# INTRODUCTION

# 1.INTRODUCTION

In today's digital world, taking exams securely and conveniently is crucial EduSecureOnline offers a solution to this challenge by providing a safe and user-friendly platform where students can attend exams from anywhere, anytime. In today's digital age, the traditional methods of conducting examinations are being challenged by various limitations and vulnerabilities. EduSecureOnline aims to mitigate these challenges by providing a secure and convenient solution that ensures students can attend exams safely from the comfort of their own environment.

## 1.1 OBJECTIVE

EduSecureOnline endeavors to provide a secure and convenient platform for college students to attend exams remotely, ensuring accessibility and inclusivity regardless of geographical location or physical abilities. Our objective is to streamline examination management for administrators, uphold academic integrity, and promote a user-friendly experience through robust security measures, efficient administration tools, and intuitive interfaces. Additionally, we strive to support scalability to accommodate the growing demand for online learning and assessment while maintaining reliability and performance.

# SYSTEM ANALYSIS

# 2. SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problem and using the fact to improve the system. Analysis is the detail study of various operations performed by the system and their relationship with in and outside of the system. This involves gathering information and using structured tools for analysis. System analysis is the way of studying a system with an eye on solving its problem using computer. To analyse a system, one has to study the working of the system in detail. The system analyst has to understand the functioning and concept of system in detail. The system analyst has to understand the functioning and concept of system in detail, before designing the appropriate computer based system that will meet all the requirements of the existing system

## 2.1 EXISTING SYSTEM.

In the existing educational system, communication between universities, colleges, departments, teachers, and students may primarily rely on traditional methods such as email, notice boards, physical newsletters, and word-of-mouth. These methods may lack efficiency, real-time updates, and centralized access to information. Consequently, students and faculty members might miss out on important announcements, events, deadlines, and other relevant updates.

## 2.2 PROPOSED SYSTEM

The proposed system Offering a centralized platform for universities and colleges to streamline communication. Through real-time updates and notifications, it addresses the inefficiencies of traditional methods, ensuring students and faculty stay informed about important announcements, events, and deadlines. By providing centralized access to information, EduSecureonline aims to enhance communication and mitigate the risk of missing crucial updates.

KEY FEATURES AND BENEFITS:

Secure Authentication: Utilizes robust user authentication mechanisms to ensure only authorized individuals can access examination materials, safeguarding the integrity of assessments.

Exam Administration: Empowers administrators to effortlessly create, schedule, and manage examination sessions, customizing parameters such as exam duration and question formats.

Real-time Monitoring: Provides real-time monitoring features during exams to detect and deter any suspicious behavior or violations of exam protocols, ensuring exam integrity.

Enhanced Security: Ensures the integrity and confidentiality of examinations, mitigating the risk of cheating and unauthorized access.

Convenience: Enables students to attend exams remotely from any location, eliminating the need for physical attendance at examination centers.

Efficiency: Streamlines exam administration processes, saving time and resources for both students and administrators.

# SYSTEM REQUIREMENTS

## 3. SYSTEM REQUIREMENTS

| | | |
|---|---|---|
| **Frame work** | **:** | **HTML 5** |
| **Language** | **:** | **Django** |
| **IDE** | **:** | **Visual Studio Code** |
| **Front End Design** | **:** | **HTML 5, CSS 3.0** |
| **Validation** | **:** | **JavaScript 1.1** |
| **Server** | **:** | **Django Development Server** |
| **Back End Database** | **:** | **SQLite3** |
| **Database IDE** | **:** | **DB Browser** |
| **Processor** | **:** | **Intel CORE i3** |
| **RAM** | **:** | **4 GB** |

| Item | Requirements |
| --- | --- |
| Hardware | Memory of 4 GB or more<br>Intel Pentium III or more<br>1 GB (or more) available hard disk space |
| Operating System | Windows 10 or above |
| Software Development Kit | HTML 5 |
| Database ODBC Driver | MySQL |

# FEASIBILITY STUDY

# 4. FEASIBILITY STUDY

The main objective of this study is to determine whether the proposed system is feasible or not. Mainly there are three types of feasibility study to which the proposed system is subjected to as described below:

- Economic feasibility
- Technical feasibility
- Behavioural feasibility

The proposed system must be evaluated from a technical view point first, and if technically feasible then their impact on organization must be accessed. If compatible, the behavioural system can be devised. Then those must be tested for economic feasibility.

## 4.1 ECONOMIC FEASIBILITY

Economic analysis is used for evaluating the cost effectiveness of a proposed system. Economically saying, our application is never cost related. It is easy to implement and the user can conveniently use our application without any kind of exceptions or errors. Our application is very simple to interact is economically feasible.

## 4.2 TECHNICAL FEASIBILITY

The technical requirements of our system are highly affordable. There is no any difficulty in migrating from the existing to our proposed system. It is platform independent the application will run smoothly on any kernel without causing any trouble to end user.

## 4.3 BEHAVIORAL FEASIBILITY

The system is intended for personal uses. Since the software is simple and provides a good Graphical User Interface, it provides high level user.

# SYSTEM IMPLEMENTATION

# 5. SYSTEM IMPLEMENTATION

The implementation and the maintenance are the last phase of software development life cycle. After testing is performed, that is checked whether the system works in the proper manner by putting realistic data items, the system is implemented in the client's computer. Here is the most important part that is user training. The working of the system is explained to the user. The main objective of their part is that the user will be able to understand the different process. Only the external working is implemented system satisfies the user various needs. Implementation is the stage of the project where the theoretical design is turned into working system, in this stage the installation of the package in the real environment, to the satisfaction of the intended user and the operation of the system. Implementation includes all those activities that take place to convert from the old system to new one. The new system may be totally new, replacing an existing system manual automates system or it may be a major modification to an existing system. Proper implementation is essential to provide reliable system to meet organization requirements, successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it.

The process of putting the developed system in actual use is called system implementation. This include all those activities that take place convert from the old system to new system can be implemented only after through testing is done and if it is found to be working according to the specification. The system personnel check the feasibility of the system.

The most crucial stage is achieving a new system for the user that will work efficiently. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods achieve the changeover. The complex system being implemented, more involved will be the system analysis and the design required effort required just for implementation. The implementation stage involves the following tasks:

- Careful planning.
- Investigating of system and constraint. 16
- Design of methods to achieve the changeover.
- Training of staff in the changeover phase.
- Evaluation of the changeover method.

## 5.1 TRAINING

After providing the necessary basic training on the computer awareness the users will have to be trained on the new system such as screen flow, screen design, type of help on the screen, type of errors while entering the data the corresponding validation check at each entry and the ways to correct the data entered. It should then cover information needed by the specific user/group to use the system while imparting the training of the program on the application

## 5.2 CONVERSION

It is the process of performing all of the operation that result directly in the turnover of the new system to the user conversion has two parts:

- The creation of the conversion plan at the start of the development phase and the implementation of the plan throughout the development phase.
- The creation of a system changes over plan at the end of the development phase and the implementation of the plan at the beginning of the operation phase.

## 5.3 POST IMPLEMENTATION REVIEW

The post implementation review in the implementation process as a whole is conducted after the implementation of the actual software in the site. It is conducted using the review document. This may contain the answers to the following question.

- What was the purpose of the project?
- What would you like about the implementation process?
- What thinks would you change if you had to do it again tomorrow?

## 5.4 SYSTEM MAINTENANCE

Once the software is fully developed and implemented, the company starts to use the software. The company also goes and more division can be attached to the company, or the database of the company can grow in size. So after sometime the software, which has been installed, needs some modification. If the software need modification all the step needed to develop new software has to be executed.

The need has to be studied, the design has to be made and coding has to be done. Then the new module has to be connected to the existing software modules. Once the software is working perfect also, we have to do routine testing and any new bug is found out, immediately it has to be fixed. No software ever developed will be bug free forever. When a new situation arises, the software can create an error, but if it is found out and repaired the software will not causing more problems. Always maintenance has to be done on the software, for to make the software work perfectly without any errors. Maintenance of the software is one of major step in the computer animation software which is developed by the engineer, should undergo maintenance process in regular interval of time goes on new problems arise and it must be corrected accordingly.

Maintenance and enhancement are a long-term process. Various types of maintenance that can be made:

- Corrective Maintenance
- Adaptive Maintenance
- Prefecture Maintenance
- Reverse Engineering
- Re Engineering

# DATA FLOW DIAGRAM

# 6.DATA FLOW DIAGRAM

## 6.1 INTRODUCTION TO DATA FLOW DIAGRAM

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

Specific operations based on the data can be represented by a flowchart. There are several notations for displaying data-flow diagrams.

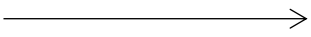
For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in a parallel unlike a flowchart which also shows this information

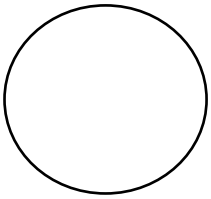
**To construct a Data Flow Diagram,we use:**

It represents data source or destination

It represents the data store

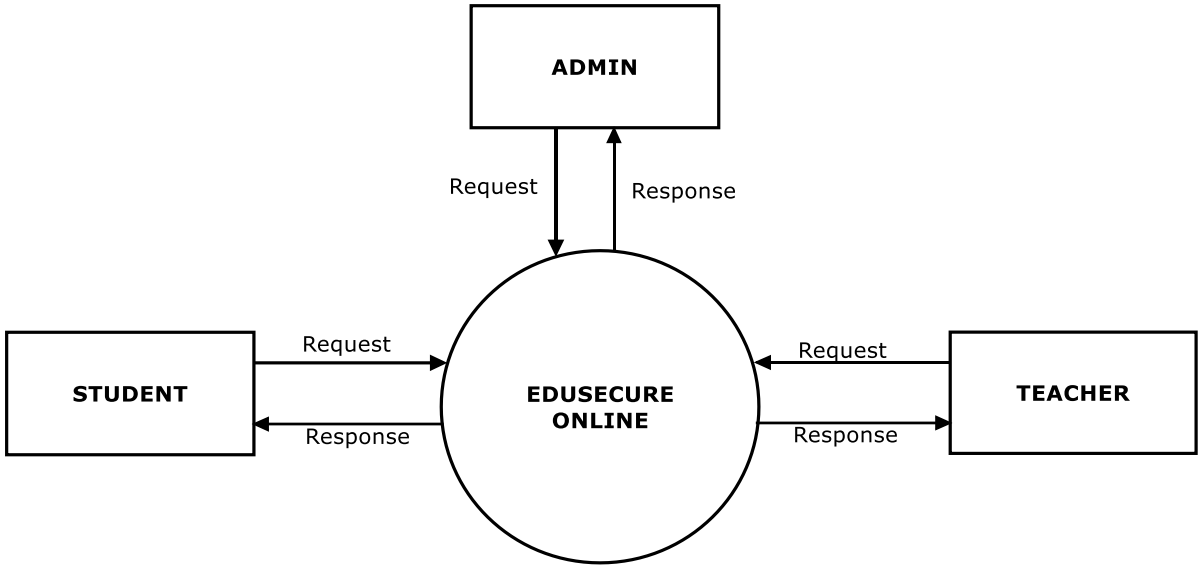


It represents flow of data

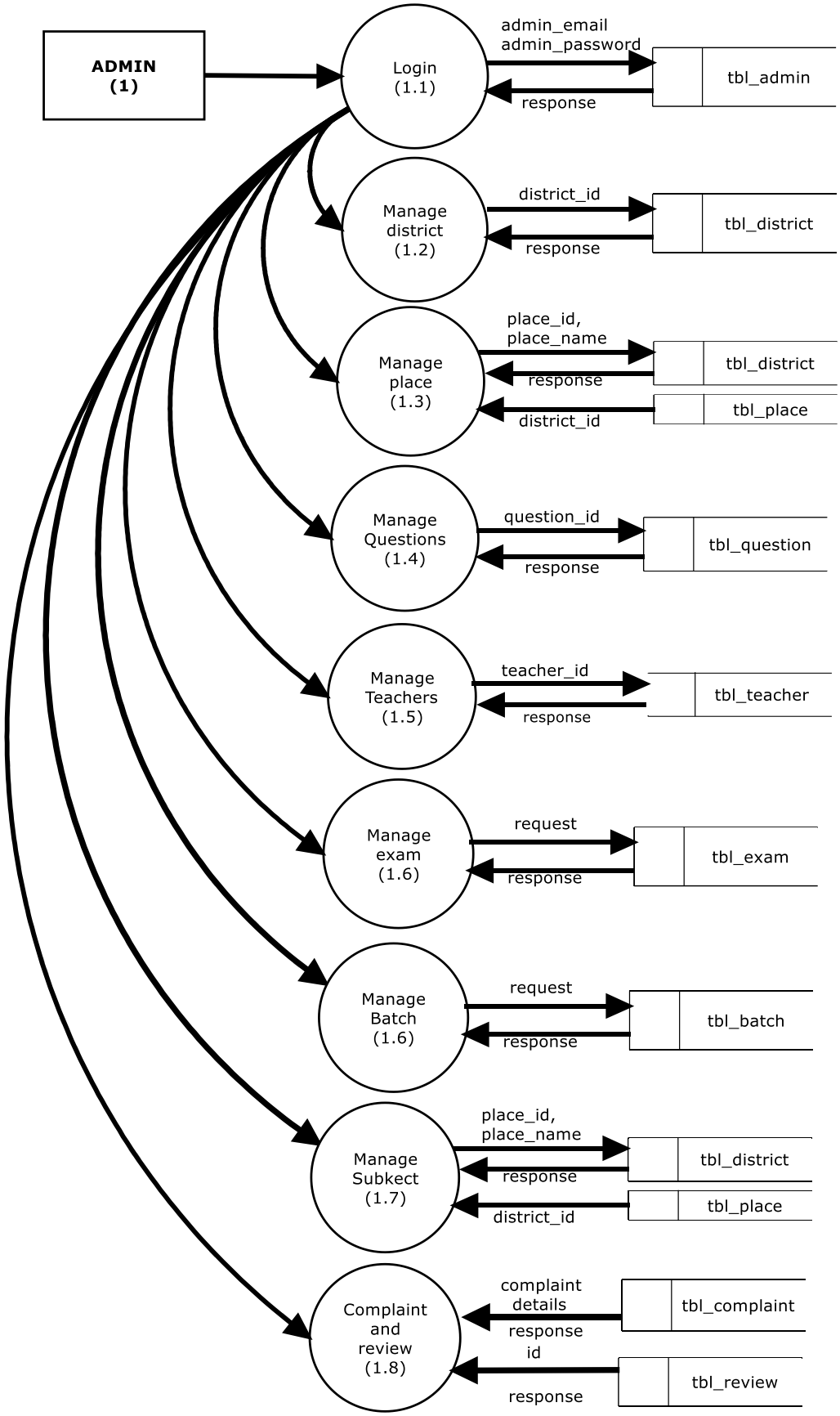


It represents a process that transforms the data

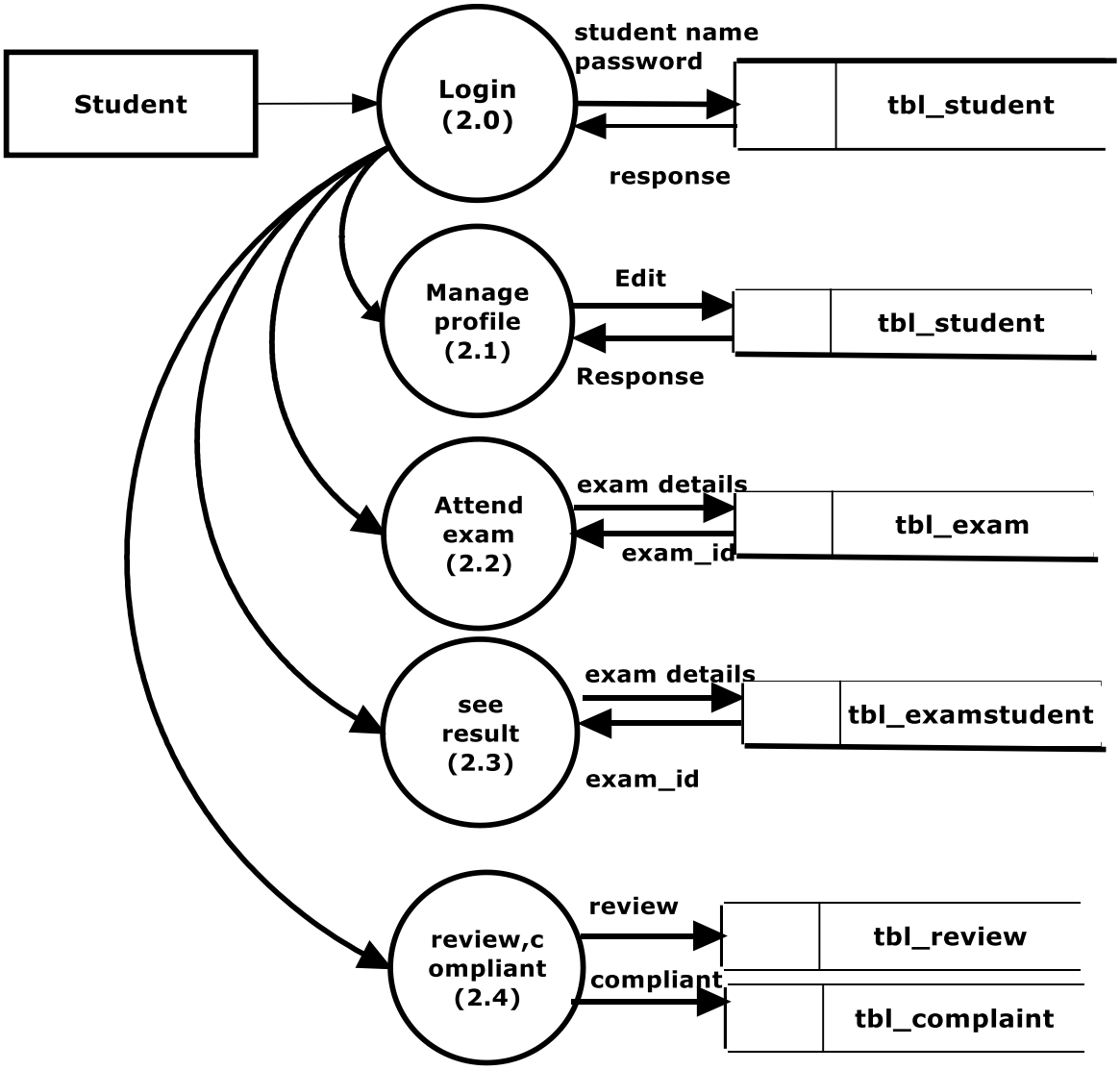
## 6.2 LEVEL-0


ADMIN
Request
Response
STUDENT
Request
Response
EDUSECURE
ONLINE
Request
Response
TEACHER

### 6.3 LEVEL 1 : ADMIN

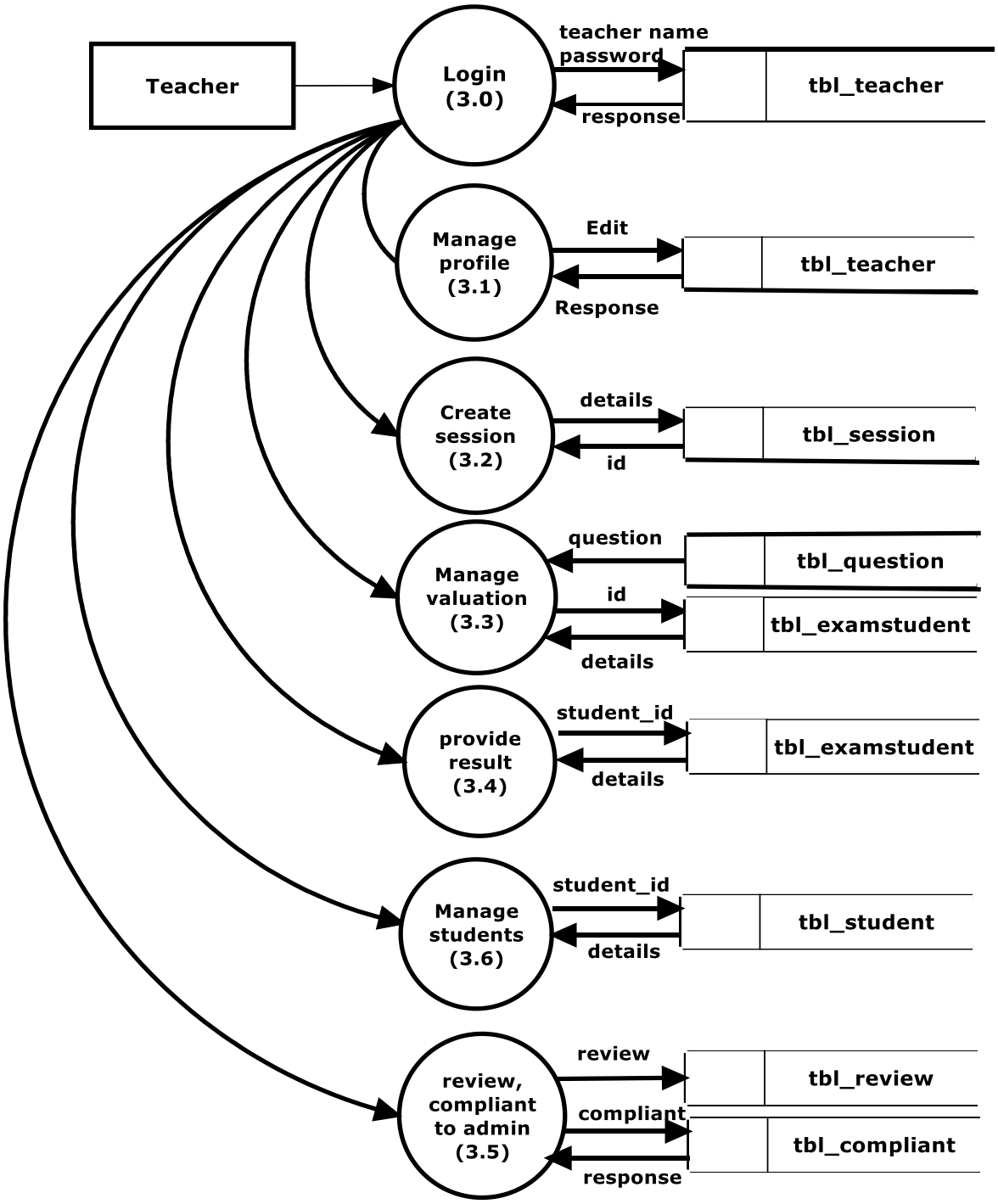


## 6.4 LEVEL 1: STUDENT




**S**

## 6.5 LEVEL 1: TEACHER




x

# SYSTEM DESIGN

# 7.SYSTEM DESIGN

## 7.1 INTRODUCTION TO SOFTWARE DESIGN

System design is the creative act of invention developing new inputs, a database offline files, method, procedures and output for processing business data to meet organization objectives. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from user oriented document to a document oriented to the programmers or database personnel.

Characteristics of a well-designed system are:

- Accessibility
- Decision making ability
- Economy
- Flexibility
- Reliability
- Simplicity

The design will determine the success of the system. System design is based on the information gathered during system analysis. System design goes through two phases of development.

- Logical design- DFD shows the logical flow of a system and defines the boundaries of the system. For the candidate the system it describes the input, outputs, databases and procedures-all in a format that meets the user's requirements.

- Physical design – This produces the working system by defining the design specification that tells programmers exactly what the candidate system must do.

## 7.2 INPUT DESIGN

Input design is the link that ties the information system into the world of its users. The input design involves determining what the inputs are, how the data should be performed, how to validate data, how minimize data entry and how to provide a multi-user facility inaccurate. Input data are the most common cause of error in data processing. Errors entered by data entry operators can be controlled by input design. Input design is the process of converting user.

Originated inputs to a computer-based format. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing. All the input data validated in the order and if any data violates any condition, he user is warned by a message. If the data satisfies all the conditions, then it is transferred to the appropriate tables in the database.

## 7.3 OUTPUT DESIGN

Output design involves specifying how production of onscreen reports and pagebased reports will occur. Output may occur to database or file for storing information entered or also for use by other systems. Computer output is the most important and direct source of information to the user. Output design is a very important phase because the output needs to be in an attractive manner. Efficient, intelligible output design improves the system relationship with the user and help in decision making. A major form of the output is the hard copy from the printer and screen reports. Printouts are designed around the output requirements of the user. Allowing the user to view to the sample screen is important because the user is ultimate judge of the quality of output. The output model of this system is the user-friendly window.

These user-friendly windows are meant for the purpose of easy view of the stored information.

## 7.4 DATABASE DESIGN

A database is a collection of data. Database design refers to the design of the tables used to store data. The database involves name of records, data item with its name, type and size.

In the design of the database program first we have to thoroughly look into the requirements of the program for the design of database. Then we have to design how much tables are required in the database. Thereafter as per the requirement of the end users we can decide which fields that must be in this table. As per a general rule a provision must be taken in the design for the future enhancement of the program. Some of important tables are

**TABLE NO:1**

Table Name : tbl_admin

Primary Key : admin_id

Foreign Key : Null

Table Description: Admin details

| Filed Name | Data Type | Constraints | Description |
|---|---|---|---|
| admin_id | Int | Primary Key | Admin Id |
| admin_name | Varchar (30) | Not Null | Admin name |
| admin_email | Varchar (50) | Not Null | Admin email |
| admin_password | Varchar (30) | Not Null | Admin password |

**TABLE NO:2**

Table Name : tbl_district

Primary Key : district_id

Foreign Key : Null

Table Description: District details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| district_id | Int(11) | Primary Key | District Id |
| district_name | Varchar (30) | Not Null | District name |

**TABLE NO:3**

Table Name : tbl_place

Primary Key : place_id

Foreign Key : district_id

Table Description: Place details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| place_id | Int(11) | Primary Key | Place Id |
| place_name | Varchar (50) | Not Null | Place name |
| district_id | Int(11) | Foreign Key | District Id |

**TABLE NO:4**

Table Name : tbl_batch

Primary Key : batch_id

Foreign Key : Null

Table Description : Batch details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| Batchid | Int(11) | Primary Key | Batch Id |
| Batchname | Varchar (30) | Not Null | Batch name |

**TABLE NO:5**

Table Name : tbl_student

Primary Key : student_id

Foreign Key : teacher_id , place_id

Table Description: Student details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| student_id | Int(11) | Primary Key | student id |
| student_name | Varchar (30) | Not Null | student name s |
| student_email | Varchar (50) | Not Null | Student email |
| student_password | Varchar (30) | Not Null | Student password |
| student_contact | Varchar (20) | Not Null | Student contact |
| student_registerno | Varchar (20) | Not Null | Register number of students |
| student_address | Varchar (500) | Not Null | Student address |
| place_id | Int | Foreign Key | Place id |
| student_gender | Varchar(30) | Not Null | Gender of student |
| teacher_id | Int(11) | Foreign Key | teacher Id |

**TABLE NO:6**

Table Name : tbl_teacher

Primary Key : teacher_id

Foreign Key : place_id

Table Description: teacher details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| teacher_id | Int(11) | Primary Key | teacher Id |
| teacher_name | Varchar (30) | Not Null | name |
| teacher_email | Varchar (50) | Not Null | email |
| teacher_password | Varchar (30) | Not Null | password |
| teacher_contact | Varchar (20) | Not Null | contact |
| teacher_address | Varchar (500) | Not Null | teacher address |
| teacher_photo | Varchar (50) | Not Null | photo |
| teacher_proof | Varchar(30) | Not Null | proof |
| teacher_vstatus | Int | Not Null | District id |
| teacher _gender | Varchar(50) | Not Null | gender |
| place_id | Int | Foreign Key | Place id |
| teacher_college | Varchar(30) | Not Null | Name of the college of the teacher |

**TABLE NO:7**

Table Name : tbl_question

Primary Key : question_id

Foreign Key : section_id

Table Description: question details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| question_id | Int(11) | Primary Key | Question Id |
| question_content | Varchar(30) | Not Null | Question content |
| section_id | Int(11) | Primary Key | Section Id |

**TABLE NO:8**

Table Name : tbl_exam

Primary Key : exam_id

Foreign Key : subject_id

Table Description: exam details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| exam_id | Int(11) | Primary Key | exam_id |
| exam_date | Varchar (30) | Not Null | Exam name |
| subject_id | Int(11) | Foreign Key | subject Id |
| exam_status | Varchar(50) | Not Null | status |

**TABLE NO:9**

Table Name : tbl_subject

Primary Key : subject_id

Foreign Key : batch_id,

Table Description: subject details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| Subject_id | Int(11) | Primary Key | student Id |
| title | Varchar (30) | Not Null | Title of exam |
| Examiner_id | Int(11) | Foreign Key | Examiner Id |

**TABLE NO:10**

Table Name : tbl_examstudent

Primary Key : examstudent_id

Foreign Key : Exam_id

Table Description: exam appling students details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| examstudent_id | Int(11) | Primary Key | Student id |
| exam_status | Varchar (30) | Not Null | exam status |
| answer_mark | Int(100) | Not Null | answer_mark |

| | | | |
|---|---|---|---|
| exam_id | Int(100) | Foreign Key | Exam id |

**TABLE NO:11**

Table Name : tbl_review

Primary Key : review_id

Foreign Key : user_id

Table Description: review details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| review_id | Int | Primary Key | review Id |
| review_rating | Varchar (50) | Not Null | review rating |
| review_content | Varchar (200) | Not Null | review content |
| review_date | Int | Foreign Key | student Id |
| user_id | int (30) | Foreign Key | User id |

**TABLE NO:12**

Table Name : tbl_complaint

Primary Key : complaint_id

Foreign Key : exam_id

Table Description: Complaint details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| complaint_id | Int | Primary Key | Complaint Id |
| complainttitle | Varchar (500) | Not Null | Complaint title |
| complaintcontent | Varchar (500) | Not Null | Complaint content |
| complaint_status | Varchar (500) | Not Null | Complaint status |
| complaint_reply | Varchar (500) | Not Null | Complaint reply |
| exam_id | Int (11) | Foreign Key | exam Id |

**TABLE NO:13**

Table Name : tbl_section

Primary Key : section_id

Foreign Key : NULL

Table Description: section details

| Field Name | Data Type | Constraints | Description |
|---|---|---|---|
| section_id | Int | Primary Key | section Id |
| section_name | Varchar (50) | Not Null | Section name |
| section_mark | Varchar (200) | Not Null | Section mark |

# SECURITY TECHNOLOGIES

# 8. SECURITY TECHNOLOGIES

## 8.1 SECURITY TECHNOLOGIES AND POLICIES

Maintenance involves the software industry captive, typing up system resources. It means restoring something to its original condition. Maintenance follows conversion to the extend. That change is necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files. Password protection and simple procedure to prevent the unauthorized access are provided to the users. The system allows the user to enter the system only through proper username and password.

# MAINTENANCE

# 9. MAINTENANCE

In "EduSecureOnline," maintenance encompasses a broad spectrum of tasks aimed at ensuring optimal functionality and adapting to evolving needs. This includes rectifying coding and design errors, updating documentation and test data, and enhancing user support. Maintenance involves efficiently utilizing system resources within the software industry. It entails restoring the system to its original condition and follows conversion processes as necessary. Additionally, maintenance addresses issues reported by users, adjusts interfaces with other software or hardware, and incorporates enhancements to meet evolving standards and requirements. It enables future updates and modifications even after deployment, ensuring the software remains adaptable to changing environments and user needs.

# CONCLUSION AND FUTURE ENHANCEMENT

# 10. CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, EduSecureOnline represents a robust and comprehensive solution for conducting secure, efficient, and user-friendly online examinations for college students. With its array of features including secure authentication, exam administration, real-time monitoring, and intuitive user interface, EduSecureOnline ensures the integrity and fairness of assessments while offering convenience and flexibility to both students and administrators.By leveraging advanced security measures and sophisticated technology, EduSecureOnline addresses the challenges associated with traditional examination methods, providing a reliable and scalable platform for academic institutions. The platform's emphasis on user satisfaction, efficiency, and transparency underscores its commitment to delivering an exceptional examination experience for all stakeholders involved.As we continue to evolve and refine EduSecureOnline, our focus remains on enhancing its capabilities, expanding its features, and adapting to the evolving needs of the educational landscape. Through ongoing research and development efforts, we aim to introduce future enhancements that further elevate the platform's performance, security, and user experience.

Looking ahead, several areas of improvement and future enhancements have been identified for EduSecureOnline.

a) Enhanced Proctoring Features: Introduce advanced proctoring capabilities such as AI-based facial recognition and behavior analysis to further enhance exam monitoring and deterrence of cheating.

b) Adaptive Testing: Implement adaptive testing algorithms to personalize exam content based on students' performance, providing tailored assessments and optimizing learning outcomes.

c) Mobile Application: Develop a dedicated mobile application for EduSecureOnline to offer greater flexibility and accessibility for students to attend exams on their smartphones or tablets.

d) Enhanced Analytics and Reporting: Enhance analytics and reporting capabilities to provide administrators with deeper insights into exam performance, student behavior, and assessment trends, facilitating data-driven decision-making.

e) Accessibility Features: Implement accessibility features to ensure compliance with accessibility standards and accommodate students with disabilities, promoting inclusivity and equal access to examinations.

f) Collaborative Exam Features: Introduce collaborative exam features to facilitate group assessments and project-based evaluations, promoting collaborative learning and teamwork among students.

# APPENDIX

# 11.APPENDIX

## 11.1 CODES

### LOGIN PAGE

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Login Page</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         margin: 0;
         padding: 0;
         background-image: url('your-background-image.jpg'); /* Add your image path */
         background-size: cover;
         background-position: center;
         display: flex;
         justify-content: center;
         align-items: center;
         height: 100vh;
      }
      form {
         width: 300px;
         padding: 30px;
         border-radius: 8px;
         background-color: rgba(255, 255, 255, 0.8); /* Added background color with opacity
*/
         box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1);
      }
      table {
         width: 100%;
      }
      th, td {
         padding: 10px;
         border: none;
         border-bottom: 1px solid #ddd;
      }
      th {
         text-align: left;
      }
```

```
        input[type="text"], input[type="password"] {
            width: calc(100% - 20px);
            padding: 8px;
            margin-bottom: 20px;
            border: 1px solid #ddd;
            border-radius: 4px;
        }
        input[type="submit"] {
            width: 100%;
            padding: 10px;
            border: none;
            border-radius: 4px;
            background-color: #007bff;
            color: #fff;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }
        input[type="submit"]:hover {
            background-color: #0056b3;
        }
        input[type="submit"]:last-child {
            margin-top: 10px;
            background-color: #ddd;
            color: #555;
        }
        input[type="submit"]:last-child:hover {
            background-color: #ccc;
        }
    </style>
</head>
<body>
    <form id="form1" name="form1" method="post" action="">
        <div align="center">
            {% csrf_token %}
            <table>
                <tr>
                    <th scope="row">Email</th>
                    <td><input type="text" name="txt_email" id="txt_email" /></td>
                </tr>
                <tr>
                    <th scope="row">Password</th>
                    <td><input type="password" name="txt_password" id="txt_password" /></td>
                </tr>
                <tr>
```

```
                <th colspan="2" scope="row">
                    <input type="submit" name="Login" value="Login" />
                    <input type="submit" name="cancel" value="Cancel" />
                </th>
            </tr>
        </table>
    </div>
  </form>
</body>
</html>
```

**VIEW QUESTIONS**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ViewQuestions</title>
  <style>
    /* CSS styles */
    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        margin: 0;
        padding: 0;
        background-color: #f8f9fa; /* Light gray background */
        color: #333; /* Dark gray text color */
        line-height: 1.6;
    }

    .container {
        max-width: 800px;
        margin: 20px auto;
        padding: 20px;
        background-color: #fff;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        border-radius: 10px;
        position: relative; /* Position relative for absolute positioning of button */
    }

    .section {
        padding: 20px;
        margin-bottom: 30px;
```

```
    background-color: #f0f0f0;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.section-title {
    color: #007bff; /* Blue color for section titles */
    font-size: 24px;
    margin-bottom: 10px;
    text-transform: uppercase;
    text-align: center; /* Center the section title */
}

.question {
    margin-bottom: 15px;
    padding: 15px;
    background-color: #ffffff;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    position: relative;
}

.question p {
    margin: 0;
    font-size: 16px; /* Increase font size for better readability */
}

.question-number {
    font-weight: bold;
    color: #007bff; /* Blue color for question numbers */
}

.answer-link {
    position: absolute;
    top: 50%;
    right: 15px;
    transform: translateY(-50%);
    color: #007bff;
    text-decoration: none;
}

.answer-link:hover {
    text-decoration: underline;
}
```

```
        .finish-button {
            display: block;
            margin: 0 auto;
            margin-top: 20px; /* Add some space between questions and finish button */
            padding: 10px 20px;
            background-color: #007bff;
            color: #fff;
            text-decoration: none; /* Remove default underline */
            border-radius: 5px;
            cursor: pointer;
        }

        .finish-button:hover {
            background-color: #0056b3; /* Darker shade of blue on hover */
        }

        @media (max-width: 768px) {
            .container {
                padding: 10px;
            }
            .section-title {
                font-size: 20px;
            }
            .question {
                padding: 10px;
            }
            .answer-link {
                right: 10px;
            }
        }
    </style>
</head>
<body>
    <div class="container">
        <h1 id="timer"align="center">Time Left: 10:00</h1> <!-- Timer display -->
        {% for i in exam %}
        <div class="section">
            <h2 class="section-title">
                {{ i.section }}
            </h2>
            {% for j in i.examqst %}
            <div class="question">
                <p>
```

```
                <span class="question-number">{{ forloop.parentloop.counter }}.{{ forloop.counter }}. </span>{{ j.question }}
            </p>
            <a href="{% url 'wstudent:answer' j.id exid %}" class="answer-link">Answer</a>
          </div>
          {% endfor %}
        </div>
        {% endfor %}
        <a href="{% url 'wstudent:examcompletecheck' exid=exid %}" class="finish-button">Finish</a>
    </div>
    <script>
        // Timer function
        function startTimer(duration, display) {
            console.log("Timer")
            var timer = duration, minutes, seconds;
            setInterval(function () {
                minutes = parseInt(timer / 60, 10);
                seconds = parseInt(timer % 60, 10);

                minutes = minutes < 10 ? "0" + minutes : minutes;
                seconds = seconds < 10 ? "0" + seconds : seconds;

                display.textContent = "Time Left: " + minutes + ":" + seconds;

                if (--timer < 0) {
                    timer = 0; // Set timer to 0 when time is up
                    // Perform actions when time is up
                    alert("Time's up! Your exam will now be finished.");
                    // Redirect to the finish URL
                    window.location.href = "{% url 'wstudent:timeout' exid=exid %}";
                    ;
                }
            }, 1000);
        }

        // Start the timer when the page loads
        window.onload = function () {
            console.log('Start')
            var duration = 60; // Exam duration in seconds (10 minutes)
            var display = document.getElementById("timer");
            startTimer(duration, display);
        };
    </script>
```

```
        // Check if there's a message in the page
        {% if messages %}
            // Loop through each message
            {% for message in messages %}
                // Display the message using JavaScript alert
                <script>
                alert("{{ message }}");
                </script>
            {% endfor %}
        {% endif %}
</body>
</html>
```

**STUDENT REGISTERATION**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student Registration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f2f2f2;
        }
        .container {
            max-width: 600px;
            margin: 20px auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 5px;
```

```
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.container h2 {
    text-align: center;
    margin-bottom: 20px;
}
form {
    margin-top: 20px;
}
form table {
    width: 100%;
    border-collapse: collapse;
}
form table th,
form table td {
    padding: 10px;
    border: 1px solid #ccc;
}
form table th {
    width: 30%;
    text-align: left;
    background-color: #f2f2f2;
}
form table td {
    width: 70%;
}
form table select,
form table input[type="text"],
form table input[type="file"] {
    width: calc(100% - 6px); /* Adjust for border */
```

```
        padding: 8px;
        margin: 6px 0;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
    }
    form table textarea {
        width: calc(100% - 12px); /* Adjust for border */
        padding: 8px;
        margin: 6px 0;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
        resize: vertical;
    }
    form table input[type="submit"] {
        width: auto;
        padding: 10px 20px;
        margin-top: 20px;
        background-color: #007bff;
        color: #fff;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        transition: background-color 0.3s;
    }
    form table input[type="submit"]:hover {
        background-color: #007bff;
    }
</style>
```

```
</head>
<body>
  <div class="container">
    <h2>Student Registration Form</h2>
    <form id="form1" name="form1" method="post" action="" data-places-url="{% url 'wteacher:ajax_place' %}" enctype="multipart/form-data">
      {% csrf_token %}
      <table>
        <tr>
          <th>Name</th>
          <td><input type="text" name="txt_name" id="txt_name" required></td>
        </tr>
        <tr>
          <th>Contact</th>
          <td><input type="text" name="txt_contact" id="txt_contact" required></td>
        </tr>
        <tr>
          <th>Email</th>
          <td><input type="email" name="txt_email" id="txt_email" required></td>
        </tr>
        <tr>
          <th>Address</th>
          <td><textarea name="txt_address" id="txt_address" rows="5" required></textarea></td>
        </tr>
        <tr>
          <th>Photo</th>
          <td><input type="file" name="txt_photo" id="txt_photo" accept="image/*" required></td>
        </tr>
        <tr>
```

```
            <th>District</th>
            <td>
              <select name="txt_district" id="txt_district" required>
                <option value="">--Select--</option>
                {% for i in district %}
                  <option value="{{i.id}}">{{ i.district_name }}</option>
                {% endfor %}
              </select>
            </td>
          </tr>
          <tr>
            <th>Place</th>
            <td><select name="txt_place" id="txt_place" required>
              <option value="">--Select--</option>
            </select></td>
          </tr>
          <tr>
            <th>Gender</th>
            <td>
              <input type="radio" name="gender" id="Male" value="Male" required><label
for="Male">Male</label>
              <input type="radio" name="gender" id="Female" value="Female"
required><label for="Female">Female</label>
            </td>
          </tr>
          <tr>
            <th>Password</th>
            <td><input type="password" name="txt_password" id="txt_password"
required></td>
          </tr>
          <tr>
```

```
                <td colspan="2" align="center"><input type="submit" name="txt_submit"
id="txt_submit" value="Submit"></td>
            </tr>
            <a href="{% url 'wteacher:index' %}" style="display: inline-block; padding: 10px
20px; background-color: #007bff; color: white; text-decoration: none; border: none; border-
radius: 5px; cursor: pointer; transition: background-color 0.3s ease;">Home</a>
        </table>
    </form>
</body>
</html>


<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
    $("#txt_district").change(function () {
        var did = $(this).val();
        var ur = $("#form1").attr("data-places-url");
        $.ajax({
            url: ur,
            data: { disid: did, },
            success: function (data) {
                $("#txt_place").html(data);
            },
        });
    });
</script>


<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
    $("#txt_batch").change(function () {
        var did = $(this).val();
        var ur = $("#form1").attr("data-subjects-url");
```

```
        $.ajax({
            url: ur,
            data: { bthid: did, },
            success: function (data) {
                $("#txt_subject").html(data);
            },
        });
    });
</script>
```

**VIEW EXAM**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ViewExam</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 0;
        }

        #container {
            max-width: 800px;
            margin: 20px auto;
            background-color: #fff;
            padding: 20px;
```

```
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

form {
    text-align: center;
}

table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

table, th, td {
    border: 1px solid #ddd;
}

th, td {
    padding: 10px;
    text-align: left;
}

th {
    background-color: #f2f2f2;
}

select {
    padding: 8px;
    border-radius: 5px;
```

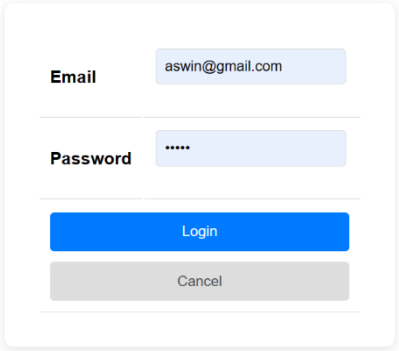
```
            border: 1px solid #ccc;
        }

        button {
            padding: 10px 20px;
            background-color: #007bff; /* Blue color */
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            transition: background-color 0.3s ease;
        }

        button:hover {
            background-color: #0056b3; /* Darker shade of blue on hover */
        }
    </style>
</head>
<body>
    <div id="container">
        <h1 style="text-align: center;">View Exam</h1>
        <form id="form1" name="form1" method="post" action="" data-subject-url="{% url
'wadmin:ajaxsubject' %}">
            <div>
                {% csrf_token %}
                <label for="batch">Batch:</label>
                <select id="ddl_batch" name="ddl_batch">
                    <option value="">Select</option>
                    {% for i in batch %}
                    <option value="{{i.id}}">{{i.batch_name}}</option>
```

```
            {% endfor %}
        </select>
    </div>
    <br>
    <div>
        <label for="subject">Subject:</label>
        <select id="ddl_subject" name="ddl_subject">
            <option value="">Select</option>
            {% for i in subject %}
            <option value="{{i.id}}">{{i.subject_name}}</option>
            {% endfor %}
        </select>
    </div>
    <br>
    <div>
        <button type="submit" name="txt_submit" id="txt_submit"
value="Submit">Submit</button>
    </div>
    <br>
        <a href="{% url 'wstudent:homepage' %}" class="back-button">Home</a>
</form>


<table>
    <thead>
        <tr>
            <th>Exam Date</th>
            <th>Subject</th>
            <th>View Questions</th>
        </tr>
        {% for i in exam %}
```

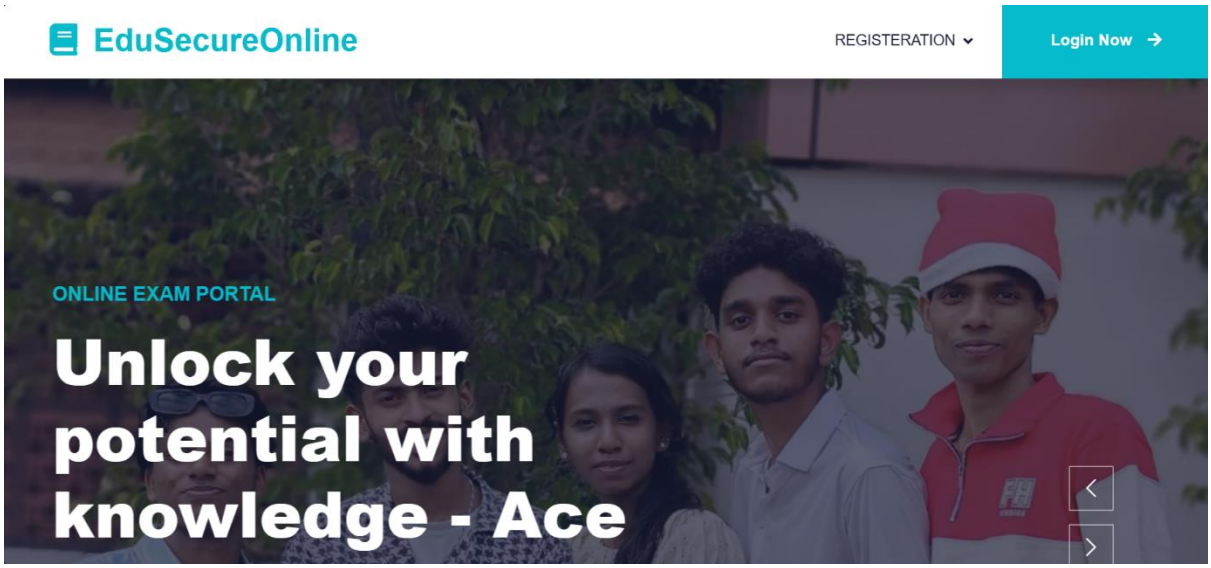
```
            <tr>
                <td>{{i.exam_date}}</td>
                <td>{{i.subjectid.subject_name}}</td>
                <td><a href="{% url 'wstudent:viewquestions' i.id  %}">Start Exam</a></td>
            </tr>
            {% endfor %}
        </thead>
        <tbody>
        </tbody>
    </table>
  </div>
</body>
</html>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script>
  $("#ddl_batch").change(function () {
    var did = $(this).val();
    var ur = $("#form1").attr("data-subject-url");
    $.ajax({
      url: ur,
      data: { bsid: did, },
      success: function (data) {
        $("#ddl_subject").html(data);
      },
    });
  });
</script>
```

## 11.2 SCREENSHOTS

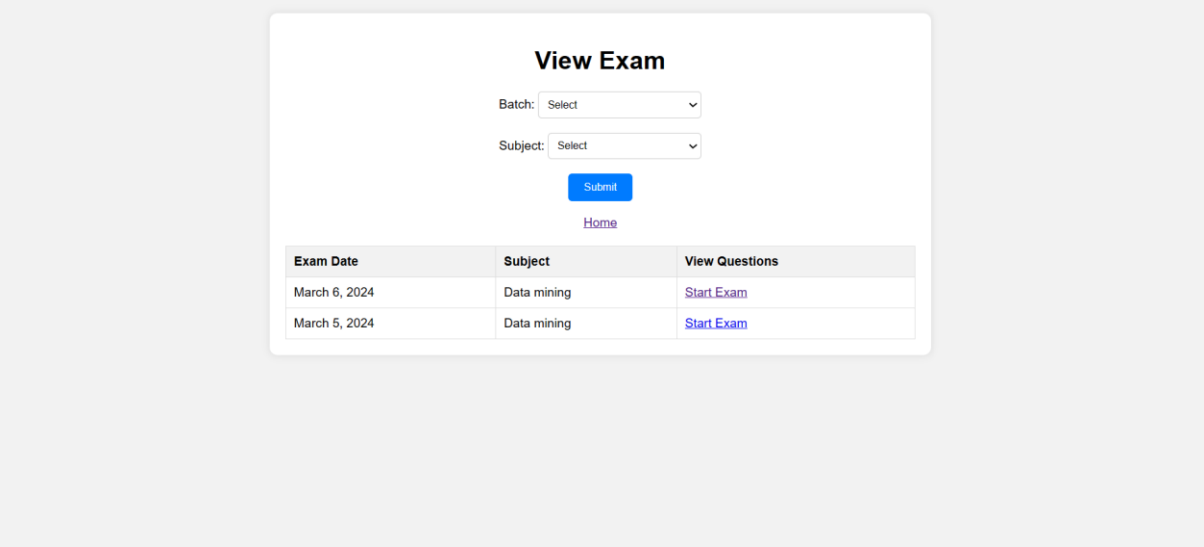
LOGIN PAGE



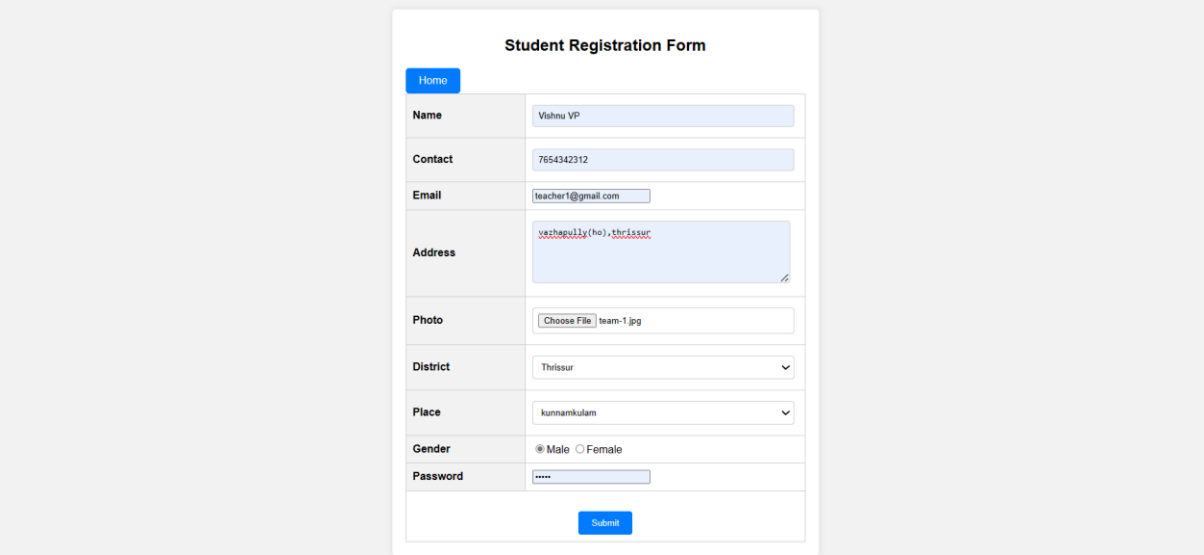

INDEX PAGE



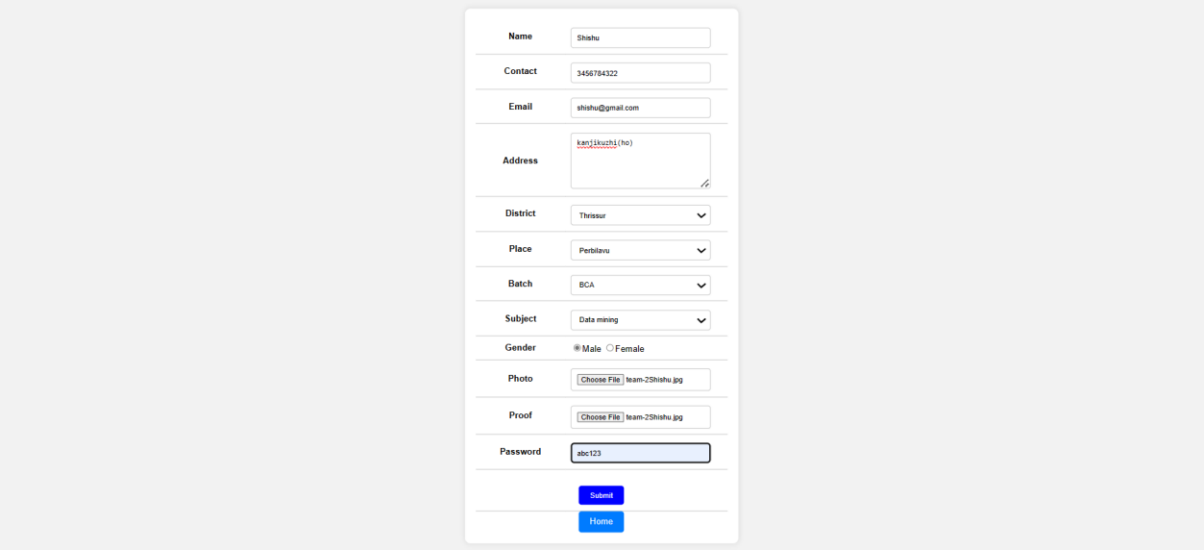
## VIEW EXAM



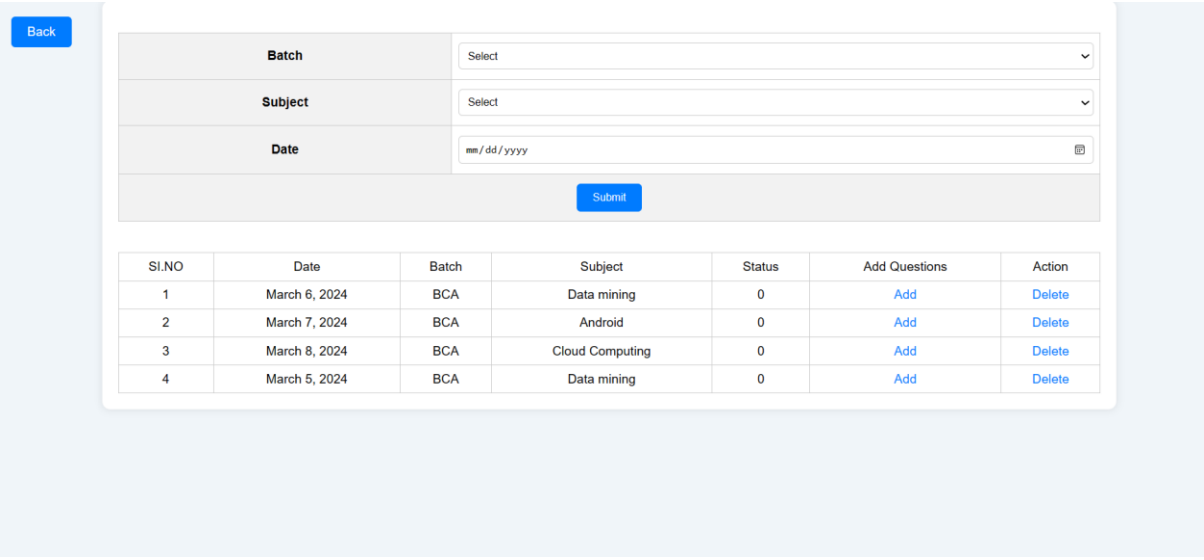

## STUDENT REGISTERATION



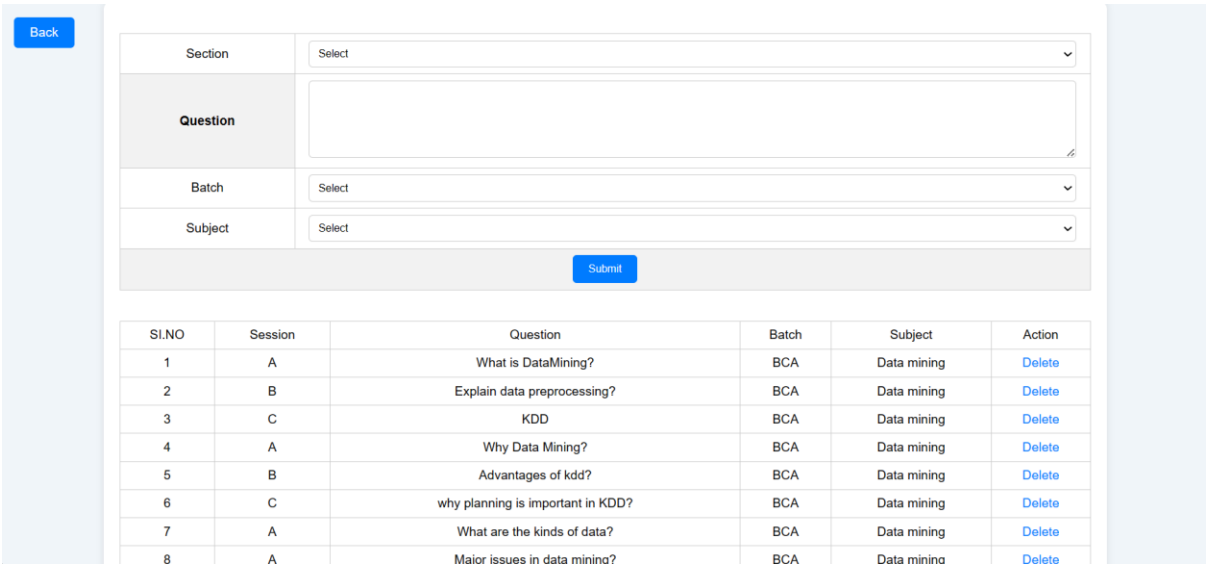
## TEACHER REGISTERATION



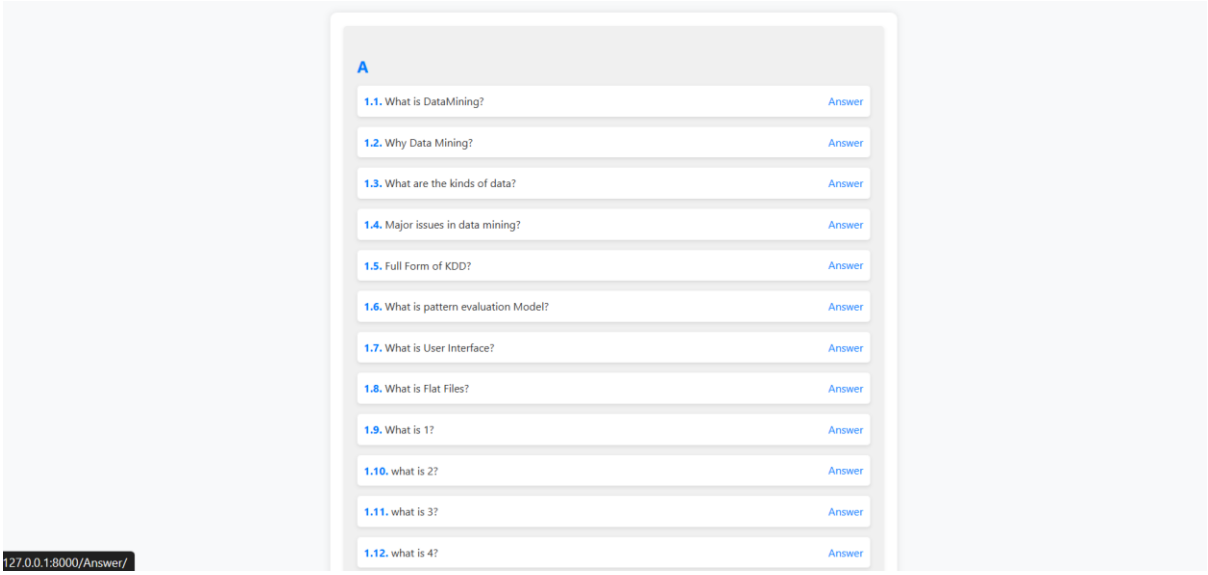

## EXAM PAGE



## QUESTIONS PAGE




## VIEW QUESTIONS



# BIBLIOGRAPHY

# 12.BIBLIOGRAPHY

1. www.google.com
2 .www.youtube.com
3. w3schools.com
4. github.com