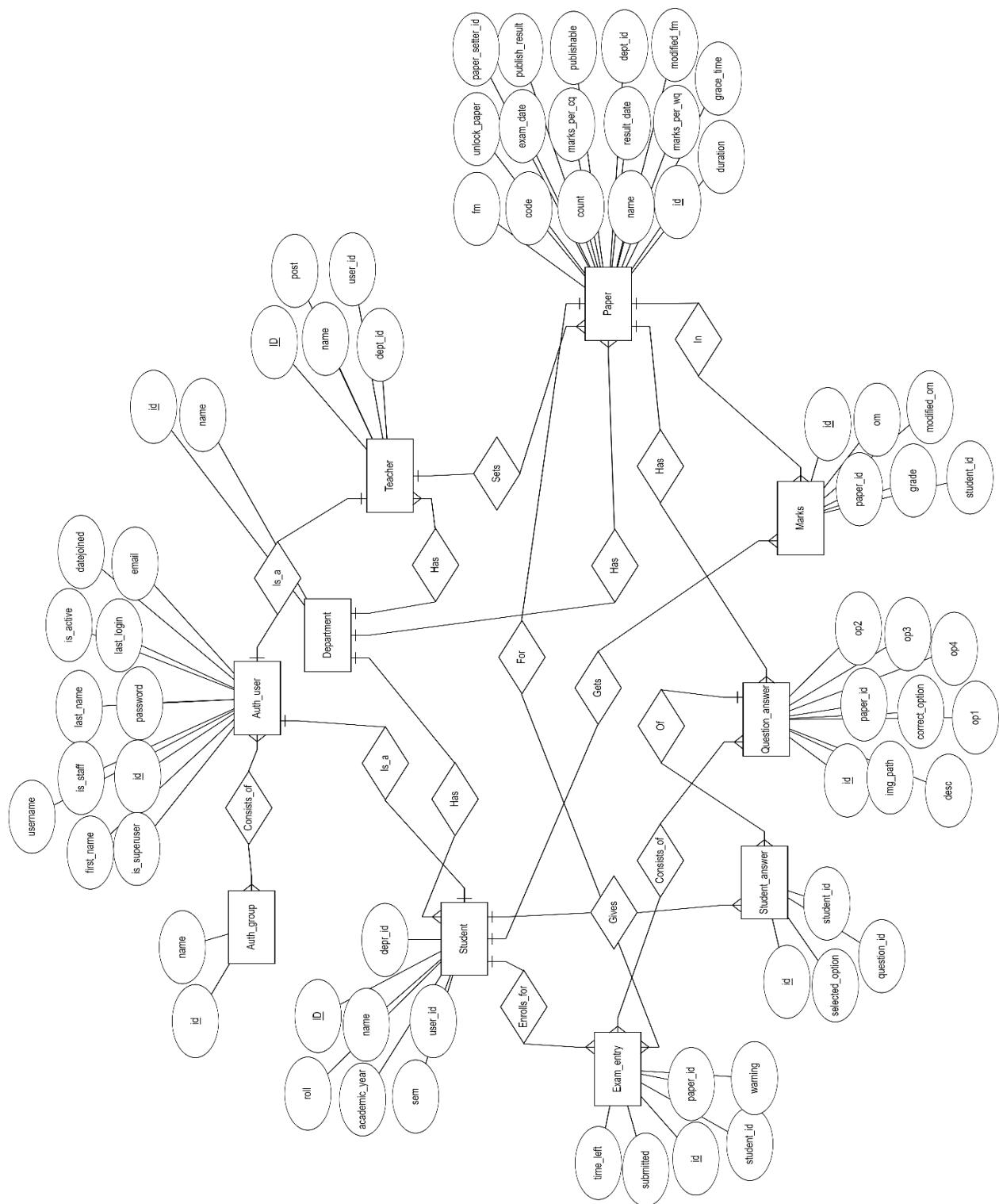
ER-Notation

There is no standard for representing data objects in a database ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

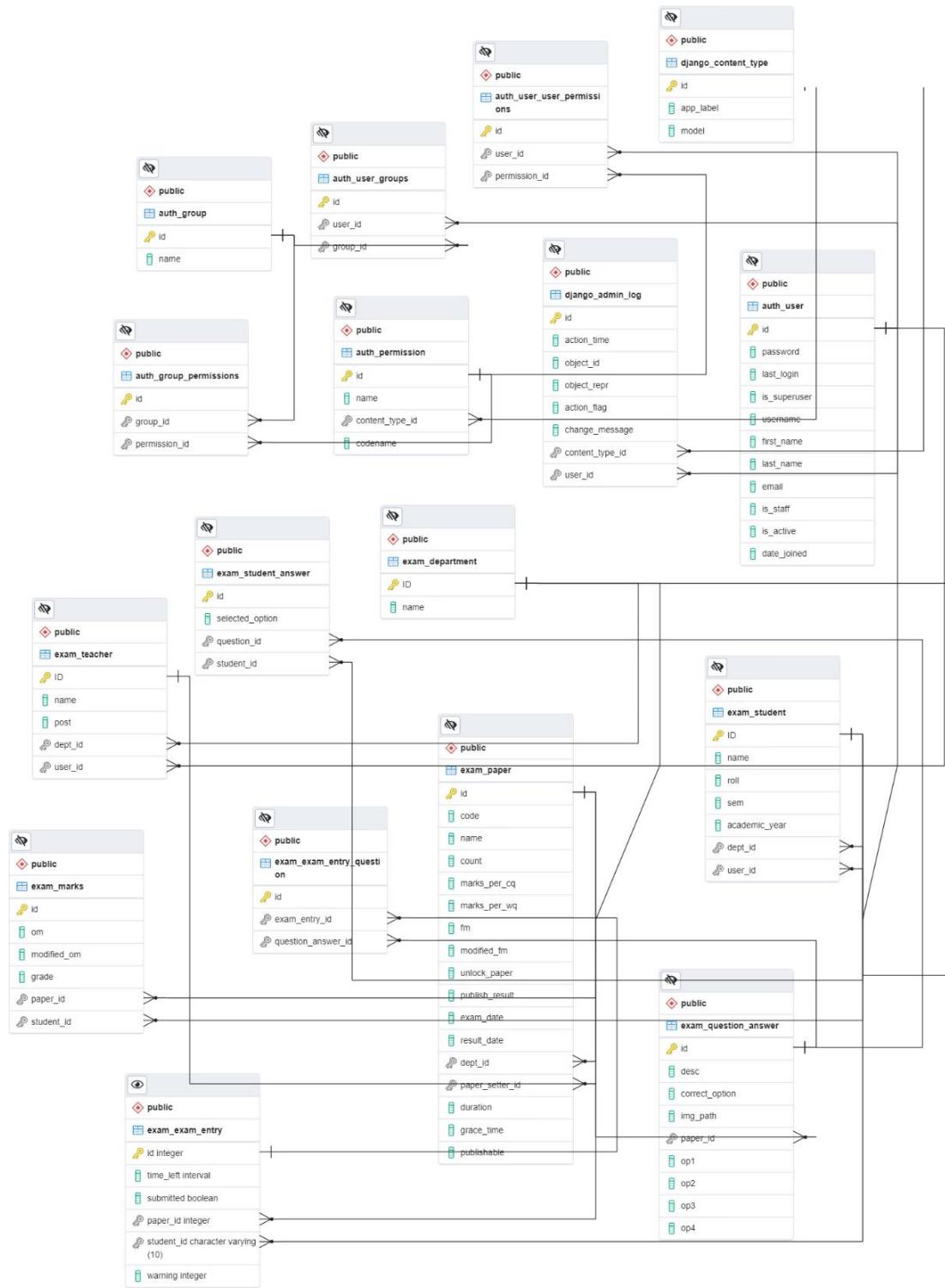
All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **Relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- **Cardinality of many** is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- **Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

Database ERD



Foreign key and many to many relationships of tables



Database Schema Description

Column Name	Datatype	Is nullable
Teacher		
ID	character varying	NO
name	character varying	NO
post	character varying	NO
dept_id	character varying	YES
user_id	integer	YES
Student		
ID	character varying	NO
name	character varying	NO
roll	integer	NO
sem	integer	NO
academic_year	integer	NO
dept_id	character varying	NO
user_id	integer	YES
Paper		
id	integer	NO
code	character varying	NO
name	character varying	NO
count	integer	NO
marks_per_cq	integer	NO
marks_per_wq	double precision	NO
fm	integer	NO
modified_fm	double precision	NO

unlock_paper	boolean	NO
publish_result	boolean	NO
exam_date	timestamp with time zone	YES
result_date	timestamp with time zone	YES
dept_id	character varying	NO
paper_setter_id	character varying	YES
duration	interval	YES
grace_time	interval	YES
publishable	boolean	NO
Question_answer		
id	integer	NO
desc	text	NO
correct_option	character varying	NO
img_path	character varying	YES
paper_id	integer	NO
op1	character varying	NO
op2	character varying	NO
op3	character varying	NO
op4	character varying	NO
Student_answer		
id	integer	NO
selected_option	character varying	NO
question_id	integer	YES
student_id	character varying	NO
Marks		

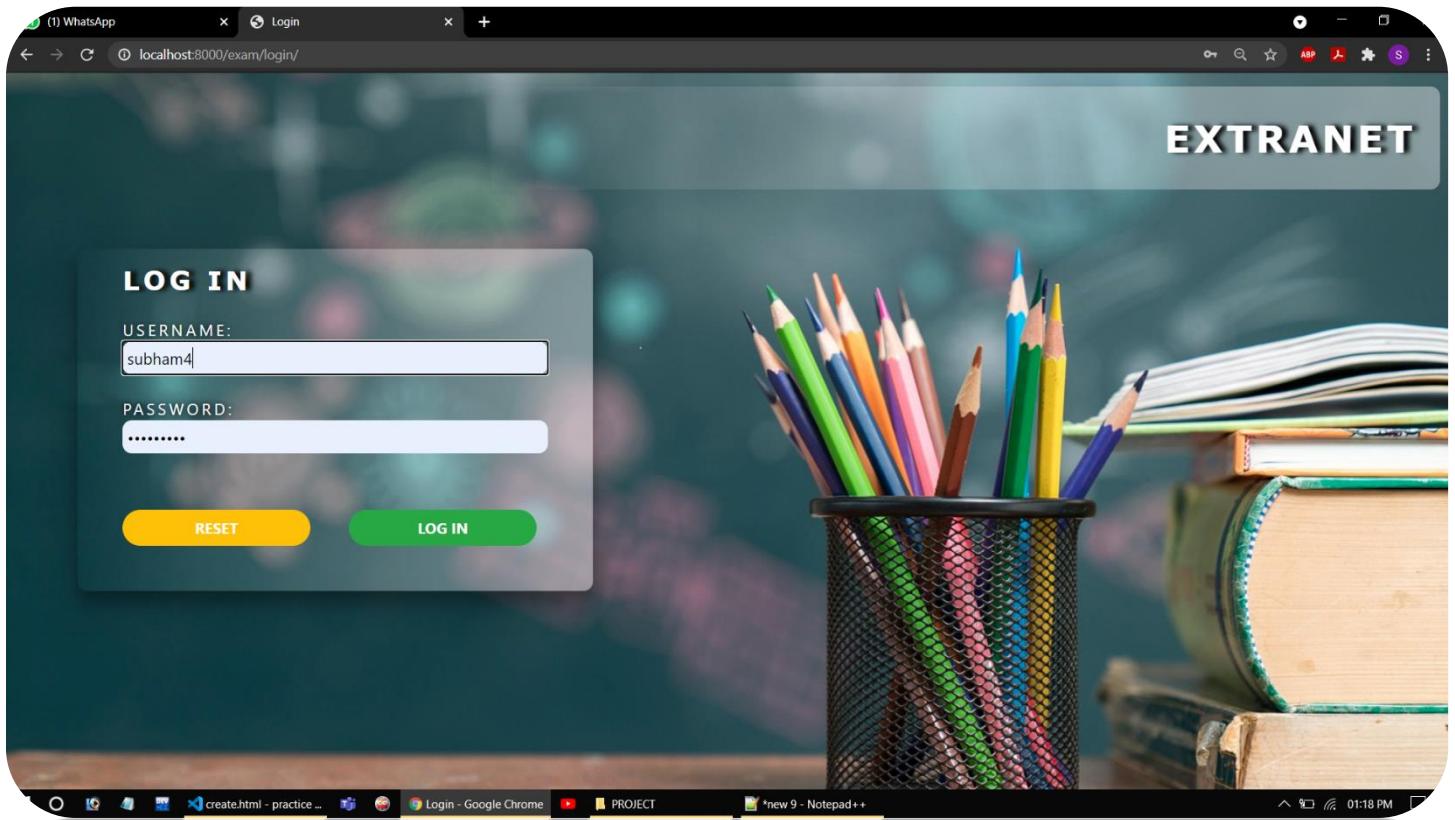
id	integer	NO
om	double precision	NO
modified_om	double precision	NO
grade	character varying	NO
paper_id	integer	NO
student_id	character varying	NO
Exam_entry		
id	integer	NO
time_left	interval	YES
submitted	boolean	NO
paper_id	integer	NO
student_id	character varying	NO
warning	integer	NO
Exam_entry_question		
id	integer	NO
exam_entry_id	integer	NO
question_answer_id	integer	NO
Department		
ID	character varying	NO
name	character varying	NO
Auth_user		
id	integer	NO
password	character varying	NO
last_login	timestamp with time zone	YES
is_superuser	boolean	NO

username	character varying	NO
first_name	character varying	NO
last_name	character varying	NO
email	character varying	NO
is_staff	boolean	NO
is_active	boolean	NO
date_joined	timestamp with time zone	NO
Auth_group		
id	integer	NO
name	character varying	NO
Auth_user_groups		
id	integer	NO
user_id	integer	NO
group_id	integer	NO

Module Descriptions and Screenshots

Homepage (login.html)

- ❖ Teacher and students can login from here.
- ❖ If wrong credentials are entered, they will be provided with appropriate message.



Teacher

Teacher Dashboard (teacher_dashboard.html)

- ❖ Homepage for teacher.
- ❖ Teachers can go to any page relevant to them from here.
- ❖ Shows a welcome note.
 - Shows the option to change password.
 - Shows the option to logout.
 - Shows 3 main options i.e.
 - Question Paper related operation
 - Student marks of past exams
 - Unlock marks and publish result of past exams.

Extranet
Home
Question papers
Student marks
Unlock paper / Publish Marks




Welcome, Mr. ACN

**Question
Papers**

**Click on this to set a
question paper.**

**Student
Marks**

**Click on this to check the
marks of the student.**

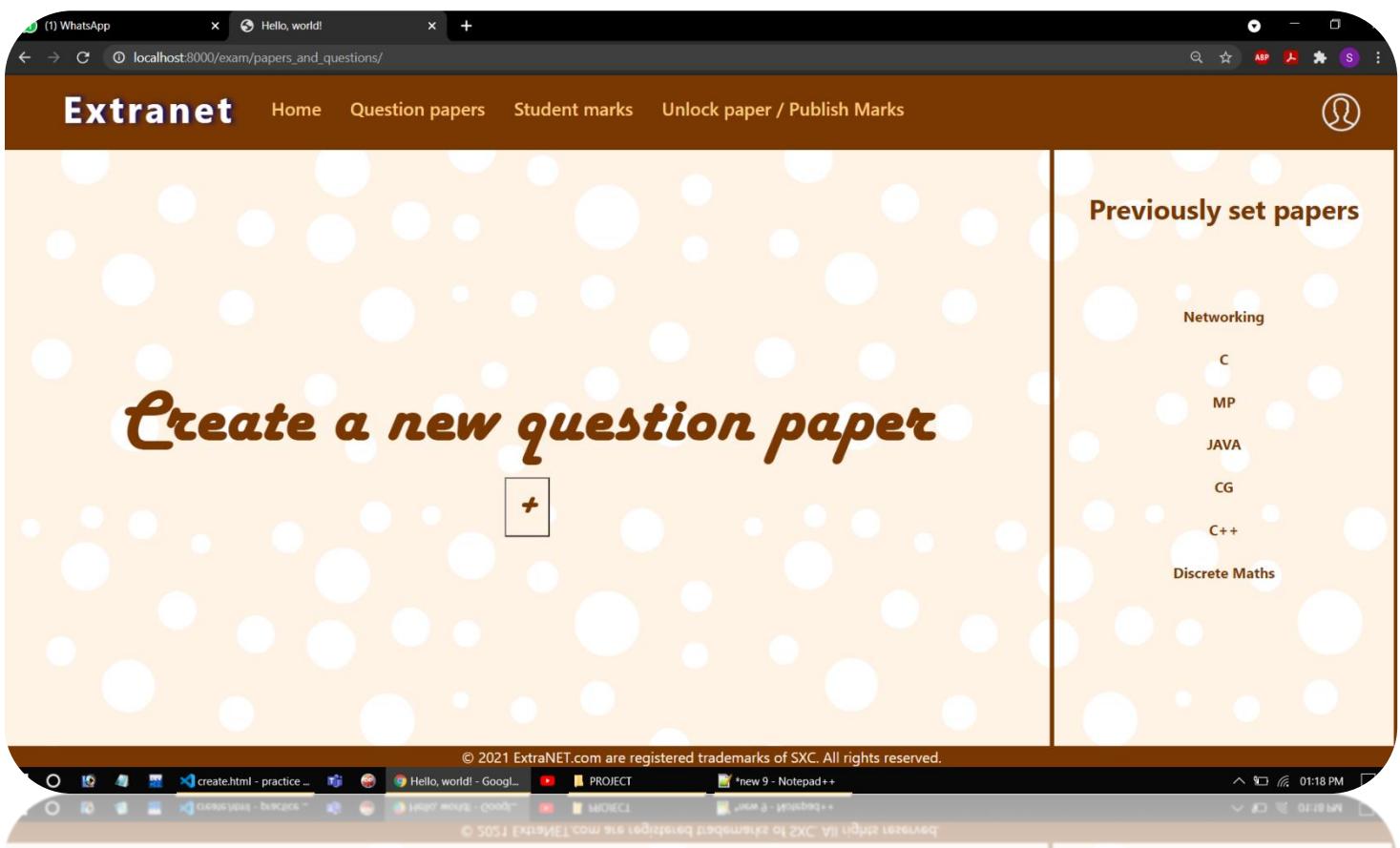
**Unlock
Paper/Publish
Marks**

**Click on this to check to
unlock the paper or
publish the marks of the
examination.**

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
© 2021 ExtraNET.com Web Content developed by SXC. All rights reserved.

Create a new question paper or edit previously set question papers (papers_and_questions.html)

- ❖ This page has all the functionalities related to papers and question-answer setting i.e.
 - Create paper: To create a new Question paper.
 - List of previously set/ created papers with their respective links.



Add / Edit questions (view_paper_questions.html)

- ❖ Shows the respective paper details like,
 - code,
 - name,
 - number of questions,
 - marks per correct and wrong question,
 - full marks,
 - modified full marks,
 - department name,
 - paper setter name.
- ❖ Shows options to update the paper details.
- ❖ Shows options to delete the entire paper.
- ❖ Shows the previously set questions along with options to view, edit, delete them.
- ❖ Shows option to add new questions.

The screenshot shows a web application interface titled "Extranet". The main navigation bar includes links for Home, Question papers, Student marks, and Unlock paper / Publish Marks. On the left, there's a sidebar titled "Questions" containing a list of ten questions. Each question has a "View" button and a "Delete" button. The right side of the screen displays "Paper Details" with the following information:

Paper Details

Code : P10
Name : C
Number of Questions : 5
Marks per correct question : 4
Marks per wrong question : -1.0
Full Marks : 20
Modified Full Marks : 10.0
Department : Computer Science
Paper Setter : Mr. AN

At the bottom of the page, there are "Update" and "Delete" buttons. The footer contains a copyright notice: "© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved." The taskbar at the bottom of the browser window shows several open tabs and windows, including "create.html - practice ...", "Document - Google C...", "PROJECT", and "new 9 - Notepad++".

(1) WhatsApp X Document X +

localhost:8000/exam/view_paper_questions/27/

Extranet

Home Question paper

Question

Desc : Comment about the code snippet given below.

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello World");
5     return 0;
6 }
```

Correct Option : b

Paper Details

Code : P10
Name : C
Number of Questions : 5
Marks per correct question : 4
Marks per wrong question : -1.0
Full Marks : 20
Modified Full Marks : 10.0
Department : Computer Science
Paper Setter : Mr. AN

Questions

- C is a
- Which one is not a keyword in C?
- Comment about the code snippet given below.
- Minimum possible integer in a 16 bit C compiler for int is
- Predict the output of following program.
- Predict the output of following program
- Predict the output of following program
- Double is used to

Delete **Delete** **Delete** **Delete** **Delete** **Delete** **Delete** **Delete** **Delete** **Delete**

create.html - practice ... Document - Google C... PROJECT *new 9 - Notepad++

01:17 PM

COLLECTOR

TITLE SUMMARY

C

JOIN ON

(1) WhatsApp Document

localhost:8000/exam/view_paper_questions/27/

Extranet

[Home](#) [Question papers](#) [Student marks](#) [Unlock paper / Publish Marks](#)

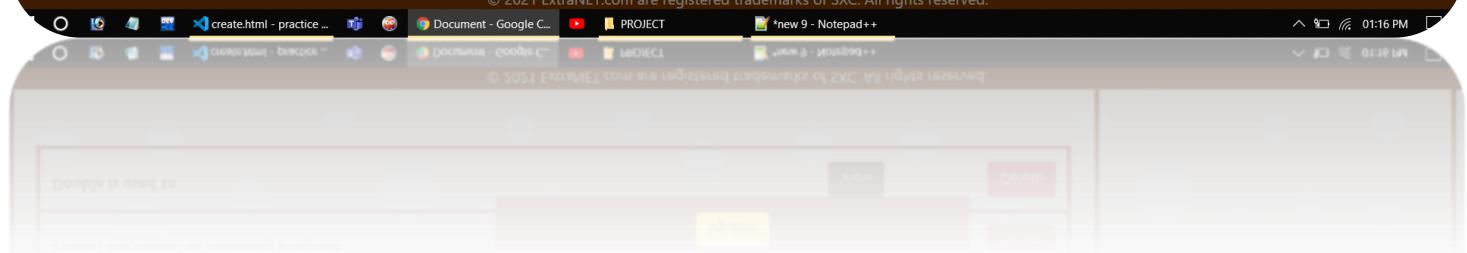
Questions

Question	x
C is a	<button>Delete</button>
Which one is not a keyword in C?	<button>Delete</button>
Comment about the statement given below.	<button>Delete</button>
Minimum possible integer in a 16 bit C compiler for int is	<button>Delete</button>
Predict the output of following program.	<button>Delete</button>
Predict the output of following program	<button>Delete</button>
Predict the output of following program	<button>Update</button>
Double is used to	<button>View</button> <button>Delete</button>

Code : P10
Name : C
Number of Questions : 5
Marks per correct question : 4
Marks per wrong question : -1.0
Full Marks : 20
Modified Full Marks : 10.0
Department : Computer Science
Paper Setter : Mr. AN

Update

Delete



The screenshot shows a web-based application titled "Extranet". The main menu includes "Home", "Question papers", "Student marks", and "Unlock paper / Publish Marks". On the left, there's a section titled "Questions" listing various C programming-related questions. A modal window titled "Deleting" is centered over one of the questions, asking "Are you sure you want to delete this question?". The modal has "Yes" and "No" buttons. To the right of the modal, a sidebar titled "Paper Details" displays the following information:
Code : P10
Name : C
Number of Questions : 5
Marks per correct question : 4
Marks per wrong question : -1.0
Full Marks : 20
Modified Full Marks : 10.0
Department : Computer Science
Paper Setter : Mr. AN

This screenshot shows a mobile device displaying the same list of C programming questions from the "Questions" section of the Extranet application. The device's status bar indicates the time is 01:17 PM.

(1) WhatsApp X Document X +

localhost:8000/exam/view_paper_questions/27/

Extranet

Home Question papers Student marks Unlock paper / Publish Marks

Questions

C is a

Which one is not a keyword in C?

Comment about the code snippet given below.

Minimum possible integer in a 16 bit C compiler for int is

Predict the output of following program.

Predict the output of following program

Predict the output of following program

Double is used to

New question +

Deleting

All questions along with the paper will be deleted. Are you sure u want to delete this paper?

Yes No

View Delete

View Delete

View Delete

View Delete

View Delete

View Delete

Paper Details

Code : P10
Name : C
Number of Questions : 5
Marks per correct question : 4
Marks per wrong question : -1.0
Full Marks : 20
Modified Full Marks : 10.0
Department : Computer Science
Paper Setter : Mr. AN

Update Delete

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.

01:19 PM

create.html - practice ... Document - Google C... PROJECT new 9 - Notepad++

01:19 PM

Form to add/edit question and question paper details (create.html)

This is basically a form to take in all the required details of a question and question paper.

Question paper

PAPER CODE:
P10

PAPER NAME:
C

NO. OF QUESTIONS:
10

MARKS PER CORRECT ANSWER:
4

MARKS PER WRONG ANSWER:
-1

FULL MARKS:
40

MODIFIED FULL MARKS:
15

DURATION OF EXAM:
00:20:00
Enter the duration in [HH:MM:SS] this format

GRACE TIME:
00:00:00
Enter the grace time in [HH:MM:SS] this format

Buttons: Save, Refresh

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
© 2021 ExtraNET.com and its licensors. All rights reserved.

Question

Extranet Home Question papers Student marks Unlock paper / Publish Marks

QUESTION DESCRIPTION:
C is a

UPLOAD RELATED PICTURE(IF ANY):
 No file chosen

OPTION NO.:1:
OOP

OPTION NO.:2:
Object Based Language

OPTION NO.:3:
Procedural Language

OPTION NO.:4:
None

CORRECT OPTION:

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.

2000 1000

C

CORRECT OPTION:
None

For the teacher to view student marks ([marks_for_teacher.html](#))

- ❖ Shows the subject wise tabulation sheet of marks of all the students, paper wise.
- ❖ It has option to change paper within the page itself.
- ❖ Shows the paper details.
- ❖ Shows the tabular form of result with the following columns,
 - Roll
 - Name
 - Obtained marks.
 - Modified obtained marks.
 - Grade.

The screenshot shows a web browser window with the URL localhost:8000/exam/marks_for_teacher/. The page has a brown header bar with the title 'Extranet' and navigation links for Home, Question papers, Student marks, and Unlock paper / Publish Marks. On the left, there's a sidebar with a dropdown menu labeled 'Which Paper?' containing the value '----'. Below it is a 'View marks' button. The main content area displays two tables. The first table provides paper details:

Code	P10	Name	C
Number of Questions	5	Department	Computer Science
Marks per correct question	4	Marks per wrong question	-1.0
Full Marks	20	Modified Full Marks	10.0
Paper Setter	Mr. AN		

The second table shows student marks:

Roll	Name	OM	Modified OM	Grade
521	Subham Jha	20.0	10.0	Excellent

At the bottom of the page, a footer bar contains the text '© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.'

For the teacher to unlock paper and publish marks of students (unlock_papers_and_marks.html)

- ❖ This has the options to unlock a previously set paper, so that students can appear for the same.
- ❖ After the exam has been held, it also gives the option to publish the result, so that students can view their own marks.

Extranet Home Question papers Student marks Unlock paper / Publish Marks

Unlock Papers and publish Marks

Paper Name	Paper Unlock Status	Result status
Networking	Unlocked	<input type="checkbox"/> Click on the checkbox to publish results
MP	Unlocked	Published
C	Unlocked	Exam Going On
JAVA	Unlocked	<input type="checkbox"/> Click on the checkbox to publish results
CG	Unlocked	Published
C++	Unlocked	Published
Discrete Maths	<input type="checkbox"/> Click on the checkbox to unlock paper	Exam has not held yet

© 2021 Extranet.com are registered trademarks of SXC. All rights reserved.

Reset Confirm

Student

Student Dashboard (student_dashboard.html)

- ❖ Homepage for an individual student.
- ❖ Students can go to any page relevant to them from here.
- ❖ Shows a welcome note.
- ❖ Shows the option to change password.
- ❖ Shows the option to logout.
- ❖ Shows the available papers i.e.,
 - Unattended papers to appear for.
 - Attended papers.
 - Missed papers.

The screenshot shows a web browser window with the URL `localhost:8000/exam/student_dashboard/`. The page has a green header bar with the text "Extranet" and "Home Past Exam Results". On the right, it says "Welcome, Subham Jha". The main content area features a background illustration of a person sitting at a desk with a laptop, a red alarm clock, and small trees. The title "Available papers" is centered above a table. The table lists six subjects: C, C++, CG, Networking, MP, and JAVA. To the right of each subject, there is a status indicator: "Give Exam" (green button), "Attended" (blue text), "Attended" (blue text), "Attended" (blue text), "Missed" (red text), and "Missed" (red text). The bottom of the page includes a footer with the text "© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved." and a taskbar showing various open windows.

C	<button>Give Exam</button>
C++	Attended
CG	Attended
Networking	Attended
MP	Missed
JAVA	Missed

The screenshot shows a web browser window with a green header bar. The header bar contains the text "Extranet" on the left, "Home" and "Past Exam Results" in the center, and a user profile icon on the right. The main content area has a light blue background with white polka dots. On the left side, there is a cartoon illustration of a person sitting at a desk with a laptop, a red alarm clock, and small trees. To the right of the illustration, the text "Available papers" is displayed in a large, bold, purple font. Below this, there is a table with a dark blue border containing six rows of exam information. The table columns are "Paper Name" (leftmost), "Status" (rightmost), and an unlabeled column between them. The rows are:

C		Resume
C++		Attended
CG		Attended
Networking		Attended
MP		Missed
JAVA		Missed

At the bottom of the page, there is a footer bar with the text "© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved." and a taskbar at the very bottom showing various open windows and system icons.

localhost:8000/exam/student.dashboard/

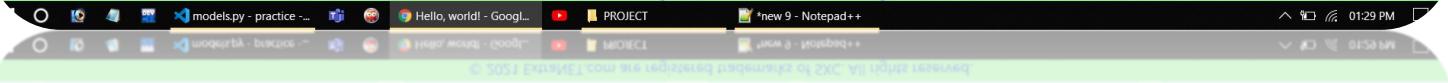
Extranet

Welcome, Subham Jha

Available papers

C++	Attended
C	Attended
CG	Attended
Networking	Attended
MP	Missed
JAVA	Missed

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.



Student's past exam results ([marks_for_student.html](#))

- ❖ Shows the subject wise marks of an individual student.
- ❖ It has option to change paper within the page itself.
- ❖ Shows the paper details.
- ❖ Shows the tabular form of result with the following columns,
 - Obtained marks.
 - Modified obtained marks.
 - Grade.
- ❖ Shows the grade system at the bottom of the screen.

Results of C

Grade: Excellent
Obtained Marks: 20.0
Modified Obtained Marks: 10.0

C Details

Code	P10	Name	C
Number of Questions	5	Department	Computer Science
Marks per correct question	4	Marks per wrong question	-1.0
Full Marks	20	Modified Full Marks	10.0
Paper Setter	Mr. AN		

Grade System

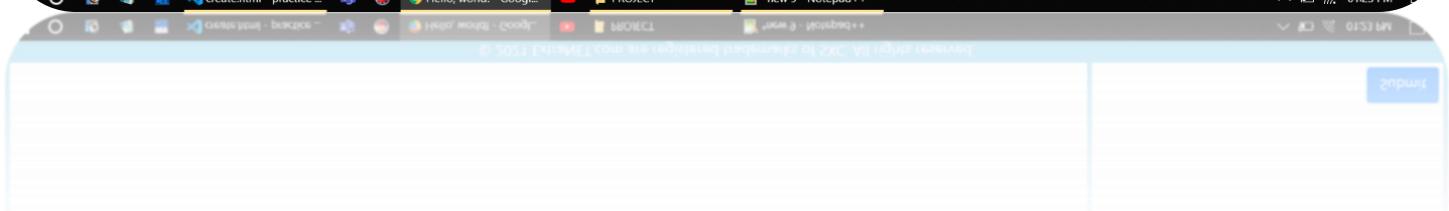
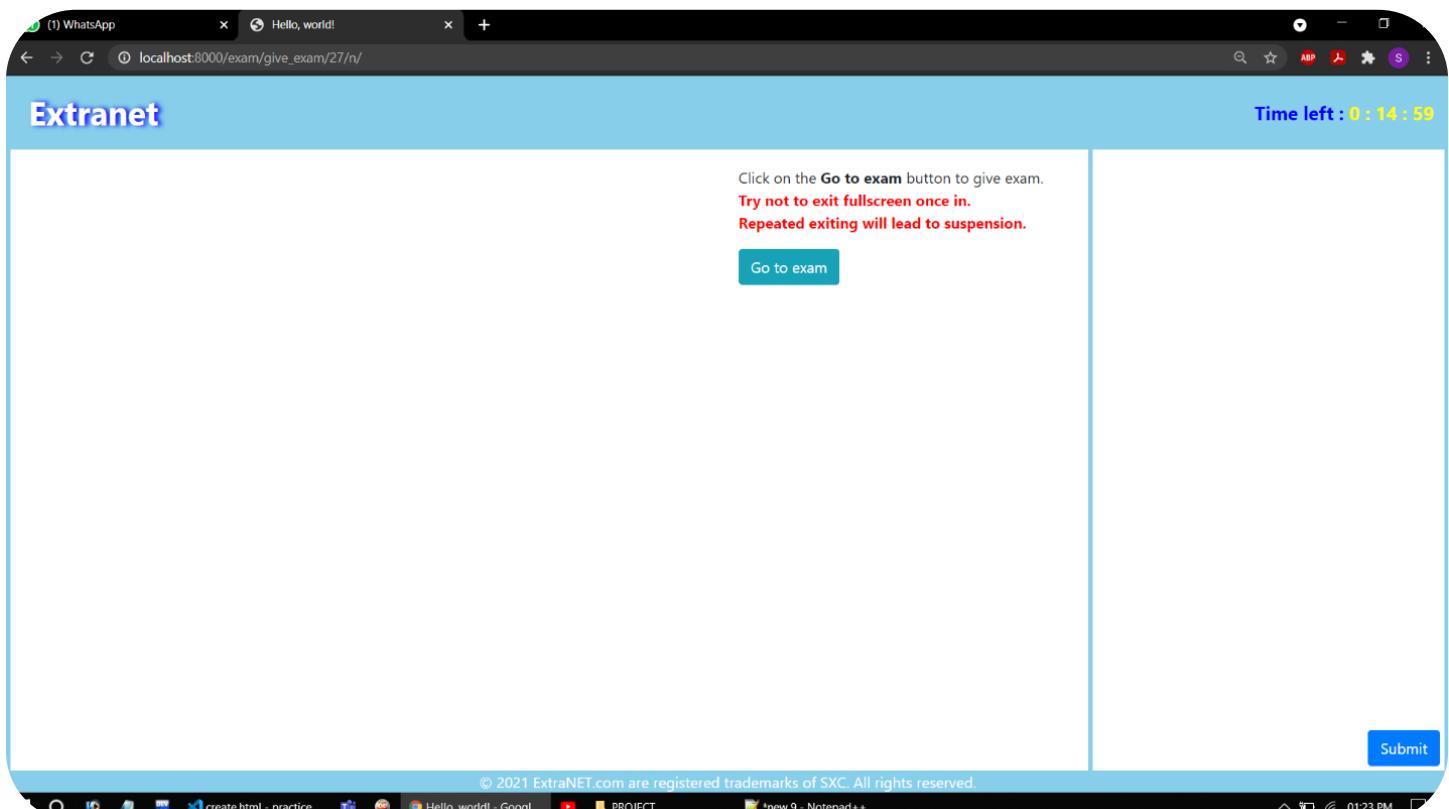
A	B	C	D	E	F
90% - 100%	80% - 90%	70% - 80%	60% - 70%	50% - 60%	< 50%

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.

20% - 20%	80% - 80%	10% - 80%	20% - 10%	20% - 20%	< 20%
A	B	C	D	E	F

Examination page (`ajax_exam.html`)

- ❖ Provides the online examination interface.
- ❖ Shows one question at a time.
- ❖ Shows a countdown timer at the top of the screen.
- ❖ Shows questions in full screen mode only with a predefined number of allowances to exit from the same.
- ❖ Shows warning if one tries to get out from the full screen mode and if number of time, he/ she does the same thing, auto submits his/her answer.
- ❖ If system fails due to technical problems, provides continuity after resuming exam with autosaved answers.



Extranet

Time left : 0 : 12 : 31

Question: 4

Predict the output of following program

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf(" GEEKS %% FOR %% GEEKS");
6     getchar();
7     return 0;
8 }
```

Choose options:

- "GEEKS % FOR % GEEKS"
 - GEEKS % FOR % GEEKS
 - "GEEKS %% FOR %% GEEKS"
 - GEEKS %% FOR %% GEEKS

Clear **Next**

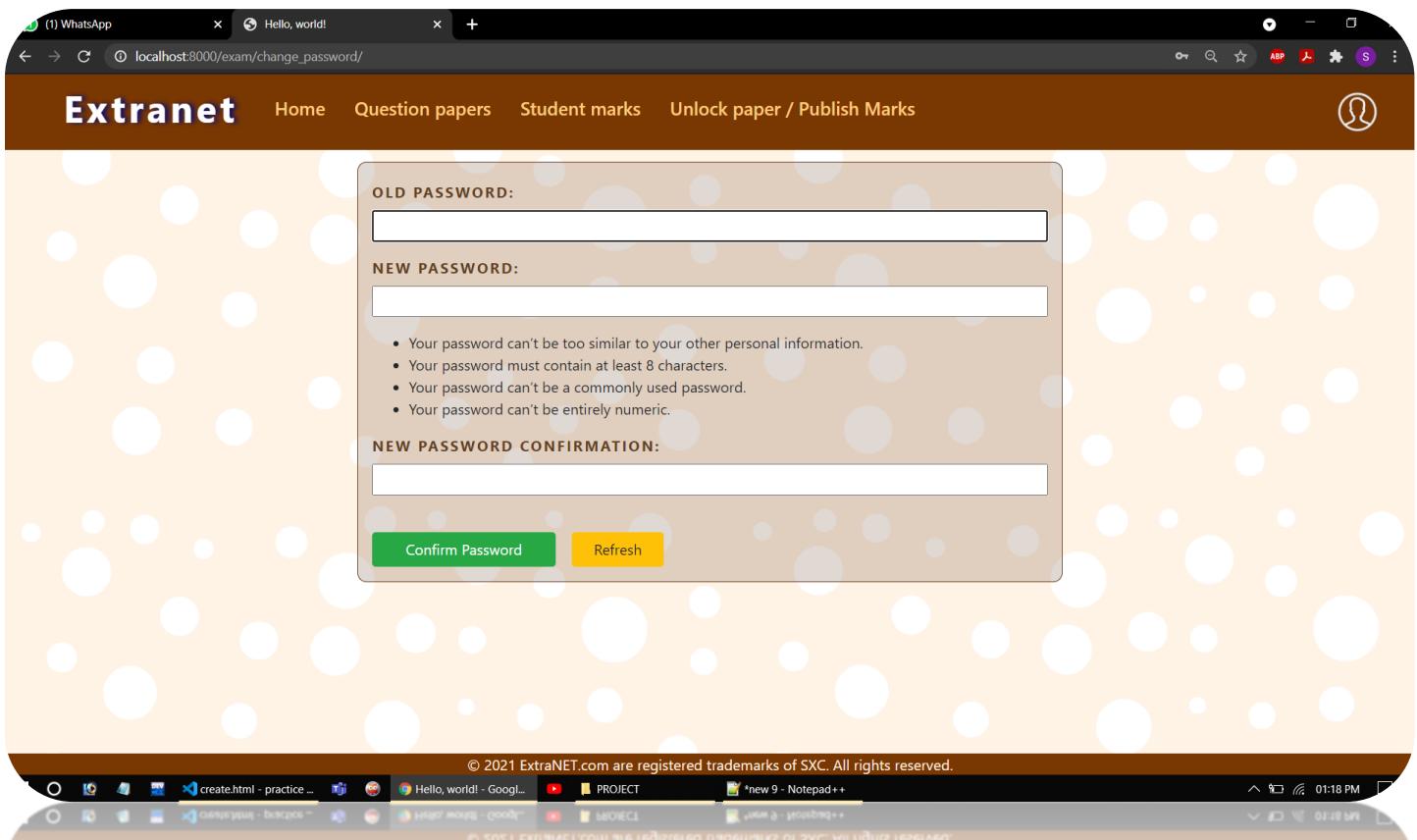
Next

Submit

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.

Change Password (change_password_html)

- ❖ Basically, an interface to change password.



The code

Homepage (login.html)

```
{%load static%}
{% static 'images' as baseUrl %}

<!doctype html>

<title>
    Login
</title>

<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
            <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIJK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
            <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
            <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-ha384-"
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
        <title>Hello, world!</title>
    </head>
    <style>
        .flex-container
        {
            display:flex;
        }
        h1, h3
        {
            font-size:60px;
            background: linear-gradient(rgb(99, 61, 29),white);
        }
    </style>

```

```
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
}
body
{
background-image: url({ % static 'images/tatga.jpg' % });
background-position: center ;
background-
attachment: fixed; /* image is fixed in the viewport so that it doesn't move when the content's hei
ght is greater than the image's height */
background-size: cover; /* image rescale based on the container's size */
background-color: #464646; /*Show when page loads*/
}
#login
{
font-size:28px;
font-weight:bold;
color:rgb(255, 255, 255);
text-shadow: 6px 0px 5px rgb(0, 0, 0);
letter-spacing: 4px;
margin-left: 36px
}

/* width */
::-webkit-scrollbar {
width: 10px;
}

/* Track */
::-webkit-scrollbar-track {
background:skyblue;
opacity:0;
}

/* Handle */
::-webkit-scrollbar-thumb {
background:darkblue;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
```

```
background: #555;
}

.b1 , .btn-success, .btn-warning
{
    width:200px;
    transition-duration: 0.4s;
    border-radius: 50px;
    color: white;
    font-weight: bold;
}
.b1:hover , .btn-success:hover, .btn-warning:hover
{
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.3),0 17px 50px 0 rgba(0,0,0,0.19);
}
table.table-login
{
    margin-left:auto;
    border:0px solid rgb(255, 255, 255);
    border-radius: 10px;
    background-image: linear-
gradient(to right, rgba(255, 255, 255, 0.0) , rgba(255, 255, 255, 0.4));
    width:90%;
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.3),0 17px 50px 0 rgba(0,0,0,0.19);
}
table.table-login > tbody > tr > th
{
    border:0px ;
    color:white;
    font-size:19px;
    text-shadow: 2px 0px 2px black;
    padding-left: 100px;
}
table.table-login > tbody > tr > td
{
    font-size:17px;
    border:0px solid yellow ;
}
input
```

```
border-width: 0px;
border-radius: 10px;
font-size: 17px;
transition-duration: 0.4s;
padding: 5px;
width: 100%  
  
}  
input:hover,input:active  
{  
    box-shadow: 0 12px 16px 0 rgba(0,0,0,0.3),0 17px 50px 0 rgba(0,0,0,0.19);  
    border-bottom:4px solid skyblue;  
}  
#label  
{  
    font-size: 17px;  
    font-weight:none;  
    color: white;  
    letter-spacing: 3px;  
    text-shadow: 0px 0px 5px rgb(0, 0, 0);  
}  
#mainheader  
{  
    margin:10px;  
    border-radius: 10px;  
    background-image: linear-gradient  
(to right, rgba(255, 255, 255, 0.0) 30% , rgba(255, 255, 255, 0.4));  
    font-family: verdana;  
    width:100%;  
    padding: 25px;  
    font-size:40px;  
    font-weight:bold;  
    color:rgb(255, 255, 255);  
    text-shadow: 6px 0px 5px rgb(0, 0, 0);  
    letter-spacing: 4px;  
}  
</style>  
  
<!--Login portal-->  
<div class=block>
```

```

<Div class="row ml-0 mr-0 pt-1"><div id="mainheader" class="text-right">EXTRANET</div></Div>
<Div class="row ml-0 mr-0">
<div class="col-5" style=" ">
<div class="" style="height:7vh"></div>
<div class="align-items-top " style="height:73vh; ">
<form method="post" action="">
<table class="table table-login" >
<tr style="">
<td colspan=2 ><div class="text-left" id="login" style="font-family: verdana;">LOG IN</div></td>
</tr>
{ % csrf_token % } <!--BACKEND-->
<tr class="">
<td class="pt-2 pl-5 pr-5" colspan=2 ><div id="label">USERNAME:</div><input type="text" name="username" size=20 placeholder="Enter your Username" REQUIRED><br></td>
</tr>
<tr>
<td class="pb-5 pl-5 pr-5" colspan=2 ><div id="label">PASSWORD:</div><input type="password" name="password" size=20 placeholder="Enter your password" REQUIRED><br></td>
</tr>
<tr style="">
<td class="pl-5 pb-5" ><input type="reset" class="btn btn-warning" value="RESET"></td>
<td><div class="d-flex justify-content-end pr-5" ><input type="submit" class="btn btn-success" value="LOG IN"></div></td>
</tr>
</table>
<div id="label" class="text-danger">{ % for m in messages% } {{m}} { % endfor % }<!--BACKEND--></div>
</form>
</div>
</div>
<div class="col-9" style="overflow: auto; ">
</div>
</Div>
</div>
</html>

```

Teacher

Teacher Dashboard (teacher_dashboard.html)

```
{%load static%}
{% static 'images' as baseUrl %}

<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-ha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
</head>
<style>
.flex-container
{
    display:flex;
}
h1, h3
{
    font-size:60px;
    background: linear-gradient(rgb(99, 61, 29),white);
    -webkit-background-clip: text;
    -webkit-text-fill-color: transparent;
}
```

```
body
{
  align-content:center;
  background-image: url({% static 'images/Untitled.png' %});
  background-size:cover;
}
.border
{
  display: block;
  margin: 3px;
}
.footer
{
  width:20%;
  margin-left:auto;
}
.table-bordered-2
{
  width:100%;
  border:0px;
  margin-left: auto;
}
table.table-bordered-2 > tbody > tr > td
{
  text-align:left;
  padding:5px;
  border:0px blue solid;
}
table.table-bordered-2 > tbody > tr > th
{
  text-align:right;
}
.table-bordered-1
{
  width:100%;
  border:0px;
  margin-left: auto;
}
table.table-bordered-1 > tbody > tr > td
{
  border:0px solid blue;
```

```
}

.fade2
{
    transform: scale(0);
    opacity: 0;
    transition: all 0.15s linear;
    display: block !important; /*For fading out effect*/
}
.fade2.show
{
    opacity:1;
    transform: scale(1);
}
.foods
{
    text-align:center;
}
.foods1
{
    text-align:center;
    align-content: center;
}
.button-1
{
    background-color:#fff3e6;
    color:white;
    transition:0.15s;
    border-radius: 50% ;
    padding:20px;
}
.button-2
{
    background-color:#793702;
    color:#fff3e6;
    transition:0.15s;
    border-radius: 15px 15px 15px 15px;
    width:100%;
}
.button-1:hover
{
    background-color:#ffbf5f;
```

```
    transition:0.15s;
}
.button-2:hover
{
  background-color:#ffbf5f;
  transition:0.15s;
}
.bottom/*footer*/
{
  background-color: #793702; color:#fff3e6;
  border-radius: 0px 0px 25px 25px;
}
.desc
{
  color:#793702;
  font-size: 25px;
  font-weight: bold;
}
.table-1
{
  border:0px;
}
table.table-1 > tbody > tr > td
{
  padding:15px;
  border:0px blue solid;
}
.dropdown-menu
{
  background-color:#fff3e6 ;
}
.dropdown-item
{
  font-weight: 500;
  color:#793702;
}
.dropdown-item:hover
{
  background-color:#ffcc80;
  transition: 0.15s;
}
```

```
.nav-link
{
  color:#ffcc80;
  transition: 0.15s;

}
.nav-link:hover
{
  color:#fff3e6;
  transition: 0.15s;
}
</style>

<!--Navigation Bar-->
<style>
.c
{
  color:#ffcc80;
  transition: 0.15s;

}
.c:hover
{
  color:#fcdbcb;
  transition: 0.15s;
  text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
  background-color:#fff3e6 ;
}
.dropdown-item
{
  font-weight: 500;
  color:#793702;
}
```

```
.dropdown-item:hover
{
  background-color:#ffcc80;
  transition: 0.15s;
}
.button-1
{
  background-color:#fff3e6;
  color:white;
  transition:0.15s;
  border-radius: 50% 50% / 50% 50%;
  padding:20px;
}
.button-1:hover
{
  background-color:#ffbfb5f;
  transition:0.15s;
}
h1, h4
{
  font-size:40px;
  font-weight:bold;
  color:rgb(255, 255, 255);
  text-shadow: 4px 0px 5px rgb(11, 13, 112);
  letter-spacing: 4px;
}
.dropdown-menu
{
  transition: 0.5s;
}
.dropdown-menu:active
{
  transition: 0.5s;
}
.dropdown-menu
{
  padding: 0px;
  border:1px solid grey;
}
```

</style>

```

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
  <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-4" href="{% url 'teacher_dashboard' % }"><div class=c><h5>Home</h5></div></a>
      </li>
    </ul>
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-4" href="{% url 'papers_and_questions' % }"><div class=c><h5>Question papers</h5></div></a>
      </li>
    </ul>
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-4" href="{% url 'marks_for_teacher' % }"><div class=c><h5>Student marks</h5></div></a>
      </li>
    </ul>
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-4" href="{% url 'unlock_papers_and_marks' % }"><div class=c><h5>Unlock paper / Publish Marks</div></a>
      </li>
    </ul>
  </div>
  <ul class="navbar-nav">
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="color:#793702;"></a>
    </li>
  </ul>
</nav>

```

```

<div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
    <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold; ">&nbsp;&
nbsp;&nbsp;&nbsp;{{ request.user.teacher.name }}</h5>
    <a class="dropdown-item" href="{% url 'change_password' %}">Change password</a>
    <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></div></button>
</li>
</ul>
</nav>

<!--body-->
<div class="d-sm-block text-primary" style="height:88vh; margin-top:80px">
    <table style="width:100%; table-layout: fixed ;">
        <tr>
            <td rowspan=2></td>
            <td>
                <div class="header" style="color:#793702; font-weight:bold; font-
family:Harlow Solid; text-align:center; font-
size: 40px;">Welcome, {{ request.user.teacher.name }}<br>
                    {% comment %} {{ request.user.teacher.paper_set.all.0 }} {% endcomment %}</div>
            </td>
        </tr>
        <tr>
            <td>
                <div class="mad" style="display: flex;
                    justify-content: center;
                    align-items: center;">
                    <table class="table table-1" style="width:70%">
                        <tr>
                            <td><button class="button button-2"><a class="nav-
link" href="{% url 'papers_and_questions' %}" style="font-
weight:bold; color:#fff3e6;">Question Papers</a></button></td>
                            <td><div class="desc">Click on this to set a question paper.</div></td>
                        </tr>
                    </table>
                </div>
            </td>
        </tr>
    </table>
</div>

```

```
</tr>
<tr>
    <td><button class="button button-2"><a class="nav-link" href="{% url 'marks_for_teacher' %}" STYLE="font-weight:bold; color:#fff3e6;">Student Marks</a></button></td>
        <td><div class=desc>Click on this to check the marks of the student.</div></td>
    </tr>
    <tr>
        <td><button class="button button-2"><a class="nav-link" href="{% url 'unlock_papers_and_marks' %}" STYLE="font-weight:bold; color:#fff3e6">Unlock Paper/Publish MArks</a></button></td>
            <td><div class=desc>Click on this to check to unlock the paper or publish the marks of the examination.</div></td>
        </tr>
    </table>
</div>
</td>
</tr>
</table>
</div>
```

```
<!--footer-->
<style>
.b/*footer*/
{
    position:fixed;
    bottom:0;
    width:100%;
    background-color: #793702; color:#fff3e6;
}
.footer
{
    width:20%;
    margin-left:auto;
}
</style>
```

```
<div class="b">
<div trademark style="text-align:center; ">
    © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
</div>
</div>
</html>
```

Create a new question paper or edit previously set question papers (papers_and_questions.html)

```
<!doctype html>
<html lang="en">
  { %load static% }
  {%- static 'images' as baseUrl %}

  <!--Linking bootstrap online-->
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIJK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-ha384-"
ha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
  <title>Hello, world!</title>
</head>
<style>
body
{
  align-content:center;
  background-color: #fff3e6;
  height: 70vh; /*text-align:center;*/
  background-image: url({%- static 'images/Untitled.png' %});
  background-size:cover;
}
</style>
```

```
<!--Navigation Bar-->
<style>
.c
{
  color:#ffcc80;
  transition: 0.15s;

}
.c:hover
{
  color:#fcdbcb;
  transition: 0.15s;
  text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
  background-color:#fff3e6 ;
}
.dropdown-item
{
  font-weight: 500;
  color:#793702;

}
.dropdown-item:hover
{
  background-color:#ffcc80;
  transition: 0.15s;
}
.button-1
{
  background-color:#fff3e6;
  color:white;
  transition:0.15s;
```

```

border-radius: 50% ;
padding:20px;
}
.button-1:hover
{
background-color:#ffbf5f;
transition:0.15s;
}
h1, h4
{
font-size:40px;
font-weight:bold;
color:rgb(255, 255, 255);
text-shadow: 4px 0px 5px rgb(11, 13, 112);
letter-spacing: 4px;
}
.dropdown-menu
{
transition: 0.5s;
}
.dropdown-menu:active
{
transition: 0.5s;
}
.dropdown-menu
{
padding: 0px;
border:1px solid grey;
}
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
  <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item ">

```

```

        <a class="nav-link pl-
4" href="{% url 'teacher_dashboard' %}"><div class=c><h5>Home</h5></div></a>
        </li>
    </ul>
    <ul class="navbar-nav">
        <li class="nav-item ">
            <a class="nav-link pl-
4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5></div></
a>
            </li>
        </ul>
        <ul class="navbar-nav">
            <li class="nav-item ">
                <a class="nav-link pl-
4" href="{% url 'marks_for_teacher' %}"><div class=c><h5>Student marks</h5></div></a>
                </li>
            </ul>
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link pl-
4" href="{% url 'unlock_papers_and_marks' %}"><div class=c><h5>Unlock paper / Publish M
arks</div> </a>
                    </li>
                </ul>
            </div>
            <ul class="navbar-nav">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false" style="color:#793702;"></a>
                    <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
                        <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold;">&nbs
p;&nbs
p;{ {request.user.teacher.name} }</h5>
                        <a class="dropdown-item" href="{% url 'change_password' %}">Change password</a>
                        <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></div></button>
                    </li>
                </ul>
            
```

```
</nav>

<!--body-->
<style>
  .submit-1
  {
    background-color:#fff3e6;
    color:#793702;
    transition:0.15s;

    padding-right: 13px;
    padding-left: 13px;
  }
  .submit-1:hover
  {
    background-color:#ffbf5f;
    transition:0.15s;
  }
  .list-group-item
  {
    background-color: transparent;
    border:0px solid #793702;
    border-radius: 0;
    color:#793702;
  }
</style>
<div class="block" style="margin-top: 80px; height:88vh;">
  <div class="row ml-0 mr-0 ">
    <div class="col-9" style="height:88vh;">
      <div class="h-100 row align-items-center" style="text-align: center;">
        <div class="col">
          <div class="a" style="color:#793702; font-weight:bold; font-family:Harlow Solid; text-align:center; font-size: 80px;">
            Create a new question paper<br></div>
```

```

        <a href="{% url 'create_paper' %}"><button class="submit submit-
1 " style="color:#793702; font-weight:bold; font-family:Harlow Solid; text-align:center; font-
size: 40px;" >+</button></a>
        </div>
        </div>
        </div>
        <div class="col-3" style="text-align: center; border-left:5px solid #793702; border-
right:5px solid #793702;">
        <div class="a" style="color:#793702; font-weight:bold; font-family:Modern No. 20; text-
align:center; font-size: 30px;"><br>Previously set papers</div>
        <br><br><br>
        <ul class="list-group list" style="font-weight:bold;">
            {% for x in papers %}
            <li class="list-group-
item"><a href="{% url 'view_paper_questions' x.id %}" style="color:#793702; font-
family:Modern No. 20; text-align:center;">{{x.name}}</a></li>
            {% endfor %}
        </ul>
        </div>
        </div>
        </div>
    </div>

<!--footer-->
<style>
    .b/*footer*/
    {
        position:fixed;
        bottom:0;
        width:100%;
        background-color: #793702; color:#fff3e6;
    }
    .footer
    {
        width:20%;
        margin-left:auto;
    }
</style>
<div class="b">
    <div trademark style="text-align:center; ">
```

© 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.

```
</div>
</div>
</html>
```

Add / Edit questions (view_paper_questions.html)

```
{%load static%}
{% static 'images' as baseUrl %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
</head>

<style>
body
{
```

```
    align-content:center;
    background-color: #fff3e6;
    height: 70vh; /*text-align:center;*/
    background-image: url({% static 'images/Untitled.png' %});
    background-size:cover;
}
</style>
```

```
<style>
body
{
    align-content:center;
    background-color: #fff3e6;
    height: 70vh; /*text-align:center;*/
    background-image: url({% static 'images/Untitled.png' %});
    background-size:cover;
}
</style>
```

```
<!--Navigation Bar-->
<style>
.c
{
    color:#ffcc80;
    transition: 0.15s;

}
.c:hover
{
```

```
color:#fcdbc9;
transition: 0.15s;
text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
background-color:#fff3e6 ;
}
.dropdown-item
{
font-weight: 500;
color:#793702;

}
.dropdown-item:hover
{
background-color:#ffcc80;
transition: 0.15s;
}
.button-1
{
background-color:#fff3e6;
color:white;
transition:0.15s;
border-radius: 50%;
padding:20px;
}
.button-1:hover
{
background-color:#ffb5f5;
transition:0.15s;
}
h1, h4
{
font-size:40px;
font-weight:bold;
color:rgb(255, 255, 255);
text-shadow: 4px 0px 5px rgb(11, 13, 112);
letter-spacing: 4px;
}
.dropdown-menu
```

```

    {
      transition: 0.5s;
    }
    .dropdown-menu:active
    {
      transition: 0.5s;
    }
    .dropdown-menu
    {
      padding: 0px;
      border: 1px solid grey;
    }
  </style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
  <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-4" href="{% url 'teacher_dashboard' %}"><div class=c><h5>Home</h5></div></a>
        </li>
      </ul>
      <ul class="navbar-nav">
        <li class="nav-item ">
          <a class="nav-link pl-4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5></div></a>
          </li>
        </ul>
        <ul class="navbar-nav">
          <li class="nav-item ">
            <a class="nav-link pl-4" href="{% url 'marks_for_teacher' %}"><div class=c><h5>Student marks</h5></div></a>
            </li>
          </ul>
        </ul>
      </li>
    </ul>
  </div>
</nav>

```

```

<ul class="navbar-nav">
    <li class="nav-item ">
        <a class="nav-link pl-
4" href="{ % url 'unlock_papers_and_marks' % }"><div class=c><h5>Unlock paper / Publish M
arks</div> </a>
    </li>
</ul>
</div>
<ul class="navbar-nav">
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false" STYLE="color:#793702;"></a>
        <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
            <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold; ">&nbsp;&
nbsp;&nbsp;&nbsp;{ { request.user.teacher.name } }</h5>
            <a class="dropdown-item" href="{ % url 'change_password' % }">Change password</a>
            <a class="dropdown-item" href="{ % url 'logout' % }">Logout</a></div></button>
        </li>
    </ul>
</nav>

<!--question details-->
<style>

    /* width */
    ::-webkit-scrollbar {
        width: 10px;
    }

    /* Track */
    ::-webkit-scrollbar-track {
        background:transparent;
        opacity:0;
    }

```

```

        }

/* Handle */
::-webkit-scrollbar-thumb {
    background:#ffbf5f;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
    background: #555;
}
.b1
{
    color:#fff3e6;
    background-color: #793702;
}
.b1:hover
{
    background-color: #ffbf5f;
}

.table-questions
{
    border:3px solid brown
}
table.table-questions > tbody > tr
{
    border:2px solid brown;
}

}
</style>

<div class="block" style="background-image: url(Untitled.png); margin-top: 80px;">
    <div class="row ml-0 mr-0">
        <div class="col-9" style="height:100vh; border-left:5px solid #793702; overflow: auto; ">
            <div class="row">
                <div class="col-md-12 text-right p-2">
                    <a href="{% url 'create_question' paper.id %}"><button type="button" class="btn btn- b
1 ">New question +</button></a>
                </div>
            </div>
        </div>
    </div>

```

```

<div class=row >
  <div class="col-md-12 p-4">
    <p style="color:#793702; font-weight:bold; font-family:Modern No. 20; font-size: 30px; text-decoration: underline;">Questions</p>
    <!--{ %load static% } { % static 'qstns' as baseUrl % }-->
    <div id="Question1111"></div>
  </div>
</div>
</div>

<!--modal to edit questions-->
<style>
  .fade2
  {
    transform: scale(0);
    opacity: 0;
    transition: all 0.15s linear;
    display: block !important; /*For fading out effect*/
  }
  .fade2.show
  {
    opacity:1;
    transform: scale(1);
  }
  .m-footer
  {
    text-align:center;
  }
</style>
<div id="view" class="modal fade2" role="dialog">
  <div class="modal-dialog modal-dialog-centered modal">
    <div class="modal-content">
      <div class="modal-header text-light" STYLE="background-color:#5e2b02">
        <h5 class="modal-title" id="login">Question</h5>
        <button type="button" class="close text-light" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
      </div>
      <div class=foods1>
        <div class="modal-body" STYLE="background-color:#fff3e6 color:#793702; font-weight:bold; font-family:Modern No. 20; font-size: 18px;">

```

```
<div id="popupques">
</div>
</div>
<div class=m-footer>
<div class="modal-footer" STYLE="background-color:#5e2b02">
<div class="col">
<div id="update "><a href="#"><button class="button" class="btn btn-warning">Update</button></a> </div>
</div>
</div>
</div>
</div>
</div>

<!--Modal to delete questions-->
<style>
.fade2
{
    transform: scale(0);
    opacity: 0;
    transition: all 0.15s linear;
    display: block !important; /*For fading out effect*/
}
.fade2.show
{
    opacity:1;
    transform: scale(1);
}
.m-footer
{
    text-align:center;
}
</style>
<div id="delete_question" class="modal fade2" role="dialog">
<div class="modal-dialog modal-dialog-centered modal">
<div class="modal-content">
<div class="modal-header text-light" STYLE="background-color:#5e2b02">
<h5 class="modal-title" id="login">Deleting</h5>
```

```

<button type="button" class="close text-light" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
</div>
<div class=foods1>
  <div class="modal-body" STYLE="background-color:#fff3e6">
    <br>
    Are you sure you want to delete this question?
    <br><br>
  </div>
  <div class="modal-footer" STYLE="background-color:#5e2b02">
    <div id="deletequestion"></div>
  </div>
</div>
</div>
</div>
</div>
</div>

```

```

<!--paper details-->
<div class="col-3" style="height:100vh; text-align: center; border-
left:5px solid #793702; border-right:5px solid #793702; overflow: auto;">
  <div class=a style="color:#793702; font-weight:bold; font-family:Modern No. 20; text-
align:center; font-size: 30px;">Paper Details</div>
  <br><br><br>
  <p style="color:#793702; font-weight:bold; font-family:Modern No. 20; text-
align:center; font-size: 18px;">
    Code : {{paper.code}} <br>
    Name : {{paper.name}} <br>
    Number of Questions : {{paper.count}} <br>
    Marks per correct question : {{paper.marks_per_cq}} <br>
    Marks per wrong question : {{paper.marks_per_wq}} <br>
    Full Marks : {{paper.fm}} <br>
    Modified Full Marks : {{paper.modified_fm}} <br>
    Department : {{paper.dept.name}} <br>
    Paper Setter : {{paper.paper_setter.name}} <br><br>
    <div class="row">
      <div class="col"> <a href="{% url 'update_paper' paper.id %}"><button type="button" cl-
ass="btn btn-none b1">Update</button></a></div>
      <div class="col"> <a a class="nasd" href="#" data-target="#delete_paper" data-
toggle="modal"> <button type="button" class="btn btn-danger">Delete</button></a></div>
    
```

```

        </div>
    </p>
</div>
</div>
</div>

<!--Modal to Delete paper--&gt;
&lt;style&gt;
.fade2
{
    transform: scale(0);
    opacity: 0;
    transition: all 0.15s linear;
    display: block !important; /*For fading out effect*/
}
.fade2.show
{
    opacity:1;
    transform: scale(1);
}
.m-footer
{
    text-align:center;
}
&lt;/style&gt;
&lt;div id="delete_paper" class="modal fade2" role="dialog"&gt;
    &lt;div class="modal-dialog modal-dialog-centered modal"&gt;
        &lt;div class="modal-content"&gt;
            &lt;div class="modal-header text-light" STYLE="background-color:#5e2b02"&gt;
                &lt;h5 class="modal-title" id="login"&gt;Deleting&lt;/h5&gt;
                &lt;button type="button" class="close text-light" data-dismiss="modal" aria-
label="Close"&gt;&lt;span aria-hidden="true"&gt;&amp;times;&lt;/span&gt;&lt;/button&gt;
            &lt;/div&gt;
            &lt;div class="modal-body" STYLE="background-color:#fff3e6"&gt;
                &lt;br&gt;All questions along with the paper will be deleted. Are you sure u want to delete this
paper?
                &lt;br&gt;&lt;br&gt;
            &lt;/div&gt;
            &lt;div class="modal-footer" STYLE="background-color:#5e2b02"&gt;
                &lt;form method="post" action="{% url 'delete_paper' paper.id %}"&gt;
</pre>

```

```
{% csrf_token %}  
    <div class="row ml-0 mr-0" >  
        <div class="col" ><input type="submit" class="btn btn-success" value="Yes"></div>  
        <div class="col" ><button type="button" class="btn btn-danger" data-  
dismiss="modal">No</button></div>  
    </form>  
    </div>  
    </div>  
    </div>  
    </div>  
</div>  
  
  
<style>  
.b/*footer*/  
{  
    position:fixed;  
    bottom:0;  
    width:100%;  
    background-color: #793702; color:#fff3e6;  
}  
.footer  
{  
    width:20%;  
    margin-left:auto;  
}  
</style>  
<div class="b">  
    <div trademark style="text-align:center; ">  
        © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.  
    </div>  
</div>  
  
  
<script type="text/javascript">  
    var ques = JSON.parse('{{ q | escapejs }}');
```

```

var ques1 = JSON.parse(ques);
let count = 1;
console.log(ques1);
var no = Object.keys(ques1).length;
console.log(no);
var x ='<table class="table table-questions m-auto">';
while(count<=no)
{
  x=x+'<tr class="d-flex">'+
    '<td class="col-8"><a href="#" style="color:#793702; font-weight:bold; font-
family:Modern No. 20; text-decoration:none;" >'+ques1[(count-1)].fields.desc+'</a></td>'+
    '<td class="col-2 text-right"><a class="nasd" href="#" data-target="#view" data-
toggle="modal"> <button type="button" onclick=showQuestions('+count+'); class="btn b1">Vie
w</button></a></td>'+
    '<td class="col-2 text-right"--><a class="nasd" href="#" data-
target="#delete_question" data-
toggle="modal"> <button type="button" onclick=delete_Questions('+count+') class="btn btn-
danger">Delete</button></a></td></tr>';
  count++;
}
x = x+'</table>';
document.getElementById("Question1111").innerHTML=x;
function showQuestions(no)
{
  var no = Number(no);
  //console.log(no);
  var ques = ques1[(no-1)].pk;
  console.log(ques);
  var view = '<p> Desc : '+ ques1[(no-1)].fields.desc+'</p> <br>';
  if((ques1[(no-1)].fields.img_path).length!=0)
    view+='<br>';
  view+='<ol type="A">'+
    '<li>'+ques1[(no-1)].fields.op1+'</li> <br>'+
    '<li>'+ques1[(no-1)].fields.op2+'</li> <br>'+
    '<li>'+ques1[(no-1)].fields.op3+'</li> <br>'+
    '<li>'+ques1[(no-1)].fields.op4+'</li> <br>'+
    '</ol>'+
    'Correct Option : '+ ques1[(no-1)].fields.correct_option+' <br>' 
  document.getElementById("popupques").innerHTML=view;
  var update ="";
```

```
update = update + '<a href="/exam/update_question/' + ques +'" style="color:black;"><button type="button" class="btn btn-warning">Update</button></a>';
document.getElementById("update").innerHTML=update;
}
function delete_Questions(no)
{
var no = Number(no);
console.log(no);
var ques = ques1[(no-1)].pk;
console.log(ques);
var delete1 = '<form method="POST" action="/exam/delete_question/' + ques + '">' +
'{% csrf_token %}' +
'<div class="row"><div class="col"><input type="submit" class="btn btn-success" value="Yes"></div>' +
'<div class="col"><button type="button" class="btn btn-danger" data-dismiss="modal">No</button></div>' +
'</form></div>';
document.getElementById("deletequestion").innerHTML=delete1;
}
</script>
</html>
```

Form to add/edit question and question paper details (create.html)

```
<!doctype html>
<html lang="en">
  {%load static%}
  {% static 'images' as baseUrl %}

  <!--Linking bootstrap online-->
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
      <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIJK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-sh1ggsg0Kw4KfQpHJdZuH7hZ070sM0mQc3mOiuRrXaAO+g30kvvPS" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
  </head>
  <style>
    body
    {
      align-content:center;
      background-color: #fff3e6;
      height: 70vh;
      background-image: url({%static 'images/Untitled.png%'});
      background-size:cover;
    }
  </style>

```

```
        }
    </style>

<!--Navigation Bar--&gt;
&lt;style&gt;
.c
{
    color:#ffcc80;
    transition: 0.15s;

}
.c:hover
{
    color:#fcdbcb;
    transition: 0.15s;
    text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
    background-color:#fff3e6 ;
}
.dropdown-item
{
    font-weight: 500;
    color:#793702;

}
.dropdown-item:hover
{
    background-color:#ffcc80;
    transition: 0.15s;
}
.button-1
{
    background-color:#fff3e6;</pre>
```

```
        color:white;
        transition:0.15s;
        border-radius: 50%;
        padding:20px;
    }
    .button-1:hover
    {
        background-color:#ffbf5f;
        transition:0.15s;
    }
    h1, h4
    {
        font-size:40px;
        font-weight:bold;
        color:rgb(255, 255, 255);
        text-shadow: 4px 0px 5px rgb(11, 13, 112);
        letter-spacing: 4px;
    }
    .dropdown-menu
    {
        transition: 0.5s;
    }
    .dropdown-menu:active
    {
        transition: 0.5s;
    }
    .dropdown-menu
    {
        padding: 0px;
        border:1px solid grey;
    }
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
    <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
```

```

<ul class="navbar-nav">
    <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'teacher_dashboard' %}" ><div class=c><h5>Home</h5></div></a>
    </li>
</ul>
<ul class="navbar-nav">
    <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5>
</div></a>
    </li>
</ul>
<ul class="navbar-nav">
    <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'marks_for_teacher' %}" ><div class=c><h5>Student marks</h5></di
v></a>
    </li>
</ul>
<ul class="navbar-nav">
    <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'unlock_papers_and_marks' %}"><div class=c><h5>Unlock paper / Pu
blish Marks</div> </a>
    </li>
</ul>
</div>
<ul class="navbar-nav">
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-
haspopup="true" aria-
expanded="false" STYLE="color:#793702;"></a>
            <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
                <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold; ">&nbsp;&nbsp;&nbsp;&nbsp;{{request.user.teacher.name}}</h5>

```

```
        <a class="dropdown-item" href="{% url 'change_password' %}">Change password</a>
            <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></div></button>
        </li>
    </ul>
</nav>

<style>
    label
    {
        font-weight:bold;
        color:rgb(99,61,29);
        letter-spacing: 2px;
    }
    input:hover,textarea:hover,select:hover
    {
        box-shadow: 0 2px 6px 0 rgba(99,61,29,0.3),0 7px 5px 0 rgba(99,61,29,0.3);
    }
    ::-webkit-scrollbar
    {
        width: 10px;
    }
    /* Track */
    ::-webkit-scrollbar-track
    {
        background:transparent;
        opacity:0;
    }

    /* Handle */
    ::-webkit-scrollbar-thumb
    {
        background:#ffbf5f;
    }

    /* Handle on hover */
    ::-webkit-scrollbar-thumb:hover
    {
        background: #555;
```

```
        }
    </style>

    <div class="block" style="margin-top:95px; ">
        <div class="row ml-0 mr-0">
            <div class="col-sm" style="height:0vh; text-align:center; border-left:0px solid #793702;"><br><br><br><br><br><br><br><br><br></div>
            <div class="col-6" style=" padding-top: 20px;background-color: rgba(99,61,29,0.2); border-radius:10px; border: 1px solid rgb(99,61,29);">
                <div class="form-group">
                    {% comment %}

                    {% endcomment %}
                    <form action="" method="POST" enctype="multipart/form-data">
                        {% csrf_token %} {{form.as_p}}<br />
                        {% comment %} {{form.errors}} {% endcomment %}
                        <input class="btn btn-success" type="submit" value="Save" style="width: 100px"/>
                        <input class="btn btn-warning" type="reset" value="Refresh" style="width: 100px"/>
                    </form>
                </div>
            </div>
            <div class="col-sm" style="height:0vh; text-align:center; border-right:0px solid #793702;"><br><br><br><br><br><br><br><br><br></div>
        </div>
    </div>
    <br> <br>
<!--footer-->
<style>
    .b
    {
        position:fixed;
        bottom:0;
        width:100%;
        background-color: #793702; color:#fff3e6;
    }
    .footer
    {
        width:20%;
```

```
    margin-left:auto;
}
</style>
<div class="b">
    <div trademark style="text-align:center; ">
        © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved
    .
    </div>
</div>
</html>
```

For the teacher to view student marks (marks_for_teacher.html)

```
{%load static%}

{ % static 'images' as baseUrl % }
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYI" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
</head>

<style>
    body
```

```
{  
  align-content:center;  
  background-color: #fff3e6;  
  height: 70vh; /*text-align:center;*/  
  background-image: url({% static 'images/Untitled.png' %});  
  background-size:cover;  
}  
</style>
```

```
<!--Navigation Bar-->  
<style>  
.c  
{  
  color:#ffcc80;  
  transition: 0.15s;  
  
}  
.c:hover  
{  
  color:#fcdcb9;  
  transition: 0.15s;  
  text-shadow: 2px 0px 2px rgb(180, 114, 59);  
}  
.dropdown-menu  
{  
  background-color:#fff3e6 ;  
}  
.dropdown-item  
{  
  font-weight: 500;  
  color:#793702;  
  
}  
.dropdown-item:hover  
{
```

```
background-color:#ffcc80;
transition: 0.15s;
}
.button-1
{
background-color:#fff3e6;
color:white;
transition:0.15s;
border-radius: 50%;
padding:20px;
}
.button-1:hover
{
background-color:#ffb5f5;
transition:0.15s;
}
h1, h4
{
font-size:40px;
font-weight:bold;
color:rgb(255, 255, 255);
text-shadow: 4px 0px 5px rgb(11, 13, 112);
letter-spacing: 4px;
}
.dropdown-menu
{
transition: 0.5s;
}
.dropdown-menu:active
{
transition: 0.5s;
}
.dropdown-menu
{
padding: 0px;
border:1px solid grey;
}
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
```

```

<a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#collapsibleNavbar">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="collapsibleNavbar">
  <ul class="navbar-nav">
    <li class="nav-item ">
      <a class="nav-link pl-
4" href="{% url 'teacher_dashboard' %}"><div class=c><h5>Home</h5></div></a>
      </li>
    </ul>
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5></div></a>
        </li>
      </ul>
      <ul class="navbar-nav">
        <li class="nav-item ">
          <a class="nav-link pl-
4" href="{% url 'marks_for_teacher' %}"><div class=c><h5>Student marks</h5></div></a>
          </li>
        </ul>
        <ul class="navbar-nav">
          <li class="nav-item ">
            <a class="nav-link pl-
4" href="{% url 'unlock_papers_and_marks' %}"><div class=c><h5>Unlock paper / Publish M
arks</div></a>
            </li>
          </ul>
        </div>
        <ul class="navbar-nav">
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false" STYLE="color:#793702;"></a>
            <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">

```

```
<h5 class="dropdown-header" style="color:#fff3e6; background-color: #793702; padding: 20px 50px; font-weight: bold; ">&nbsp;&nbsp;&nbsp;{ { request.user.teacher.name } }</h5>
<a class="dropdown-item" href="{ % url 'change_password' % }">Change password</a>
<a class="dropdown-item" href="{ % url 'logout' % }">Logout</a></div></button>
</li>
</ul>
</nav>

<!--body-->
<style>
.submit-1
{
background-color:#fff3e6;
color:#793702;
transition:0.15s;
padding-bottom: 0px;
padding-top:3px;
padding-right: 13px;
}
.submit-1:hover
{
background-color:#ffbf5f;
transition:0.15s;
}
.list-group-item
{
background-color: transparent;
border:0px solid #793702;
border-radius: 0;
color:#793702;
}
.b1
{
color:#fff3e6;
background-color: #793702;
```

```
        }
    .b1:hover
    {
        background-color: #ffbf5f;
    }

table.table-questions
{
    border:3px solid #793702;
}
table.table-questions > tbody > tr
{
    border:2px solid #793702;
}
table.table-questions > tbody > tr > td
{
    border:2px solid #793702;
}
table.table-questions > tbody > tr > th
{
    border:2px solid #793702;
}
.table-grade
{
    width:100%;
    border:3px solid #793702;
}
table.table-grade > tbody > tr
{
    border:2px solid #793702;
    color: color:yellowgreen;
}
table.table-grade > tbody > tr > td
{
    border:2px solid #793702;
}
</style>
<div class="block" style=" background-image: url(Untitled.png); background-size:cover; margin-top:80px;">
    <div class="row ml-0 mr-0">
        <div class="col-3" style="height:88vh; text-align:center; border-left:5px solid #793702;">
```

```

<BR><BR><BR><br><BR><BR>
<h2 style="color:#793702; font-weight:bold;">Which Paper ?</h2>
<form method=POST action="" name=tbc>
    { % csrf_token %}
    <select name="paper" id="paper" required style="font-size:25px;">
        <option value="none" selected>----</option>
    { % for x in papers %}
        <option value="{{x.id}}>{{x.name}}</option>
    { % endfor %}
    </select><br><br>
    <button type="submit" class="btn btn-none b1" >View marks</button>
</form>
</div>
<div class="col-9" style=" height:88vh; text-align: center; border-left:5px solid #793702; border-right:5px solid #793702;">
    <div id=subds>
        <br><br>
        <table class="table table-grade table-sm m-auto" style=" color:brown; font-size:17px; font-family:Modern No. 20; font-weight:bold;">
            <tr>
                <td>
                    Code
                </td>
                <td style="color:black">
                    {{paper.code}}
                </td>
                <td>
                    Name
                </td>
                <td style="color:black">
                    {{paper.name}}
                </td>
            </tr>
            <tr>
                <td>
                    Number of Questions
                </td>
                <td style="color:black">
                    {{paper.count}}
                </td>
                <td >

```

```
Department
</td>
<td style="color:black">
  {{paper.dept.name}}
</td>
</tr>
<tr>
<td>
  Marks per correct question
</td>
<td style="color:black">
  {{paper.marks_per_cq}}
</td> <td>
  Marks per wrong question
</td>
<td style="color:black">
  {{paper.marks_per_wq}}
</td>
</tr>
<tr>
<td>
  Full Marks
</td>
<td style="color:black">
  {{paper.fm}}
</td>
<td>
  Modified Full Marks
</td>
<td style="color:black">
  {{paper.modified_fm}}
</td>
</tr>
<tr>
<td>
  Paper Setter
</td>
<td colspan=3 style="color:black">
  {{paper.paper_setter.name}}
</td>
</tr>
```

```
</table>
<br><br>
<table class="table table-questions m-auto">
<tr style="color:brown;">
<th>Roll</th>
<th>Name</th>
<th>OM</th>
<th>Modified OM</th>
<th>Grade</th>
</tr>
{ % for x in marks % }
<tr>
<td>{{ x.student.roll }}</td>
<td>{{ x.student.name }}</td>
<td>{{ x.om }}</td>
<td>{{ x.modified_om }}</td>
<td>{{ x.grade }}</td>
</tr>
{ % endfor %}
</table>
</div>
</div>
</div>
</div>
```

```
<!--footer-->
<style>
.b/*footer*/
{
position:fixed;
bottom:0;
width:100%;
background-color: #793702; color:#fff3e6;
}
.footer
{
width:20%;
```

```
    margin-left:auto;
}
</style>
<div class="b">
<div trademark style="text-align:center; ">
    © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
</div>
</div>
```

```
<!--javascript-->
<body>
<script type='text/javascript'>
    var no=Number(no);
    var y=document.getElementsByTagName("option")[ document.getElementById("paper").selectedIndex].value;
    console.log(y);
    console.log("yoyoy");
</script>
</body>
</html>
```

**For the teacher to unlock paper and publish marks of students
(unlock_papers_and_marks.html)**

```
<!doctype html>
<html lang="en">
{ %load static% }
{ % static 'images' as baseUrl % }

<!--Linking bootstrap online-->
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIJK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-ha384-"
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
</head>
<style>
body
{
    align-content:center;
    background-color: #fff3e6;
    height: 70vh;
    background-image: url({ % static 'images/Untitled.png' % });
    background-size:cover;
}
</style>
```

```
<!--Navigation Bar-->
<style>
.c
{
  color:#ffcc80;
  transition: 0.15s;

}
.c:hover
{
  color:#fcdb9;
  transition: 0.15s;
  text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
  background-color:#fff3e6 ;
}
.dropdown-item
{
  font-weight: 500;
  color:#793702;

}
.dropdown-item:hover
{
  background-color:#ffcc80;
  transition: 0.15s;
}
.button-1
{
  background-color:#fff3e6;
  color:white;
  transition:0.15s;
  border-radius: 50%;
  padding:20px;
```

```

        }
.button-1:hover
{
    background-color:#ffbf5f;
    transition:0.15s;
}
h1, h4
{
    font-size:40px;
    font-weight:bold;
    color:rgb(255, 255, 255);
    text-shadow: 4px 0px 5px rgb(11, 13, 112);
    letter-spacing: 4px;
}
.dropdown-menu
{
    transition: 0.5s;
}
.dropdown-menu:active
{
    transition: 0.5s;
}
.dropdown-menu
{
    padding: 0px;
    border:1px solid grey;
}
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
    <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
        <ul class="navbar-nav">
            <li class="nav-item ">
                <a class="nav-link pl-4" href="{% url 'teacher_dashboard' %}"><div class=c><h5>Home</h5></div></a>

```

```

</li>
</ul>
<ul class="navbar-nav">
  <li class="nav-item ">
    <a class="nav-link pl-
4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5></div></
a>
  </li>
</ul>
<ul class="navbar-nav">
  <li class="nav-item ">
    <a class="nav-link pl-
4" href="{% url 'marks_for_teacher' %}"><div class=c><h5>Student marks</h5></div></a>
  </li>
</ul>
<ul class="navbar-nav">
  <li class="nav-item ">
    <a class="nav-link pl-
4" href="{% url 'unlock_papers_and_marks' %}"><div class=c><h5>Unlock paper / Publish M
arks</div> </a>
  </li>
</ul>
</div>
<ul class="navbar-nav">
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false" style="color:#793702;"></a>
      <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
        <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold; ">&nbsp;&
nbsp;&nbsp;&nbsp;{{ request.user.teacher.name }}</h5>
        <a class="dropdown-item" href="{% url 'change_password' %}">Change password</a>
        <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></div></button>
      </li>
    </ul>
  </nav>

```

```
<body>
<style>

    /* width */
::webkit-scrollbar {
    width: 10px;
}

/* Track */
::-webkit-scrollbar-track {
    background: transparent;
    opacity: 0;
}

/* Handle */
::-webkit-scrollbar-thumb {
    background: #ffbf5f;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
    background: #555;
}

.b1
{
    color: #fff3e6;
    background-color: #793702;
}

.b1:hover
{
    background-color: #ffbf5f;
}

.table-questions
{
    border: 3px solid brown
}
table.table-questions > tbody > tr
{
    border: 2px solid brown;
```



```
Exam Going On<!--backend-->
{ % endif % }<!--backend-->
{ % else % } Exam has not held yet
{ %endif% }<!--backend-->
</div>
</div></td>
</tr>
{ % endfor % }<!--backend-->
</table>
<br><div class="po text-right ">
<input type="reset" class="btn btn-none b1" value="Reset" /><!--backend-->
<input type="submit" class="btn btn-none b1" value="Confirm" /></div><!--backend-
->
</form>
</div>
</div>
</div>
</div>
</div>

<!--footer-->
<style>
.b/*footer*/
{
position:fixed;
bottom:0;
width:100%;
background-color: #793702; color:#fff3e6;
}
.footer
{
width:20%;
margin-left:auto;
}
</style>
<div class="b">
<div trademark style="text-align:center; ">
    © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
</div>
```

```
</div>
</body>
</html>
```

Student

Student Dashboard (student_dashboard.html)

```

{%-load static%}
{% static 'images' as baseUrl %}

<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
      <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtlkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
      <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
      <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQxSfFWpi1MquVdAyjUar5+76PVCmYI" crossorigin="anonymous"></script>
    <title>Hello, world!</title>
  </head>

  <style>
    body
    {
      align-content:center;
      border: 0px;
      background-image: url("{% static 'images/untitled1.png' %}");
      background-size:cover;
    }
  </style>

```

```
<!--Navigation Bar-->
<style>
.c
{
    color:rgba(10, 70, 15, 0.705);
    transition: 0.15s;

}
.c:hover
{
    color:white;
    transition: 0.15s;
    text-shadow: 2px 0px 2px black;
}
.dropdown-menu
{
    background-color:white;
}
.dropdown-item
{
    font-weight: 500;
    color:black;

}
.dropdown-item:hover
{
    background-image: linear-gradient(to right,white,skyblue);
    transition: 0.15s;
}
.button-1
{
    background-color:#fff3e6;
    color:white;
    transition:0.15s;
    border-radius: 50% ;
    padding:20px;
}
.button-1:hover
```

```

{
  background-color:#ffbf5f;
  transition:0.15s;
}
h1, h4
{
  font-size:40px;
  font-weight:bold;
  color:rgb(255, 255, 255);
  text-shadow: 4px 0px 5px rgb(11, 13, 112);
  letter-spacing: 4px;
}
.dropdown-menu
{
  transition: 0.5s;
}
.dropdown-menu:active
{
  transition: 0.5s;
}
.dropdown-menu {
  padding: 0px;
  border:1px solid grey;
}
</style>

```

```

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-
COLOR:lightgreen; ">
  <a class="navbar-brand pl-5" style="white"><h4>Extranet</h4></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link pl-
4" href="{% url 'student_dashboard' %}"><div class=c><h5>Home</h5></div></a>
        </li>
    </ul>

```

```

{ % if published_papers %}
<ul class="navbar-nav">
<li class="nav-item ">

    <a class="nav-link pl-
4" href="{% url 'marks_for_student' published_papers.0.id %}"><div class=c><h5>Past Exam
Results</h5></div></a>
    </li>
</ul>
{ % endif %}
</div>
<ul class="navbar-nav">
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-
toggle="dropdown" aria-haspopup="true" aria-
expanded="false" STYLE="color:#793702;"></a>
        <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
            <h5 class="dropdown-header" style="color:; background-
color: lightgreen; padding: 20px 50px; font-
weight: bold;">&nbsp;&nbs
p;&nbsp;&nbsp;{ {request.user.student.name}}</h5>
            <a class="dropdown-
item" href="{% url 'change_password' %}">Change password</a>
                <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></button>
            </div>
        </li>
</ul>
</nav>

<!--question details-->
<style>

/* width */
::-webkit-scrollbar {
    width: 10px;
}

```

```
/* Track */
::-webkit-scrollbar-track {
    background:skyblue;
    opacity:0;
}

/* Handle */
::-webkit-scrollbar-thumb {
    background:green;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
    background: yellowgreen;
}

.b1
{
    color:white;
    background-color: blue;
}
.b1 , .btn-success, .btn-danger
{
    width:100px;
}

.b1:hover
{
    transition: 0.15s;

    background-color:#020AC4;
}
.table-questions
{
    border:3px solid darkblue;
}
table.table-questions > tbody > tr
{
    border:2px solid darkblue;
}
</style>
```

```

<div class="block" style="margin-top:80px;">
  <Div class="row ml-0 mr-0" style="">
    <div class="col-4" style="overflow: auto;">
      
    </div>
    <div class="col-8" style="overflow: auto;">

      <!--Welcome text-->
      <div class="row">
        <div class="col-md-12 text-right p-2" style="color:blue; font-weight:bold; font-family:Modern No. 20; font-size: 30px; text-shadow: 2px 2px 5px grey;">
          Welcome, {{request.user.student.name}}
        </div>
      </div>

      <!--Available Papers-->
      <div class="row">
        <div class="col-12 p-4" >
          <p style="color:blueviolet; font-weight:none; font-family:Modern No. 20; font-size:50px; letter-spacing:3px; text-decoration: none; text-shadow: 0px 0px black;">Available papers</p>
          <table class="table table-questions m-auto" style="color:blue; font-family:Modern No. 20; text-decoration:none; text-shadow: 0px 0px grey;">
            {% for x in resume_papers %}
              <tr>
                <td>

                  <div class="row">
                    <div class="col text-info" style="font-size: 23px;">{{x}}</a></div>
                    <div class="col text-right"><a href="{% url 'give_exam' x.id %}"><button type="button" class="btn btn-success">Resume</button></a>
                  </div>

                </td>
              </tr>
            {% endfor %}
            {% for x in give_papers %}
              <tr>
                <td>

```



```
.b
{
    background-COLOR:lightgreen;;
    color:white;
    position:fixed;
    bottom:0;
    width:100% ;
}
.footer
{
    width:20%;
    margin-left:auto;
}
</style>
<div class="b">

    <div trademark style="text-align:center; color: rgb(6, 74, 201); ">
        © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
    </div>
</div>

</html>
```

Student's past exam results (marks_for_student.html)

```

{%load static%}
{% static 'images' as baseUrl %}

<!DOCTYPE html>
<html lang="en">
    <head>
        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
        <!-- Bootstrap CSS -->
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.
0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" crossorigin="an
onymous">
        <script src="https://code.jquery.com/jquery-
3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkYIJK3UENmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KKN" crossorigin="an
onymous"></script>
        <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/
popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="an
onymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap
.min.js" integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="an
onymous"></script>
        <title>Hello, world!</title>
    </head>

    <style>
        body
        {
            align-content:center;
            border: 0px;
            background-image: url("{%static 'images/untitled1.png'%}");
            background-size:cover;
        }
    </style>

```

```
<!--Navigation Bar-->
<style>
    .c
    {
        color:rgba(10, 70, 15, 0.705);
        transition: 0.15s;
    }
    .c:hover
    {
        color:white;
        transition: 0.15s;
        text-shadow: 2px 0px 2px black;
    }
    .dropdown-menu
    {
        background-color:white;
    }
    .dropdown-item
    {
        font-weight: 500;
        color:black;

    }
    .dropdown-item:hover
    {
        background-image: linear-gradient(to right,white,skyblue);
        transition: 0.15s;
    }
    .button-1
    {
        background-color:#fff3e6;
        color:white;
        transition:0.15s;
        border-radius: 50%;
        padding:20px;
    }
    .button-1:hover
    {
        background-color:#ffbf5f;
        transition:0.15s;
    }
</style>
```

```

        }
    h1, h4
    {
        font-size:40px;
        font-weight:bold;
        color:rgb(255, 255, 255);
        text-shadow: 4px 0px 5px rgb(11, 13, 112);
        letter-spacing: 4px;
    }

    .dropdown-menu
    {
        transition: 0.5s;
    }
    .dropdown-menu:active
    {
        transition: 0.5s;
    }
    .dropdown-menu
    {
        padding: 0px;
        border:1px solid grey;
    }
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-COLOR:lightgreen; ">
    <a class="navbar-brand pl-5" style="white"><h4>Extranet</h4></a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar">
        <ul class="navbar-nav">
            <li class="nav-item ">
                <a class="nav-link pl-4" href="{% url 'student_dashboard' %}"><div class=c><h5>Home</h5></div></a>
            </li>
        </ul>
        {% if published_papers %}
        <ul class="navbar-nav">

```

```

        <li class="nav-item ">
            <a class="nav-link pl-
4" href="{% url 'marks_for_student' published_papers.0.id %}"><div class=c><h5>P
ast Exam Results</h5></div></a>
        </li>
    </ul>
    {% endif %}
</div>
<ul class="navbar-nav">
    <li class="nav-item dropdown">
        <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-
haspopup="true" aria-
expanded="false" style="color:#793702;"></a>
            <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
                <h5 class="dropdown-header" style="color:; background-
color: lightgreen; padding: 20px 50px; font-
weight: bold;">&
nbsp;&ampnbsp&ampnbsp{{request.user.student.name}}</h5>
                <a class="dropdown-
item" href="{% url 'change_password' %}">Change password</a>
                <a class="dropdown-
item" href="{% url 'logout' %}">Logout</a></div></button>
            </li>
        </ul>
    </nav>

<!--Available Result of previously given Papers-->
<style>

/* width */
::-webkit-scrollbar {
width: 10px;
}

/* Track */
::-webkit-scrollbar-track {
background:lightgreen;
}

```

```
    opacity:0;
}

/* Handle */
::-webkit-scrollbar-thumb {
background:darkolivegreen;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover {
background: #555;
}
.b1
{
    color:white;
    background-color: blue;
}
.b1 , .btn-success, .btn-danger
{
    width:100px;
}
.b1:hover
{
    transition: 0.15s;
    background-color:#020AC4;
}
.table-questions
{
    border:3px solid darkblue;
}
table.table-questions  > tbody > tr
{
    border:2px solid darkblue;
}
table.table-questions  > tbody > tr > td
{
    border:2px solid darkblue;
    color: darkgreen;
}
.table-grade
{
```

```

        width:100%;
        border:1px solid darkblue;
    }
    table.table-grade > tbody > tr
    {
        border:1px solid darkblue;

    }
    table.table-grade > tbody > tr > td
    {
        border:1px solid darkblue;
    }
</style>
<div class=block Style="margin-top:80px; height:88vh;">
    <Div class="row ml-0 mr-0" style="">
        <div class="col-4" Style="height:88vh; overflow: auto; ">
            <div class=row >
                <div class="col-12 p-4" >
                    <p style="color:blueviolet;; font-weight:bold; font-
family:Modern No. 20; font-size: 30px; text-
shadow: 2px 2px 5px grey;">Exam papers</p>
                    <table class="table table-questions m-
auto" style="color:white; font-family:Modern No. 20; text-decoration:none; text-
shadow: 2px 0px grey;">
                        {% for x in published_papers%}
                        <tr>
                            <td>
                                <a href="{% url 'marks_for_student' x.id %}" 
STYLE="color:BLUE;  font-family:Modern No. 20; font-size: 15px; ">{{x}}</a>
                            </td>
                        </tr>
                        {% endfor %}
                    </table>
                </div>
            </div>
        </div>
    </div>

    <div class="col-8" style="overflow: auto; text-
align:center; color:rgb(255, 0, 0); font-family:Modern No. 20; font-
size: 20px; "><!--height:77vh-->
        <div class=row >

```

```
<div class="col-12 p-4" >
    <div id=res style="color:blue; font-weight:bold; font-
family:Modern No. 20; font-size: 30px; text-
shadow: 1px 1px 1px grey;">Results of {{paper}}</div>
    <div id=results style="color:darkgreen; font-
weight:bold; font-family:Modern No. 20; font-size: 25px;">
        Grade:<a style="color:blue"> {{m.grade}} </a>      <br>
        Obtained Marks: <a style="color:blue">{{m.om}}</a><br>
        Modified Obtained Marks:<a style="color:blue"> {{m.modifi
ed_om}}</a><br><br>
    </div>
    <div id=res style="color:blue; font-weight:bold; font-
family:Modern No. 20; font-size: 22px; text-
shadow: 1px 1px 1px grey;">{{paper}} Details</div>
    <table class="table table-grade table-sm m-
auto" style=" color:darkgreen; font-size:17px; font-family:Modern No. 20; font-
weight:bold;">
        <tr>
            <td>
                Code
            </td>
            <td style="color:blue">
                {{paper.code}}
            </td>
            <td>
                Name
            </td>
            <td style="color:blue">
                {{paper.name}}
            </td>
        </tr>
        <tr>
            <td>
                Number of Questions
            </td>
            <td style="color:blue">
                {{paper.count}}
            </td>
            <td >
                Department
            </td>
        </tr>
    </table>
</div>
```

```
<td style="color:blue">
    {{paper.dept.name}}
</td>
</tr>
<tr>
    <td>
        Marks per correct question
    </td>
    <td style="color:blue">
        {{paper.marks_per_cq}}
    </td> <td>
        Marks per wrong question
    </td>
    <td style="color:blue">
        {{paper.marks_per_wq}}
    </td>
    </tr>
</tr>
<tr>
    <td>
        Full Marks
    </td>
    <td style="color:blue">
        {{paper.fm}}
    </td>
    <td>
        Modified Full Marks
    </td>
    <td style="color:blue">
        {{paper.modified_fm}}
    </td>
</tr>
<tr>
    <td>
        Paper Setter
    </td>
    <td style="color:blue">
        {{paper.paper_setter.name}}
    </td>
</tr>
</table>
```

```
        </div>
    </div>

    <div class=row>
        <div class="col-12 p-4">
            <h3 style="color:blue; font-weight:bold; font-family:Modern No. 20; font-size: 22px; text-shadow: 1px 1px 1px grey;">Grade System</h3>
                <table class="table table-sm table-grade m-auto" style="color:BLUE; font-family:Modern No. 20; text-decoration:none; ">
                    <tr style="color:darkgreen; font-weight:bold;">
                        <td>
                            A
                        </td>
                        <td>
                            B
                        </td>
                        <td>
                            C
                        </td>
                        <td>
                            D
                        </td>
                        <td>
                            E
                        </td>
                        <td>
                            F
                        </td>
                    </tr>
                    <tr>
                        <td>
                            90% - 100%
                        </td>
                        <td>
                            80% - 90%
                        </td>
                        <td>
                            70% - 80%
                        </td>
                        <td>
```

```
    60% - 70%
  </td>
<td>
  50% - 60%
</td>
<td>
  < 50%
</td>
</tr>
</table>
<br /><br />
</div>
</div>
</Div>
</div>

<!--footer-->
<style>
.b
{
background-COLOR:lightgreen;;
color:white;
position:fixed;
bottom:0;
width:100%;
}
.footer
{
width:20%;
margin-left:auto;
}
</style>
<div class="b">

<div trademark style="text-align:center; color: rgb(6, 74, 201); ">
  © 2021 ExtraNET.com are registered trademarks of SXC. All rights rese
rved.
</div>
</div>
</html>
```

Examination page (ajax_exam.html)

```
{%load static%}

<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" cross
origin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UEZzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" cro
ssorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integri
ty="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorig
in="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="s
ha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorig
in="anonymous"></script>
    <title>Hello, world!</title>
  </head>

  <style>
    body
    {
      align-content:center;
      border: 0px;
      background-size:cover;
    }
    #view
    {
      -webkit-user-select: none;
      -ms-user-select: none;
      user-select: none;
    }
  </style>
```

```
        }
    </style>
<!--Navigation Bar-->
<style>
.c
{
    color:rgba(10, 70, 15, 0.705);
    transition: 0.15s;
}
.c:hover
{
    color:white;
    transition: 0.15s;
    text-shadow: 2px 0px 2px black;
}
.dropdown-menu
{
    background-image: linear-gradient(to bottom right, #004b05, #00ff2f);
}
.dropdown-item
{
    font-weight: 500;
    color:black;
}
.dropdown-item:hover
{
    background-color:yellowgreen;
    transition: 0.15s;
}
.button-1
{
    background-color:#fff3e6;
    color:white;
    transition:0.15s;
    border-radius: 50%;
    padding:20px;
}
.button-1:hover
{
    background-color:#ffbf5f;
    transition:0.15s;
```

```

        }
    h1, h3
    {
        font-size:60px;
        font-weight:bold;
        color:rgb(255, 255, 255);
        text-shadow: 4px 0px 5px rgb(11, 13, 112);
    }

.dropdown-menu
{
    transition: 0.5s;
}
.dropdown-menu:active
{
    transition: 0.5s;
}
.dropdown-menu
{
    padding: 0px;
    border:1px solid grey;
}
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-
COLOR:SKYBLUE; ">
    <a class="navbar-brand pl-2" style="font-size:35px; font-
weight:bold; color:rgb(255, 255, 255); text-shadow: 4px 0px 5px blue;">Extranet</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="collapsibleNavbar"></div>
    <ul class="navbar-nav">
        <div id=timer style="font-weight:bold; color:yellow; font-size:20px;"></div>
    </ul>
</nav>

<body onload="autoSubmit(){% if attempted_answers %}; fillAnswers();{% endif %}">
    <style>
```

```
.submit-1
{
    background-color:#fff3e6;
    color:#793702;
    transition:0.15s;
    padding-bottom: 0px;
    padding-top:3px;
    padding-right: 13px;
}
.submit-1:hover
{
    background-color:#ffbf5f;
    transition:0.15s;
}
.list-group-item
{
    background-color: transparent;
    border:0px solid #793702;
    border-radius: 0;
    color:#793702;
}
.b1
{
    color:#fff3e6;
    background-color: #793702;
}
.b1:hover
{
    background-color: #ffbf5f;
}
.table-questions
{
    border:3px solid brown
}
table.table-questions > tbody > tr
{
    border:2px solid brown;
}
</style>
```

```
<!--question-->
<div class="block" style=" background-image: url(Untitled.png); background-size:cover;">
<div class="row ml-0 mr-0" >
<div class="col-9" style="height:100vh; border-left:5px solid skyblue ;">
<!-- <body onload="autoSubmit()" -->
<br><br><br> <br>
<div class="row ml-0 mr-0" >
<div class="col-8">
<form action= method=POST>
{ % csrf_token % }
<div id="view" style="visibility: hidden;">
</div>
</form>
</div>
<div class="col-4">
<p id="mssg1" style="visibility: visible;">Click on the <a style="color:BLSCK; font-weight:bold;">Go to exam</a> button to give exam.<a style="color:red; font-weight:bold;"><br>Try not to exit fullscreen once in.<br>Repeated exiting will lead to suspension.</a></p>
<button type="button" class="btn btn-info" id="toggle" onclick="openFullscreen()" >
    Go to exam
</button>
</div>
</div>
<!-- </body> -->
</div>
<div class="col-3" style=" border-left:5px solid skyblue; border-right:5px solid skyblue;">
<br><br><br><br><br>
<p id="switch" style="visibility: hidden;"></p>
<div>
<p id="mssg" style="visibility: hidden;"></p>
</div>
</div>
</div>
</div>
```

```
<!--footer-->
<style>
.b
{
    text-align:right;
    color:white;

    position:fixed;
    bottom:0;
    width:100%;

}
</style>
<div class="b" >
    <div class=sub style="padding-right: 10px; padding-bottom:5px"><input type="button" class="btn btn-primary" onclick="submit_form()" value=Submit "></div>
    <div trademark style="text-align:center; background-COLOR:SKYBLUE;">
        © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved.
    </div>
</div>

<script src="{ % static 'js/jquery.js' % }"></script>

<script type="text/javascript">
var gfg = document.documentElement;
var warning = { {warning} } ;
let timer_left_time=0;
var t=0, h=0, m=0,s=0;
let ans_list = { };
//let ans_list1 = { };
let csr = document.getElementsByName("csrfmiddlewaretoken")[0].value;
ans_list.csrfmiddlewaretoken = csr;

let count = 1;
var no = { { arrL } };
```

```

var x = "<div class=\"container-fluid\"> <div class=\"row\">";
var results = JSON.parse(JSON.parse('{{ arr | escapejs }}'));
console.log(results);
console.log(ans_list);
while(count<=no)
{
  x=x+' <div class="col-
3"> <input type=button id='"+count+"' value='"+count+"' onclick=showQuestions('+count+') style="
font-size:25px; background-color:#ECECEC;"></div>';
  count++;
}
x=x+'</div></div>';
document.getElementById("mssg").innerHTML=x;

var output = document.getElementById("mssg1");
document.addEventListener("fullscreenchange", function() {
//output.innerHTML = "A computer science portal for geeks";
if (getFullelement()) {
  console.log(`entered full-screen mode.`);
  document.getElementById("view").style.visibility = "visible";
  document.getElementById("mssg").style.visibility = "visible";
  document.getElementById("toggle").style.visibility = "hidden";
  document.getElementById("mssg1").style.visibility = "hidden";
//output.innerHTML = "full screen";
} else {
  console.log('Leaving full-screen mode.');
  warning--;
  output.innerHTML = 'Click on the <a style="color:BLSCK; font-
weight:bold;">Go to exam</a> button to go back to the exam.<a style="color:red; font-
weight:bold;"><br>Try not to exit fullscreen once in.<br>Repeated exiting will lead to suspensio
n.</a></p>';
  document.getElementById("view").style.visibility = "hidden";
  document.getElementById("mssg").style.visibility = "hidden";

  if (warning>0)
  {
    document.getElementById("toggle").style.visibility = "visible";
    document.getElementById("mssg1").style.visibility = "visible";
    document.getElementById("switch").style.visibility = "visible";
    document.getElementById("switch").innerHTML= "<span style=\"color:red; font-
weight:bold\">Warning:</span> <span style=\"color:blue;\">A new window has been opened or

```

a pop has occurred. Doing it for " + warning + " more time[s] will get you disqualified from the exam.";

```

    }
    }
});

var view = ""

document.getElementById("view").innerHTML=view;
console.log(results[0].pk)

function showQuestions(no)
{
    var no = Number(no);
    //console.log(no);
    var ques = results[(no-1)].fields.op1
    //console.log(ques);
    checked_option = ans_list[results[(no-1)].pk];
    console.log('anslist from show')
    console.log(ans_list)

    switch(checked_option) {
        case 'a':
            {var view = '<p style="font-size:20px;"><a style="font-size:30px; font-
weight:bold; color:blue; text-decoration:underline;">Question: '+ no+'<br></a><a style="font-
size:20px; font-weight:bold; color:black;">'+results[(no-1)].fields.desc+'</a><br>';
            if((results[(no-1)].fields.img_path).length!=0)
                view+='<br>
                //'}

            view+='<br><a style="font-size:20px; font-
weight:bold; color:blue;">Choose options:</a> '+ '<br><br /><input type=radio class=ans oncli-
ck=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-
1)].pk+' value=a checked/>&nbsp'+results[(no-1)].fields.op1+
}
}
});
```

```

'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=b />&nbsp'+results[(no-1)].fields.op2+
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=c />&nbsp'+results[(no-1)].fields.op3+
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=d />&nbsp'+results[(no-1)].fields.op4+
'<br /><br /> <input type=button class="btn btn-secondary" value=Clear onclick=clearSelection('+results[(no-1)].pk+', '+no+') /> <input type=button class="btn btn-success" onclick=showQuestions('+(no+1)+') value=Next /> <br /><br /></p>';
document.getElementById(no).style.backgroundColor="green";
}
break;
case 'b':
var view = '<p><a style="font-size:30px; font-weight:bold; color:blue; text-decoration:underline;">Question: '+ no+'<br></a><a style="font-size:20px; font-weight:bold; color:black;">'+results[(no-1)].fields.desc+'</a><br>' 
if((results[(no-1)].fields.img_path.length!=0)
view+='<br>' 

view+='<br><a style="font-size:20px; font-weight:bold; color:blue;">Choose options:</a> '+ '<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=a />&nbsp'+results[(no-1)].fields.op1+
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=b checked/>&nbsp'+results[(no-1)].fields.op2+
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=c />&nbsp'+results[(no-1)].fields.op3+
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=d />&nbsp'+results[(no-1)].fields.op4+
'<br /><br /> <input type=button class="btn btn-secondary" value=Clear onclick=clearSelection('+results[(no-1)].pk+', '+no+') /> <input type=button class="btn btn-success" onclick=showQuestions('+(no+1)+') value=Next /> <br /><br />'; document.getElementById(no).style.backgroundColor="green";}
break;
case 'c':

```

```

{ var view = '<p><a style="font-size:30px; font-weight:bold; color:blue; text-decoration:underline;">Question: '+ no+'<br></a><a style="font-size:20px; font-weight:bold; color:black;">' + results[(no-1)].fields.desc + '</a><br>'  

if((results[(no-1)].fields.img_path).length!=0)  

view += '<br>'  

view += '<br><a style="font-size:20px; font-weight:bold; color:blue;;">Choose options:</a> '+  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=a />&nbsp'+results[(no-1)].fields.op1 +  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=b />&nbsp'+results[(no-1)].fields.op2 +  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=c checked/>&nbsp'+results[(no-1)].fields.op3 +  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=d />&nbsp'+results[(no-1)].fields.op4 +  

'<br /><br /><input type=button class="btn btn-secondary" value=Clear onclick=clearSelection(' + results[(no-1)].pk + ', '+ no +') /> <input type=button class="btn btn-success" onclick=showQuestions(' +(no+1)+ ') value=Next /> <br /><br />'; document.getElementById(no).style.backgroundColor="green";}  

break;  

case 'd':  

{ var view = '<p><a style="font-size:30px; font-weight:bold; color:blue; text-decoration:underline;">Question: '+ no+'<br></a><a style="font-size:20px; font-weight:bold; color:black;">' + results[(no-1)].fields.desc + '</a><br>'  

if((results[(no-1)].fields.img_path).length!=0)  

view += '<br>'  

view += '<br><a style="font-size:20px; font-weight:bold; color:blue;;">Choose options:</a> '+  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=a />&nbsp'+results[(no-1)].fields.op1 +  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=b />&nbsp'+results[(no-1)].fields.op2 +  

'<br><br /><input type=radio class=ans onclick=radioActivity(' + results[(no-1)].pk + ', '+ no +') name=' + results[(no-1)].pk + ' value=c />&nbsp'+results[(no-1)].fields.op3 +

```

```

'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=d checked/>&nbsp'+results[(no-1)].fields.op4+
' <br /><br /> <input type=button class="btn btn-secondary" value=Clear onclick=clearSelection('+results[(no-1)].pk+', '+no+') /> <input type=button class="btn btn-success" onclick=showQuestions('+(no+1)+') value=Next /> <br /><br />'; document.getElementById(no).style.backgroundColor="green";}
break;
default:
{var view = '<p><a style="font-size:30px; font-weight:bold; color:blue; text-decoration:underline;">Question: '+ no+'<br></a><a style="font-size:20px; font-weight:bold; color:black;">' + results[(no-1)].fields.desc + '</a><br>' +
if((results[(no-1)].fields.img_path).length!=0)
view+= '<br>' +
view+= '<br><a style="font-size:20px; font-weight:bold; color:blue;;">Choose options:</a> +' +
'<br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=a />' + results[(no-1)].fields.op1+
' <br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=b />' + results[(no-1)].fields.op2+
' <br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=c />' + results[(no-1)].fields.op3+
' <br><br /><input type=radio class=ans onclick=radioActivity('+results[(no-1)].pk+', '+no+') name='+results[(no-1)].pk+' value=d />' + results[(no-1)].fields.op4+
' <br /> <br /><input type=button class="btn btn-secondary" value=Clear onclick=clearSelection('+results[(no-1)].pk+', '+no+') /> <input type=button class="btn btn-success" onclick=showQuestions('+(no+1)+') value=Next /> <br /><br />';document.getElementById(no).style.backgroundColor="#ECECEC";}
}
document.getElementById("view").innerHTML=view;
}

function pause() {
console.log("lost focus");
warning--;
}

```

```

function play() {
    console.log("focus");
    if (warning>0)
    {
        document.getElementById("switch").style.visibility = "visible";
        document.getElementById("switch").innerHTML= "<span style=\"color:red; font-weight:bold\">Warning:</span> <span style=\"color:blue;\">A new window has been opened or a pop has occured. Doing it for " + warning + " more time[s] will get you disqualified from the exam.</span>";
    }
    else
    {
        document.getElementById("view").style.visibility = "hidden";
        document.getElementById("mssg").style.visibility = "hidden";
        document.getElementById("toggle").style.visibility = "hidden";
        document.getElementById("mssg1").style.visibility = "hidden";
        document.getElementById("switch").style.visibility = "visible";
        document.getElementById("switch").innerHTML= "disqualified";
        submit_form();
    }
}

window.addEventListener('blur', pause);
window.addEventListener('focus', play);

function autoSubmit(){
    let server_left_time={server_left_time};
    timer_left_time={timer_left_time};
    let left_time=server_left_time<timer_left_time?server_left_time:timer_left_time;

    setTimeout(submit_form, (left_time-1)*1000);

    setInterval(() => {
        timer_left_time--;
        //time coversion into hh:mm:ss
        t=timer_left_time;
        h=(t/3600);
        m=((h-Math.floor(h))*60);
        s=t%60;
    })
}

```

```

        document.getElementById("timer").innerHTML='<a style="color:blue">Time left : </a>'  

+Math.floor(h)+'&nbsp:&nbsp'+Math.floor(m)+'&nbsp:&nbsp'+Math.floor(s);  

    }, 900);  

    setInterval(save_form, 5000);  

}  
  

function submit_form()  

{  

$.ajax({  

    url: "{% url 'give_exam' paper.id 'dummy' % }",  

    method: "POST",  

    data: ans_list,  

});  
  

setTimeout(() => {  

window.location.replace("{% url 'student_dashboard' % }");  

}, 200);  
  

}  
  

function fillAnswers() {  

let ans_list_temp=JSON.parse(JSON.parse('{ { attempted_answers | escapejs } }'));  

for(x in ans_list_temp){  

    ans_list[ans_list_temp[x].fields.question]=ans_list_temp[x].fields.selected_option  

}  
  

console.log('anslist from fillans')  

console.log(ans_list)  
  

/*for(let q_id in answers)  

{  

    let q = document.getElementsByName(q_id);  

//let selectedOption="";  

let index=answers[q_id].charCodeAt(0)-97;  

    q[index].checked=true;  

}*/  

}  
  

function radioActivity(q_id,serial_no) {
```

```

let q = document.getElementsByName(q_id);
//let selectedOption="";
for (let i = 0; i < q.length; i++) {
  if (q[i].checked) {
    ans_list[q_id] = q[i].value;
    //ans_list1[q_id] = q[i].value;
    console.log('ans_list from radio ac')
    console.log(ans_list);
    document.getElementById(serial_no).style.backgroundColor="green";
    break;
  }
}
//save_option(q_id);
}

function clearSelection(q_id,serial_no) {

//console.log('clearsel')

let q = document.getElementsByName(q_id);
for (let i = 0; i < q.length; i++)
  if (q[i].checked) {
    q[i].checked = false;
    document.getElementById(serial_no).style.backgroundColor="#ECECEC";
    break;
  }
  console.log(q_id)
  ans_list[q_id] = "u";
  console.log(ans_list)
}

function save_form() {
//let temp_ans_list = JSON.parse(JSON.stringify(ans_list));
ans_list.timer_left_time=timer_left_time;
ans_list.warning=warning;
$.ajax({
  url: "{% url 'ajax_exam' paper.id %}",
  method: "POST",
  data: ans_list,
});
}

```

```
}

document.addEventListener('contextmenu',function(e){
  e.preventDefault()
})

function getFullelement()
{
  return document.fullscreenElement
  || document.webkitFullscreenElement
  || document.mozFullscreenElement
  || document.msFullscreenElement;
}

function openFullscreen() {
  if (!getFullelement()) {
    gfg.requestFullscreen();
  }
  showQuestions(1);
}

</script>

</body>

</html>
```

Change Password (change_password.html)

```
<!doctype html>
<html lang="en">
  {%load static%}
  {% static 'images' as baseUrl %}

  <!--Linking bootstrap online-->
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
      <!-- Bootstrap CSS -->
      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
        <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkYIJK3UENmM7KCKR/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
          <title>Hello, world!</title>
    </head>
    <style>
      body
      {
        align-content:center;
        background-color: #fff3e6;
        height: 70vh; /*text-align:center;*/
        background-image: url({%static 'images/Untitled.png'%});
        background-size:cover;
      }
    </style>
```

```
<!--Navigation Bar-->
<style>
.c
{
    color:#ffcc80;
    transition: 0.15s;

}
.c:hover
{
    color:#fcdbcb;
    transition: 0.15s;
    text-shadow: 2px 0px 2px rgb(180, 114, 59);
}
.dropdown-menu
{
    background-color:#fff3e6 ;
}
.dropdown-item
{
    font-weight: 500;
    color:#793702;

}
.dropdown-item:hover
{
    background-color:#ffcc80;
    transition: 0.15s;
}
.button-1
{
    background-color:#fff3e6;
    color:white;
    transition:0.15s;
    border-radius: 50%;
```

```
padding:20px;
}
.button-1:hover
{
  background-color:#ffbf5f;
  transition:0.15s;
}
h1, h4
{
  font-size:40px;
  font-weight:bold;
  color:rgb(255, 255, 255);
  text-shadow: 4px 0px 5px rgb(11, 13, 112);
  letter-spacing: 4px;
}
.dropdown-menu
{
  transition: 0.5s;
}
.dropdown-menu:active
{
  transition: 0.5s;
}
.dropdown-menu
{
  padding: 0px;
  border:1px solid grey;
}
</style>

<nav class="navbar navbar-expand-lg navbar-light fixed-top" style="background-color: #793702;">
  <a class="navbar-brand pl-5" style="color: #ffcc80;"><h4>Extranet</h4></a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="collapsibleNavbar">
    <ul class="navbar-nav">
      <li class="nav-item ">
```

```

        <a class="nav-link pl-
4" href="{% url 'teacher_dashboard' %}" ><div class=c><h5>Home</h5></div></a>
        </li>
    </ul>
    <ul class="navbar-nav">
        <li class="nav-item ">
            <a class="nav-link pl-
4" href="{% url 'papers_and_questions' %}"><div class=c><h5>Question papers</h5>
</div></a>
            </li>
        </ul>
        <ul class="navbar-nav">
            <li class="nav-item ">
                <a class="nav-link pl-
4" href="{% url 'marks_for_teacher' %}" ><div class=c><h5>Student marks</h5></di
v></a>
                </li>
            </ul>
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link pl-
4" href="{% url 'unlock_papers_and_marks' %}"><div class=c><h5>Unlock paper / Pu
blish Marks</div> </a>
                    </li>
                </ul>
            </div>
            <ul class="navbar-nav">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-
toggle" href="#" id="navbarDropdownMenuLink" data-toggle="dropdown" aria-
haspopup="true" aria-
expanded="false" STYLE="color:#793702;"></a>
                    <div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdownMenuLink">
                        <h5 class="dropdown-header" style="color:#fff3e6; background-
color: #793702; padding: 20px 50px; font-
weight: bold; ">&nbsp;&nbsp;&nbsp;{{request.user.teacher.name}}</h5>
                        <a class="dropdown-
item" href="{% url 'change_password' %}">Change password</a>

```

```
        <a class="dropdown-item" href="{% url 'logout' %}">Logout</a></div></button>
      </li>
    </ul>
  </nav>

<style>
  label
  {
    font-weight:bold;
    color:rgb(99,61,29);
    letter-spacing: 2px;
    text-transform: uppercase;
  }
  input:hover,textarea:hover,select:hover
  {
    box-shadow: 0 2px 6px 0 rgba(99,61,29,0.3),0 7px 5px 0 rgba(99,61,29,0.3);
  }
  input:active
  {
    box-shadow: 0 2px 6px 0 rgba(99,61,29,0.3),0 7px 5px 0 rgba(99,61,29,0.3);
  }
  input
  {
    width: 100%;
    font-size:19px;
  }
  ::-webkit-scrollbar
  {
    width: 10px;
  }

  /* Track */
  ::-webkit-scrollbar-track
  {
    background:transparent;
    opacity:0;
  }
  /* Handle */
```

```
::-webkit-scrollbar-thumb
{
    background:#ffbf5f;
}

/* Handle on hover */
::-webkit-scrollbar-thumb:hover
{
    background: #555;
}
</style>
<div class="block" style="margin-top:95px; ">
    <div class="row ml-0 mr-0">
        <div class="col-sm" style="height:0vh; text-align:center; border-left:0px solid #793702;"><br><br><br><br><br><br><br><br><br><br></div>
        <div class="col-6" style=" padding-top: 20px;background-color: rgba(99,61,29,0.2); border-radius:10px; border: 1px solid rgb(99,61,29)">
            <div class="form-group">
                {% comment %}
                {% endcomment %}
                <form action="" method="POST" enctype="multipart/form-data">
                    {% csrf_token %} {{form.as_p}}<br />
                    {% for m in messages%} {{m}} {% endfor %}
                    <input class="btn btn-success" type="submit" value="Confirm Password" style="width: 200px"/>&nbsp;&nbsp;
                    <input class="btn btn-warning" type="reset" value="Refresh" style="width: 100px"/>
                </form>
            </div>
            </div>
            <div class="col-sm" style="height:0vh; text-align:center; border-right:0px solid #793702;"><br><br><br><br><br><br><br><br><br></div>
        </div>
    </div>
    <br> <br>
<!--footer-->
<style>
    .b
    {
        position:fixed;
```

```
bottom:0;
width:100%;
background-color: #793702; color:#fff3e6;
}
.footer
{
width:20%;
margin-left:auto;
}
</style>
<div class="b">
<div trademark style="text-align:center; ">
    © 2021 ExtraNET.com are registered trademarks of SXC. All rights reserved
.
</div>
</div>
</html>
```

Backend

Manages the view of Model View Template (MVT) Architecture (Views.py)

```
from django.shortcuts import render, redirect
from django.db.models import F
from .forms import *
from .models import *
from django.contrib.auth.forms import UserCreationForm, PasswordChangeForm
from django.contrib.auth import authenticate, login, logout, update_session_auth_hash
from django.contrib import messages
from django.http import JsonResponse
from .decorators import *
import datetime
from django.utils import timezone
from django.core import serializers
import json
```

```
def homepage(request):
    return redirect('login')

@must_login
@allowed_users(allowed_roles=['teacher'])

def create_paper(request):
    if request.method == 'POST':
        formdata = PaperForm(request.POST)
```

```
if formdata.is_valid():
    paper = formdata.save(commit=False)
    paper.dept = request.user.teacher.dept
    paper.paper_setter = request.user.teacher
    paper.save()
    return redirect('papers_and_questions')

form = PaperForm()
context = {'form': form}
return render(request, 'create.html', context)
```

```
@must_login
@allowed_users(allowed_roles=['teacher'])

def create_question(request, pk):
    if request.method == 'POST':
        formdata = Question_answerForm(request.POST, request.FILES)
        if formdata.is_valid():
            print(request.POST)
            q = formdata.save(commit=False)
            paper = Paper.objects.get(id=pk)
            q.paper = paper
            q.save()
            formdata.save_m2m()

    return redirect('view_paper_questions', pk)
form = Question_answerForm()
```

```
context = {'form': form}
return render(request, 'create.html', context)

@must_login
@allowed_users(allowed_roles=['teacher'])
def update_paper(request, pk):
    paper = Paper.objects.get(id=pk)

    if request.method == 'POST':
        if not paper.unlock_paper:
            formdata = PaperForm(request.POST, instance=paper)
        else:
            formdata = PaperForm_unlocked(request.POST, instance=paper)
        if formdata.is_valid():
            formdata.save()
            if paper.unlock_paper:
                m = Marks.objects.filter(paper=paper)
                for x in m:
                    x.modified_om = x.om/paper.fm*paper.modified_fm
                    x.save()

    return redirect('papers_and_questions')
if not paper.unlock_paper:
    form = PaperForm(instance=paper)
else:
```

```

form = PaperForm_unlocked(instance=paper)
context = {'form': form}
return render(request, 'create.html', context)

@must_login
@allowed_users(allowed_roles=['teacher'])
def update_question(request, pk):
    q = Question_answer.objects.get(id=pk)
    if request.method == 'POST':
        if q.paper.unlock_paper:
            formdata = Question_answerForm_unlocked(request.POST, instance=q)
        else:
            formdata = Question_answerForm(
                request.POST, files=request.FILES, instance=q)
        if formdata.is_valid():
            formdata.save()
            if q.paper.unlock_paper:
                s = Student.objects.filter(dept=q.paper.dept)
                for x in s:
                    check_and_save_marks(x, q.paper)
            return redirect('view_paper_questions', q.paper.id)

    if q.paper.unlock_paper:
        form = Question_answerForm_unlocked(instance=q)
    else:

```

```
form = Question_answerForm(instance=q)
context = {'form': form}
return render(request, 'create.html', context)

@must_login
@allowed_users(allowed_roles=['teacher'])
def delete_paper(request, pk):
    paper = Paper.objects.get(id=pk)

    if request.method == 'POST':
        paper.delete()
        return redirect('papers_and_questions')
    context = {'paper': paper}
    return render(request, 'delete_paper.html', context)

@must_login
@allowed_users(allowed_roles=['teacher'])
def delete_question(request, pk):
    q = Question_answer.objects.get(id=pk)

    if request.method == 'POST':
        q.delete()
        return redirect('view_paper_questions', q.paper.id)
    context = {'q': q}
```

```
return render(request, 'delete_question.html', context)

@auth_check
def login_page(request):

    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user:
            login(request, user)
            if user.groups.all()[0].name == 'teacher':
                return redirect('teacher_dashboard')
            else:
                return redirect('student_dashboard')
        else:
            messages.info(request, 'Wrong Credentials')

    context = {}
    return render(request, 'login.html', context)

@must_login
def change_password(request):
```

```

if request.method == 'POST':
    formdata = PasswordChangeForm(user=request.user, data=request.POST)
    if formdata.is_valid():
        formdata.save()
        update_session_auth_hash(request, formdata.user)
        group = None
        if formdata.user.groups.exists():
            group = request.user.groups.all()[0].name
            if group == 'teacher':
                return redirect('teacher_dashboard')
            else:
                return redirect('student_dashboard')
        else:
            messages.info(
                request, 'Invalid old or new password(s). Follow Instructions properly.')
            return redirect('change_password')

form = PasswordChangeForm(user=request.user)
context = {'form': form}
return render(request, 'change_password.html', context)

```

```

@must_login
@allowed_users(allowed_roles=['teacher'])

def teacher_dashboard(request):
    context = {}

```

```
return render(request, 'teacher_dashboard.html', context)

@must_login
@allowed_users(allowed_roles=['teacher'])

def papers_and_questions(request):
    papers = request.user.teacher.paper_set.all()
    context = {'papers': papers}
    return render(request, 'papers_and_questions.html', context)
```

```
@must_login
@allowed_users(allowed_roles=['teacher'])

def view_paper_questions(request, pk):
    paper = Paper.objects.get(id=pk)
    q = Question_answer.objects.filter(paper=paper)
    tmpJson = serializers.serialize(json, q)
    tmpObj = json.dumps(tmpJson)
    context = {'paper': paper, 'q': tmpObj}
    return render(request, 'view_paper_questions.html', context)
```

```
@must_login
@allowed_users(allowed_roles=['teacher'])

def view_question(request, pk):
    q = Question_answer.objects.get(id=pk)
```

```
context = {'q': q}
return render(request, 'view_question.html', context)
```

```
@must_login
```

```
@allowed_users(allowed_roles=['teacher'])
```

```
def unlock_papers_and_marks(request):
```

```
    papers = request.user.teacher.paper_set.all()
```

```
    curr_time = timezone.now()
```

```
    for x in papers:
```

```
        if x.publishable or not x.unlock_paper:
```

```
            continue
```

```
        if x.exam_date + x.duration + x.grace_time < curr_time:
```

```
            x.publishable = True
```

```
            x.save()
```

```
    if request.method == 'POST':
```

```
        print(request.POST)
```

```
        d = dict(request.POST)
```

```
        del d['csrfmiddlewaretoken']
```

```
    for p, c in d.items():
```

```
        paper = Paper.objects.get(id=p)
```

```
        if c[0] == 'unlock_paper':
```

```
            paper.unlock_paper = True
```

```

paper.exam_date = curr_time
else:
    paper.publish_result = True
    paper.result_date = curr_time
paper.save()
return redirect('teacher_dashboard')

context = {'papers': papers, 'curr_time': curr_time}
return render(request, 'unlock_papers_and_marks.html', context)

```

@must_login

@allowed_users(allowed_roles=['teacher'])

```

def marks_for_teacher(request):
    papers = request.user.teacher.paper_set.all()
    if request.method == 'POST':
        papers = request.user.teacher.paper_set.all()

```

```

p_id = request.POST['paper']
paper1 = Paper.objects.get(id=p_id)
marks = Marks.objects.filter(paper=paper1)

```

```

context = {'papers': papers, 'marks': marks, 'paper': paper1}
return render(request, 'marks_for_teacher.html', context)

```

```

context = {'papers': papers}
return render(request, 'marks_for_teacher.html', context)

```

```
@must_login  
@allowed_users(allowed_roles=['student'])  
def marks_for_student(request, pk):  
    paper = Paper.objects.get(id=pk)  
    m = Marks.objects.get(paper=paper, student=request.user.student)  
    published_papers = Paper.objects.filter(  
        dept=request.user.student.dept, publish_result=True)  
    context = {'m': m, 'paper': paper, 'published_papers': published_papers}  
    return render(request, 'marks_for_student.html', context)
```

```
@must_login  
@allowed_users(allowed_roles=['student'])  
def student_dashboard(request):  
    papers = Paper.objects.filter(  
        dept=request.user.student.dept, unlock_paper=True)  
  
    unattended_papers = papers.exclude(  
        exam_entry__student=request.user.student)  
  
    attended_papers = papers.filter(exam_entry__student=request.user.student)  
  
    missed_papers = []  
    give_papers = []
```

```
attended1_papers = []
resume_papers = []
curr_time = timezone.now()
for x in unattended_papers:
    if x.exam_date + x.duration + x.grace_time < curr_time:
        missed_papers.append(x)
    else:
        give_papers.append(x)

for x in attended_papers:
    e = Exam_entry.objects.get(paper=x, student=request.user.student)
    if x.exam_date + x.duration + x.grace_time < curr_time or e.submitted:
        attended1_papers.append(x)
    else:
        resume_papers.append(x)

published_papers = Paper.objects.filter(
    dept=request.user.student.dept, publish_result=True)

context = {
    'missed_papers': missed_papers,
    'give_papers': give_papers,
    'resume_papers': resume_papers,
    'attended_papers': attended1_papers,
    'published_papers': published_papers
}

}
```

```
return render(request, 'student_dashboard.html', context)

def logout_user(request):
    logout(request)
    return redirect('login')

@must_login
@allowed_users(allowed_roles=['student'])
def give_exam(request, pk, type1):
    paper = Paper.objects.get(id=pk)
    student = request.user.student
    if request.method == 'POST':
        print(request.POST)
        if paper.exam_date + paper.duration + paper.grace_time >= timezone.now():
            Student_answer.objects.filter(
                student=student, question__paper=paper).delete()
            e = Exam_entry.objects.get(student=student, paper=paper)
            save_answer_ajax(student, e, dict(request.POST))
            e.submitted = True
            e.save()
            check_and_save_marks(student, paper)
    return redirect('student_dashboard')

if type1 == 'n':
```

```
if Exam_entry.objects.filter(student=student, paper=paper).count() > 0:  
    e = Exam_entry.objects.get(  
        paper=paper, student=student)  
  
    if not (paper.exam_date + paper.duration + paper.grace_time < timezone.now() or not  
            paper.unlock_paper or e.submitted):  
        return redirect('give_exam', pk, 'r')  
  
    else:  
        return redirect('student_dashboard')  
  
n = paper.count  
arr1 = Question_answer.objects.filter(paper=paper)  
arr2 = arr1.random(n)  
  
timer_left_time = paper.duration  
  
e1 = Exam_entry(student=student, paper=paper,  
                time_left=timer_left_time)  
e1.save()  
e2 = Exam_entry.objects.get(student=student, paper=paper)  
for x in arr2:  
    e2.question.add(x)  
e3 = e2  
  
temp_arr = serializers.serialize('json', arr2)  
arr_json = json.dumps(temp_arr)  
context = {'arr': arr_json, 'attempted_answers': None, 'arrL': n,  
          'timer_left_time': timer_left_time.seconds}
```

else:

 e = Exam_entry.objects.get(paper=paper, student=request.user.student)

 if paper.exam_date + paper.duration + paper.grace_time < timezone.now() or not
 paper.unlock_paper or e.submitted:

 return redirect('student_dashboard')

 e = Exam_entry.objects.get(student=student, paper=paper)

 arr2 = e.question.all()

 attempted_answers = Student_answer.objects.filter(

 student=student, question__in=arr2)

 temp_arr = serializers.serialize('json', arr2)

 arr_json = json.dumps(temp_arr)

 temp_attempted_answers = serializers.serialize(

 'json', attempted_answers)

 attempted_answers_json = json.dumps(temp_attempted_answers)

 context = {'arr': arr_json, 'attempted_answers': attempted_answers_json, 'arrL': arr2.count(),

 'timer_left_time': e.time_left.seconds}

 e3 = e

 total_time = paper.exam_date + paper.duration + paper.grace_time

 server_left_time = total_time-timezone.now()

 context['server_left_time'] = server_left_time.seconds

 context['paper'] = paper

 context['warning'] = e3.warning

return render(request, 'ajax_exam.html', context)

```

@must_login
@allowed_users(allowed_roles=['student'])

def ajax_exam(request, pk):
    if request.method == 'POST':
        print(request.POST)
        paper = Paper.objects.get(id=pk)
        if paper.exam_date + paper.duration + paper.grace_time >= timezone.now():
            student = request.user.student
            e = Exam_entry.objects.get(student=student, paper=paper)

            save_answer_ajax(student, e, dict(request.POST))
            return JsonResponse({'status': 1})

def check_and_save_marks(student, paper):
    ans = Student_answer.objects.filter(student=student, question__paper=paper)
    count = ans.count()

    correct_count = ans.filter(
        question__correct_option__iexact=F('selected_option')).count()

    wrong_count = count - correct_count
    om = correct_count*(paper.marks_per_cq)+wrong_count*(paper.marks_per_wq)

    modified_om = om/(paper.fm)*(paper.modified_fm)

```

```
p = om/(paper.fm)*100

if p >= 90:
    grade = 'Excellent'
elif p >= 75:
    grade = 'Good'
elif(p >= 60):
    grade = 'Not bad'
else:
    grade = 'Fail'

marks = Marks.objects.filter(student=student, paper=paper)

if marks.count() == 0:
    mark = Marks(student=student, paper=paper, om=om,
                 modified_om=modified_om, grade=grade)
    mark.save()
else:
    m = marks[0]
    m.om = om
    m.modified_om = modified_om
    m.grade = grade
    m.save()

def save_answer_ajax(student, e, d1):
    del d1['csrfmiddlewaretoken']
    for q, a in d1.items():
```

```
if len(a[0]) == 0:  
    continue  
  
if q == 'timer_left_time':  
    e.time_left = datetime.timedelta(seconds=int(a[0]))  
    e.save()  
    continue  
  
if q == 'warning':  
    e.warning = int(a[0])  
    e.save()  
    continue  
  
temp1 = student.student_answer_set.filter(question__id=q)  
if temp1:  
    if temp1[0].selected_option == a[0]:  
        pass  
    elif a[0] == 'u':  
        temp1[0].delete()  
    else:  
        temp1[0].selected_option = a[0]  
        temp1[0].save()  
else:  
    if a[0] == 'u':  
        pass  
    else:  
        question = Question_answer.objects.get(id=q)  
        selected_option = a[0]
```

```
s_a_temp = Student_answer(  
    student=student, question=question, selected_option=selected_option)  
s_a_temp.save()
```

Definition of model classes (models.py)

```
from django.db import models
from django.contrib.auth.models import User
from django_random_queryset import RandomManager
sdaaaaaaa

class Department(models.Model):
    ID = models.CharField(primary_key=True, max_length=10)
    name = models.CharField(max_length=20)

    def __str__(self):
        return self.name


class Teacher(models.Model):
    ID = models.CharField(primary_key=True, max_length=10)
    name = models.CharField(max_length=50)
    user = models.OneToOneField(User, null=True, on_delete=models.SET_NULL)
    post = models.CharField(max_length=20)
    dept = models.ForeignKey(Department, null=True, on_delete=models.SET_NULL)

    def __str__(self):
        return self.name


class Student(models.Model):
    ID = models.CharField(primary_key=True, max_length=10)
    name = models.CharField(max_length=50)
    user = models.OneToOneField(User, null=True, on_delete=models.SET_NULL)
    roll = models.IntegerField()
    sem = models.IntegerField()
    academic_year = models.IntegerField()
    dept = models.ForeignKey(Department, on_delete=models.CASCADE)

    def __str__(self):
        return self.name
```

```

class Paper(models.Model):
    code = models.CharField(max_length=10)
    name = models.CharField(max_length=20)
    count = models.IntegerField()
    marks_per_cq = models.IntegerField()
    marks_per_wq = models.FloatField()
    fm = models.IntegerField()
    modified_fm = models.FloatField()
    duration = models.DurationField(null=True, blank=True)
    grace_time = models.DurationField(null=True, blank=True)
    dept = models.ForeignKey(Department, on_delete=models.CASCADE)
    paper_setter = models.ForeignKey(
        Teacher, null=True, on_delete=models.SET_NULL)
    unlock_paper = models.BooleanField(default=False)
    publish_result = models.BooleanField(default=False)
    exam_date = models.DateTimeField(null=True)
    publishable = models.BooleanField(default=False)
    result_date = models.DateTimeField(null=True)

    def __str__(self):
        return self.name


class Question_answer(models.Model):
    objects = RandomManager()
    options = [
        ('a', 'A'),
        ('b', 'B'),
        ('c', 'C'),
        ('d', 'D')
    ]
    paper = models.ForeignKey(Paper, on_delete=models.CASCADE)
    desc = models.TextField(max_length=100)
    op1 = models.CharField(max_length=50)
    op2 = models.CharField(max_length=50)
    op3 = models.CharField(max_length=50)
    op4 = models.CharField(max_length=50)

    correct_option = models.CharField(choices=options, max_length=1)
    img_path = models.ImageField()

```

```
blank=True, null=True, upload_to='exam_images')

def __str__(self):
    return self.desc[:10]

class Student_answer(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
    question = models.ForeignKey(
        Question_answer, null=True, on_delete=models.SET_NULL)
    selected_option = models.CharField(max_length=1)

    def __str__(self):
        return self.selected_option

class Marks(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
    paper = models.ForeignKey(Paper, on_delete=models.CASCADE)
    om = models.FloatField()
    modified_om = models.FloatField()
    grade = models.CharField(max_length=10)

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

class Exam_entry(models.Model):
    student = models.ForeignKey(Student, on_delete=models.CASCADE)
    paper = models.ForeignKey(Paper, on_delete=models.CASCADE)
    time_left = models.DurationField(null=True, blank=True)
    submitted = models.BooleanField(default=False)
    question = models.ManyToManyField(Question_answer)
    warning = models.IntegerField(default=3)
```

Definition of Django forms (forms.py)

```

from django.forms import ModelForm
from .models import *
from django import forms

class PaperForm(ModelForm):
    class Meta:
        model = Paper
        fields = ['code', 'name', 'count', 'marks_per_cq',
                  'marks_per_wq', 'fm', 'modified_fm', 'duration', 'grace_time']
        widgets = {
            'code': forms.TextInput(attrs={'class': 'form-control'}),
            'name': forms.TextInput(attrs={'class': 'form-control'}),
            'count': forms.NumberInput(attrs={'class': 'form-control'}),
            'marks_per_cq': forms.NumberInput(attrs={'class': 'form-control'}),
            'marks_per_wq': forms.NumberInput(attrs={'class': 'form-control'}),
            'fm': forms.NumberInput(attrs={'class': 'form-control'}),
            'modified_fm': forms.NumberInput(attrs={'class': 'form-control'}),
            'duration': forms.TextInput(attrs={'class': 'form-control', 'placeholder':
                'HH:MM:SS'}),
            'grace_time': forms.TextInput(attrs={'class': 'form-control', 'placeholder':
                'HH:MM:SS'})}

        labels = {
            'code': 'PAPER CODE',
            'name': 'PAPER NAME',
            'count': 'NO. OF QUESTIONS',
            'marks_per_cq': 'MARKS PER CORRECT ANSWER',
            'marks_per_wq': 'MARKS PER WRONG ANSWER',
            'fm': 'FULL MARKS',
            'modified_fm': 'MODIFIED FULL MARKS',
            'duration': 'DURATION OF EXAM',
            'grace_time': 'GRACE TIME',
        }

        help_texts = {
            'duration': 'Enter the duration in [HH:MM:SS] this format',
        }
    
```

```

        'grace_time': 'Enter the grace time in [HH:MM:SS] this format',
    }

class PaperForm_unlocked(ModelForm):
    class Meta:
        model = Paper
        fields = ['modified_fm', 'grace_time']
        widgets = {
            'modified_fm': forms.NumberInput(attrs={'class': 'form-control'}),
            'grace_time': forms.TextInput(attrs={'class': 'form-control'}),
        }
        labels = {
            'modified_fm': ' MODIFIED FULL MARKS',
            'grace_time': 'GRACE TIME',
        }
        help_texts = {
            'grace_time': 'Enter the grace time in [HH:MM:SS] this format',
        }

class Question_answerForm(ModelForm):
    class Meta:
        model = Question_answer
        fields = ['desc', 'img_path', 'op1', 'op2', 'op3',
                  'op4', 'correct_option']
        widgets = {
            'desc': forms.Textarea(attrs={'class': 'form-control', 'rows': '3'}),
            'img_path': forms.FileInput(attrs={'class': 'form-control'}),
            'op1': forms.TextInput(attrs={'class': 'form-control'}),
            'op2': forms.TextInput(attrs={'class': 'form-control'}),
            'op3': forms.TextInput(attrs={'class': 'form-control'}),
            'op4': forms.TextInput(attrs={'class': 'form-control'}),
            'correct_option': forms.Select(attrs={'class': 'form-control'}),
        }
        labels = {
            'desc': 'QUESTION DESCRIPTION',
            'img_path': 'UPLOAD RELATED PICTURE(IF ANY)',
            'op1': 'OPTION NO.:1',
            'op2': 'OPTION NO.:2',
        }

```

```
'op3': 'OPTION NO.:3',
'op4': 'OPTION NO.:4',
'correct_option': 'CORRECT OPTION'
}
```

```
class Question_answerForm_unlocked(ModelForm):
    class Meta:
        model = Question_answer
        fields = ['correct_option']
        widgets = {'correct_option': forms.Select(
            attrs={'class': 'form-control'}), }
        labels = {'correct_option': 'CORRECT OPTION'}
```

Controls the user permission and privileges (decorators.py)

```
from django.http import HttpResponseRedirect
from django.shortcuts import redirect

def auth_check(view_func):
    def wrapper_func(request, *args, **kwargs):
        if request.user.is_authenticated:
            group = None
            if request.user.groups.exists():
                group = request.user.groups.all()[0].name
                if group == 'teacher':
                    return redirect('teacher_dashboard')
                else:
                    return redirect('student_dashboard')
            else:
                return view_func(request, *args, **kwargs)
        return wrapper_func
    def must_login(view_func):
        def wrapper_func(request, *args, **kwargs):
            if not request.user.is_authenticated:
                return redirect('login')
            return view_func(request, *args, **kwargs)
        return wrapper_func
    return must_login
```


Adds the models in Django Admin panel (admin.py)

```
from django.contrib import admin  
from .models import Department, Teacher, Student, Paper, Question_answer  
  
admin.site.register(Department)  
admin.site.register(Teacher)  
admin.site.register(Student)  
admin.site.register(Paper)  
admin.site.register(Question_answer)
```

Manages all URLs of the exam application (urls.py)

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

urlpatterns = [
    path("", homepage, name='homepage'),
    path('create_paper/', create_paper, name='create_paper'),
    path('give_exam/', give_exam, name='give_exam'),
    path('login/', login_page, name='login'),
    path('logout/', logout_user, name='logout'),
    path('change_password/', change_password, name='change_password'),
    path('ajax/<str:pk>/', ajax_exam, name='ajax_exam'),
    path('give_exam/<str:pk>/<str:type1>/', give_exam, name='give_exam'),
    path('teacher_dashboard/', teacher_dashboard, name='teacher_dashboard'),
    path('unlock_papers_and_marks/',
         unlock_papers_and_marks,
         name='unlock_papers_and_marks'),
    path('marks_for_teacher/', marks_for_teacher, name='marks_for_teacher'),
    path('marks_for_student/<str:pk>/',
         marks_for_student,
         name='marks_for_student'),
    path('create_question/<str:pk>/', create_question, name='create_question'),
    path('update_paper/<str:pk>/', update_paper, name='update_paper'),
    path('update_question/<str:pk>/', update_question, name='update_question'),
    path('delete_paper/<str:pk>/', delete_paper, name='delete_paper'),
    path('delete_question/<str:pk>/', delete_question, name='delete_question'),
    path('view_question/<str:pk>/', view_question, name='view_question'),
    path('student_dashboard/', student_dashboard, name='student_dashboard'),
```

```
path('view_paper_questions/<str:pk>', view_paper_questions,  
     name='view_paper_questions'),  
path('papers_and_questions/', papers_and_questions,  
     name='papers_and_questions'),  
]
```

Manages configuration/settings of the project (Main project (settings.py))

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '^g$7@v2$*aei$&ync$z%npf4g58yzb#q)$l)4u-9rsoef6=*3c'

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

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'travello.apps.TravelloConfig',
    'django.contrib.admin',
```

```
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'queries',
'exam'

]
```

```
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 = 'practice.urls'
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        # for templates look into the 'templates' folder
}
```

```
'DIRS': [os.path.join(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',  
    ],  
},  
,  
]  
  
WSGI_APPLICATION = 'practice.wsgi.application'
```

```
# Database  
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql',  
        'NAME': 'practice',  
        'USER': 'postgres',  
        'PASSWORD': '1234',
```

```
'HOST': 'localhost'  
}  
}  
  
# Password validation  
# https://docs.djangoproject.com/en/3.1/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/3.1/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/3.1/howto/static-files/
```

```
STATIC_URL = '/static/'
```

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

```
STATIC_ROOT = os.path.join(BASE_DIR, 'assets')
```

```
MEDIA_URL = '/media/'
```

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

To redirect every request to the exam application (Main project (urls.py))

```
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('exam/', include('exam.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Testing

Introduction

The testing phase is an important part of software development. It is the putrefied system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met, and the user requirements are satisfied.

Software testing is carried out in three steps:

- 1) The first includes unit testing, where in each module is tested to provide its correctness, validity and determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately. Unit testing is the important and major part of the project. So, errors are rectified easily in particular module and program clarity is increased. In this project entire system is divided into several modules and is developed individually. So, unit testing is conducted to individual modules.
- 2) The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole. The individual modules are clipped under this major module and tested again and verified the results. This is due to poor interfacing, which may result in data being lost across an interface. A module can have inadvertent, adverse effect on any other or on the global data structures, causing serious problems.
- 3) The final step involves validation and testing which determines which the software functions as the user expected. Here also some modifications were. In the completion of the project, it is satisfied fully by the end user. Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

Testing Objectives:

- 1) Testing is a process of executing a program with the intent of finding an error.
- 2) A good test case is one that has a probability of finding a yet undiscovered error.
- 3) A successful test is one that uncovers an undiscovered error.

Testing Principles

- 1) All tests should be traceable to end user requirements.
- 2) Tests should be planned long before testing begins.
- 3) Testing should begin on a small scale and progress towards testing in large.
- 4) Exhaustive testing is not possible.
- 5) To be most effective testing should be conducted by an independent third party

The primary objective for test case design is to derive a set of tests that has the highest likelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are: -

White box testing:

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Block-box testing:

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Testing strategies:

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Testing fundamentals:

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully, it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

Testing Information flow:

Information flow for testing flows the pattern. Two class of input provided to test the process.

The software configuration includes a software requirements specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are conducted and all the results are evaluated. That is test results are compared with expected results. When erroneous data are uncovered, an error is implied and debugging commences.

Unit testing:

Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to uncover errors within the boundary of the modules. These tests were carried out during the programming stage itself.

Integration testing:

Integration testing focuses on unit tested modules and build the program structure that is dictated by the design phase.

System testing:

System testing tests the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specification and system documentation. The primary concern is the compatibility of individual modules. Entire system is working properly or not will be tested here and giving output or not are tested here. These verifications and validations are done by giving input values to the system and by comparing with expected output. Top-down testing implementing is done here.

Acceptance Testing:

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

Test coverage Analyzer – records the control paths followed for each test case.

Timing Analyzer – also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.

Coding standards – static analysers and standard checkers are used to inspect code for deviations from standards and guidelines.

Test Cases:

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that.

- ❖ Guarantee that logical decisions on their true and false sides.
- ❖ Exercise all logical decisions on their true and false sides.
- ❖ Execute all loops at their boundaries and within their operational bounds.
- ❖ Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences.

Test case ID	What is done	Expected Results
1.	Login testing: In the login page correct and wrong credentials are entered periodically for several times.	When correct credentials are entered user is successfully logged in to the system. Otherwise, user is shown the ‘wrong credentials’ message.
2.	Hyperlink testing: All the links to the other webpages are clicked and checked multiple times whether or not the corresponding pages are opening properly.	Webpages corresponding to the links should open with no issue.
3.	Question paper CRUD operation testing: <ol style="list-style-type: none"> a. By clicking the create a new paper button all the details are properly entered and the save button is clicked and this is performed multiple times. b. Sometimes some fields are kept blank 	a. If all the fields are entered correctly the new paper is created successfully. b. If some fields are kept blank proper message is shown. c. If details are entered correctly with no blank field the paper is updated with no issue. d. If user confirms to delete a paper it's

	<p>purposely to see the effect.</p> <p>c. Clicking the update button and updating the paper details entered previously.</p> <p>d. Clicking the delete button and trying to delete button and clicking the yes button to delete a paper.</p>	<p>deleted from database with all its question and respective student answers if the exam has been taken by any of them.</p>
4.	<p>Question Answer CRUD operation testing:</p> <p>a. By clicking the new question button all the details are properly entered and the save button is clicked and this is performed multiple times.</p> <p>b. Sometimes some fields are kept blank purposely to see the effect.</p> <p>c. Clicking the update button and updating the question details entered previously.</p> <p>d. Clicking the delete button and trying to delete button and clicking the yes button to delete a question answer pair.</p>	<p>a. If all the fields are entered correctly the new question-answer pair is created successfully.</p> <p>b. If some fields are kept blank proper message is shown.</p> <p>c. If details are entered correctly with no blank field the question-answer pair is updated with no issue and if exam is over for that paper all the answers are evaluated successfully.</p> <p>d. If user confirms to delete a question, it's deleted from database.</p>

5.	<p>Unlock paper and publish marks functionality testing:</p> <ol style="list-style-type: none"> 1. A already created, paper is unlocked from the teacher's ID and it's availability is checked from a student's profile. 2. While an exam is going on attempting to publish its result. 3. When an exam is over trying to publish its result. 	<ol style="list-style-type: none"> a. Paper is unlocked properly and becomes available to the concerned students. b. While an exam is going on it publish result button will not appear. c. Result gets published. Respective teacher and students can check it from the results section.
6.	<p>Exam interface testing:</p> <p>During an exam,</p> <ol style="list-style-type: none"> a. different options are selected, b. Some selected options are cleared (by clicking the clear button) c. Different question numbers are clicked to see proper questions are coming or not with previously selected option (if any). 	<ol style="list-style-type: none"> a. Options should be changed when clicked on another one. b. When clear button is clicked for a question it's previously selected option should be deselected and accordingly the color of the respective question number button should be vanished. c. Upon clicking on a question number, the respective question should appear with the previously selected option (if any).
7.	<p>Exam Continuity testing:</p> <p>During the exam the student exits from the exam by closing the respective tab in the browser and logs in again to</p>	<p>The student should be able to resume his/her exam with all the previously done answers present in it. Also, the timer should start recounting from where it stopped earlier.</p>

	his account to resume his exam if time is left.	
8.	<p>Warning testing:</p> <p>During the exam the student tries to,</p> <ul style="list-style-type: none"> a. Exit from full screen mode, b. Open another tab, c. Open some other application 	In all the mentioned scenarios he/she gets a warning every time such an action is performed and if that number exceeds a predefined threshold the answer gets auto submitted.
9.	<p>Auto submit testing:</p> <p>When the timer is about to finish its countdown, no action is taken to see whether the answers get auto submitted or not.</p>	The answers should be auto submitted when the timer finishes its counting, and the student should be redirected to his/her dashboard.

Test Report

Test case ID	Functionality being tested	Number of testcases run	Success rate
1.	Login	7	100%
2.	Hyperlinks of all webpages	20	100%
3.	Question Paper CRUD	6	100%
4.	Question Paper CRUD	15	100%

5.	Paper unlocking and Result publishing	7	100%
6.	Exam interface testing	20	100%
7.	Exam Continuity testing	10	98%
8.	Warning testing:	10	100%
9.	Auto submit testing	6	100%

Security and Special Features

Secure Browsing

Introduction

Just like any other sector, the educational sector got affected by the coronavirus pandemic tremendously. The authorities in the educational sector were aware of the effects which this lockdown would cause on the future of the students and hence decided to overcome the obstacles by embracing technology.

Hence, instead of accepting defeat and shutting down, the educational sector inculcated technologically advanced ways of teaching and learning. It adopted systems for conducting online classes and exams. The students appeared for the exams from their homes or any remote location which brought in the biggest challenge of online examination i.e. cheating or carrying out any malpractices.

We have developed a platform to conduct online examinations which is known as Extranet. Every feature was designed carefully for this platform after putting in a lot of thought procedure about the ways to overcome each and every challenge faced while conducting exams online. One of the main features offered by Extranet is the secure browser for Online Exams.

Let's understand how Secure Browser for Online Exams actually work.

Need for Secure Browser in Online Exam

As the students started giving exams from different remote locations, the supervision for these exams became difficult. The supervisors could not remain physically present to monitor the students as they write the exams. This increased the chances of cheating or malpractices being carried out by the students with Online Examinations.

The candidates might try to open a new browser or window to search for the answers on the internet. This would hamper the integrity of the exam as the browser in which the exam is going on should be the only one which can be accessed by the candidate.

The students might as well try to take screenshots of the questions or answers or even try to record the exam. To prevent students from doing so, the feature for secure browser for online exams has been used in Extranet's platform.

How Secure Browser for Online Exams work?

This feature detects the students opening any window or switching between different tabs.

- It ensures that the candidate's exam window is blocked in Full Screen Mode and only then the students can see the questions and answer them. This helps to prevent the students from opening a new tab and thus secures the exam from unauthorized cheating from the internet.

Click on the **Go to exam** button to give exam.

Try not to exit fullscreen once in.

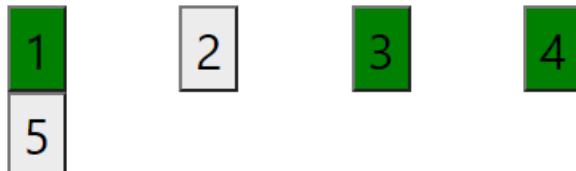
Repeated exiting will lead to suspension.

Go to exam

- This feature also blocks access to different shortcut keys such as cut, paste and copy. It can also stop the feature of right click.
- If the student tries to open a new window or change tabs or run any unwanted apps/software in the background, the secure browser sends a warning message to the student in exam interface.
- These warning messages are displayed in the screen of exam interface. If the system sends more than <no of warnings> such messages, then the exam process stops immediately, and the student gets disqualified. The students' access is blocked, and they cannot continue with the online exam, the student is then redirected to the student dashboard page automatically.

Time left : 0 : 12 : 31

Warning: A new window has been opened or a pop has occurred. Doing it for 2 more time[s] will get you disqualified from the exam.



Random Question Selector

As we all know in Online Examinations cheating happens very rampantly. For sake of security and to prevent any malpractice by the students we have developed a system of Random Question Selector.

In this system the no two student will get the same set of question. Every student will have same number of question but the questions may not be the same. Even the ordering of the question will vary from student to student.

In the question database of certain paper, the teacher will set up questions equal or greater than the number of questions for that exam. then at the time of examination when the student gets into the exam portal then this system selects the predefined number of questions from the question database randomly and displays them to the exam portal of a certain student. But the selected questions of this student might differ from the selected questions of the second student.

For example, say there are two students A &B. The number of Questions for this examination say, 4. The teacher has set 10 questions for this test. now when the student A gives the exam, the

questions selected for Student A may be 1, 5, 3, 9, 6. This selection of question and the order of the questions will not be the same for student B. Student B might get the questions say, 10, 4, 8, 1, 2.

Up to a certain extent this facility ensures that a student will not get the full freedom to copy from another student.

Autosave Feature

Online Examinations are very much dependent on computer devices, Internet connections, Electricity and so on. These things are very likely to malfunction. If the malfunction occurs in the middle of the exam, then that will be a very unfortunate for a student as all his answers will be deleted. To make our system secure of these malfunctions we included the feature of Autosave.

In this feature, the answers of the student will continue to be saved in the server in certain interval of time. This ensures that the attempted answers of a student are saved till the malfunction occur. This feature also saves the time remaining to the end of the exam, so when the student gets back to resume the exam, the student gets directly in the state where he left the examination. The student gets all the attempted answers and the time left to end the exam as they were before the malfunction.

This feature ensures that even if there is a malfunction in the system or in the internet connection the student answers are saved and the student can catch up the exam where it was left.

The system accepts two types of “give exam” request, one a new exam and Resume exam. When there has been a problem during the exam and the student system malfunctions, the student can get back to the student dashboard and then there this option of “resume exam”. This time the previous answers of the student will be saved in the exam window.

Future Scope and Improvements

Remote Proctoring

Proctoring has an important role in ensuring the credibility and authenticity of any test and its outcome. The growth of online exams and assessments has automatically given rise to online exam proctoring. As long as the traditional and online versions of exams coexist, so will both types of proctoring, namely the traditional or offline proctoring and online exam proctoring.

Online exam proctoring: It is also referred to as remote proctoring in which the candidate is usually monitored via a webcam, mic and access to the screen of the candidate during the exam. There are mainly three types of remote proctoring:

- **Live online exam proctoring:** A qualified proctor monitors the candidates' audio-video and screen share feeds in real-time sitting at a remote location. Therefore, it is not very different from offline proctoring except for the flexibility of the proctoring location. Human involvement renders live online exam proctoring unsuitable for scalability as it turns out to be expensive.
- **Recorded proctoring:** Candidates' audio-video and screen share feeds are recorded and reviewed by a proctor at a later time. Usually, the recordings are played in a fast-forward manner for the proctor to detect any suspicious activities. The human involvement retains the disadvantages of the live online exam proctoring despite removing location and time constraints.
- **Automated proctoring:** The latest and most advanced form of proctoring, it almost removes any need for human intervention at all. Face recognition helps authenticate the candidates' identity. The system records the audio-video and screen share feeds and simultaneously monitors them for any suspicious activities using advanced video and audio analytics. This type of proctoring works by limiting candidate logins to specific IP addresses and blocking copy-pasting. It does not allow the candidate to switch the window or access any other application during the test. Automated proctoring also tracks mouse movement, candidate movement and other set functions. Removal of the location and time constraint make automated proctoring economical and fool proof.

Question Pattern

The question pattern in the system is currently created for MCQs only. However, a fully-fledged system should have a varied of questions so that the exam setter can set any pattern of paper at ease.

With a little modification it can be used for integer type questions as well. The database needs to be modified to store integer values and then can be matched with the student's answers to evaluate the correct result.

The database can also be modified to store long answers as well.

Notice Board

A notice board can be created so that the student can visit the website and can see what the appearing examinations are and preparing accordingly. The exam website can be itself used to notify the students about the examination and need not to be informed separately. This would lead to a better user interface system.

Bibliography

- “**Head First Python**” – Paul Barry
Publisher - O'Reilly Media, Inc
- “**Bootstrap 4.0 Documentation**” - <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
Last accessed on 29th March 2021.
- “**JavaScript – The Complete Edition (2012 Edition)**” - Thomas A. Powell, Fritz Schneider.
Publisher – Mc Graw Hill Education
- “**Django documentation 3.1**” - <https://docs.djangoproject.com/en/3.1/>
Last accessed on 5th March 2021.
- “**HTML & CSS: Design and Build Websites (2011 Edition)**” - John Duckett.
Publisher- John Wiley & Sons, Inc.