Project Report On

Paper (An Online Exam Management System)

Sania Rahman

2017831051

Mehedi Hasan Shifat

2017831017

20th Feb,2021

Contents

1 Project Introduction

Technology has changed our lives.Now a days technology has gone so far that everything is online.But our examination system is still dependent on offline activities.This project is about an online web platform where we will be able to init some robust changes to enrich our examination system.

2 Project Background

Coronavirus Disease (COVID-19) outbreak poses serious concerns to global education systems. Efforts to contain COVID-19 prompted unscheduled closure of schools in more than 100 countries worldwide. COVID-19 school closures left over one billion learners out of school. Most governments around the world have temporarily closed educational institutions in an attempt to reduce the spread of COVID-19. As of 30 September 2020, approximately 1.077 billion learners are currently affected due to school closures in response to the pandemic. As a result Online learning has become a critical lifeline for education, as institutions seek to minimize the potential for community transmission. Technology can enable teachers and students to access specialized materials well beyond textbooks, in multiple formats and in ways that can bridge time and space. Even if online classes are being held but still exams aren't clear. As authority is not willing to take risk of student and teacher's life. So if there's a feasible online platform to take exams, then it'll be possible to clear and prevent session jam.

3 Project Description

An online examination is conducted digitally to evaluate students' academic knowledge and understanding of the curriculum, along with their ability to use creativity to devise new ideas and solutions. With the advent of the online examination system, the new method of conducting assessments has come to fore. While the traditional methodology persists, but in place of an examination center, students log into an online system to take the test and share their responses. The evaluation and circulation of results are carried out by the assessor.

Our web based online exam management system is capable of arranging exams and

taking tests. We mainly focus on taking CQ and MCQ exams arranged by Teachers.

There are two types of user roles in our system.

3.1 Teacher

A Teacher is the Administration user of our system where teacher will be able to set exams.

3.1.1 Course Create

Teacher will create course and an automatic invitation will be sent to the students of the department.

3.1.2 Question Create

Teachers are able to create cq and mcq questions for the exams.

3.1.3 Exam Create

Teacher will create exam after questions are created and the exam Notification will be sent to all the students assigned in the course.

3.1.4 CQ Examine

There is a manually cq exam paper checking feature for teacher. Then result of cq will be sent to the student automatically.

3.1.5 Marksheet Check

Teacher can see all the results of the students and analyse the performance of the students and take necessary steps to make better.

3.1.6 Feedback Check

Teachers can check the report or feedback of students on every questions.

3.1.7 Result Print

Teacher can print the marksheet of the specific exam for offline distribution.

3.2 Student

Another user is Student. Students will be able to participate in the exams arrranged by Teachers.

3.2.1 Real-Time Exam

Students will be able to participate in realtime exam.

3.2.2 Revise Previous Exam

Students will be able to see the previous exams and questions with answers. If they had attended in the exam then they will also be able to check their wrong answers. It will be helpful for them to check their previous exams.

3.2.3 Realtime Notifications

Students can get invitation for courses instantly. They will be notify when any exam will be created in their course and the result publication of cq exam will also be notified.

3.2.4 Feedback

In exam time, if they find something wrong, they can report about it.

3.2.5 Late Participate

We have made the online examination system in such a way that students will be able to participate in the exam if they are late. They can submit the answer before the given time on the questions and go to the next one.That's how one can solve all questions even they attend in the exam late.

3.3 Exam

The main feature of our application is making the examination system online.In our app student will be able to participate in the exams online set by their teachers.

Types of exams :

- Multiple Choice Question (MCQ) Exam
- Creative Question (CQ) Exam

3.3.1 Question features

- MCQ and CQ Questions
- Specific time limit for each questions
- Specific mark set for each questions
- Report system for each questions of a exam

3.3.2 Exam Features

- Real-Time exam
- Set time and Date for upcoming Exams
- Automatic marking system for mcq questions
- Manual marking system for cq questions
- Secure exam access and answer submission
- Countdown clock in examination
- Disable copy and paste in question and answer section
- Focus change alert in exam time
- Instant result show for mcq questions
- Realtime Notifications for exams and cq exam results

4 User Interface Specifications

4.1 Authentication

4.1.1 Signup

First of all any user needs an verified account to use our application. So he need to signup for creating a new account where he have to use his verified email address given by his/her university. Without a valid email address no one will be able to create any user account.The signup information differs from user type to user type.

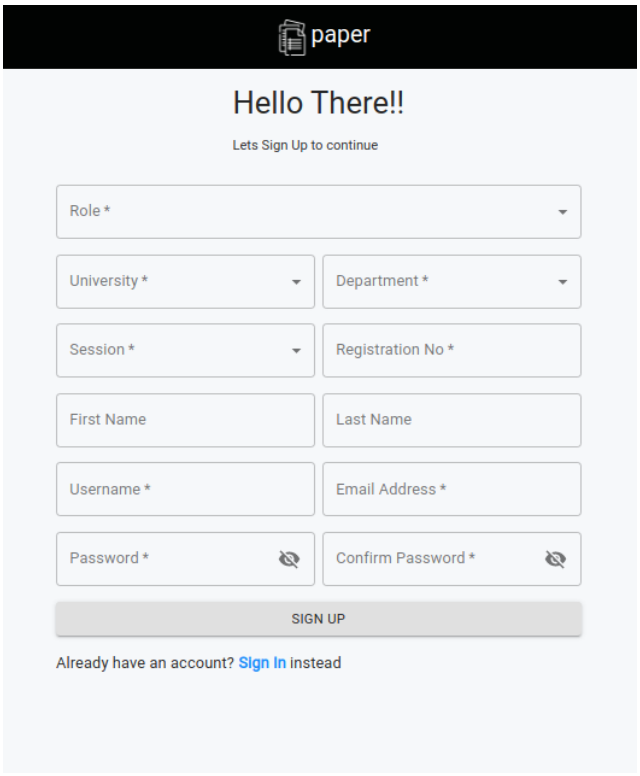

Figure 1: Signup Page

System will notify if any errors occurs such as

- Email address not authorized
- Username already exists
- Email address already exists
- Password didn't matched

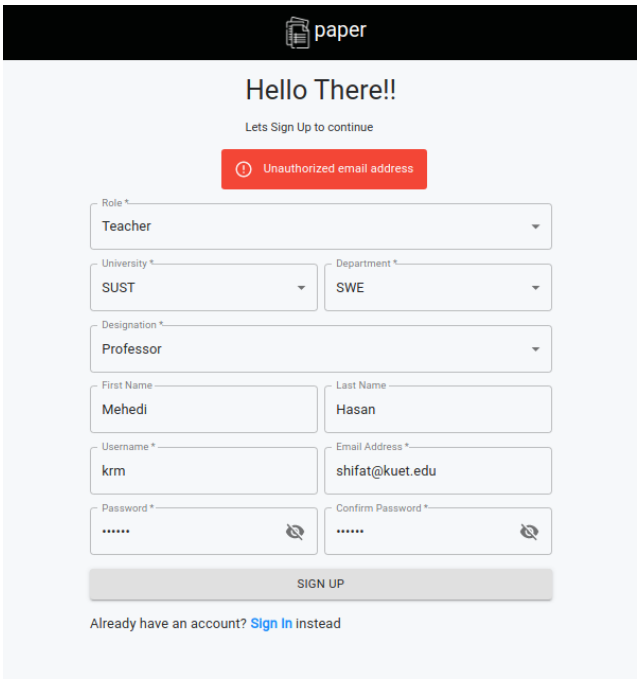

Figure 2: Signup Page : Unauthorized email address error

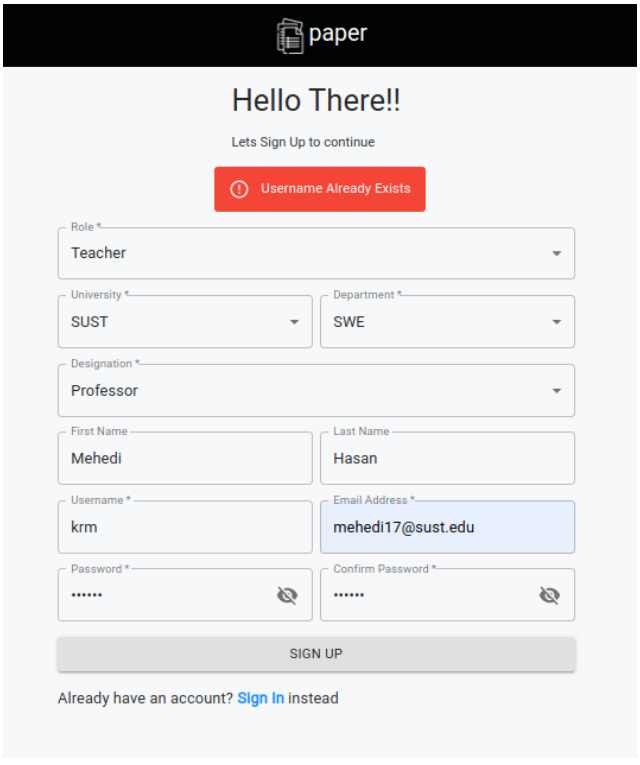

Figure 3: Signup Page : Already used username error

4.1.2 Login

When a user has a verified account they can login whenever he/she wish. Our login system. In order to login, user needs to input his/her email and password.

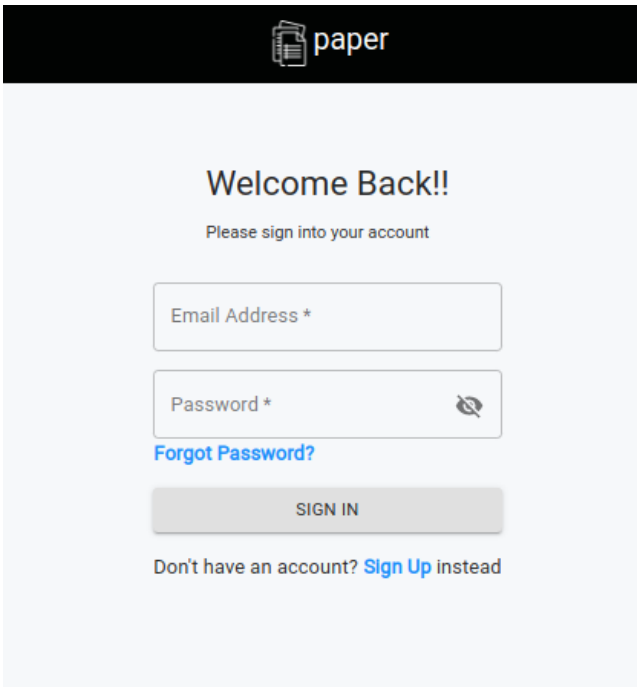

Figure 4: Login Page

Authentication verify and errors show :

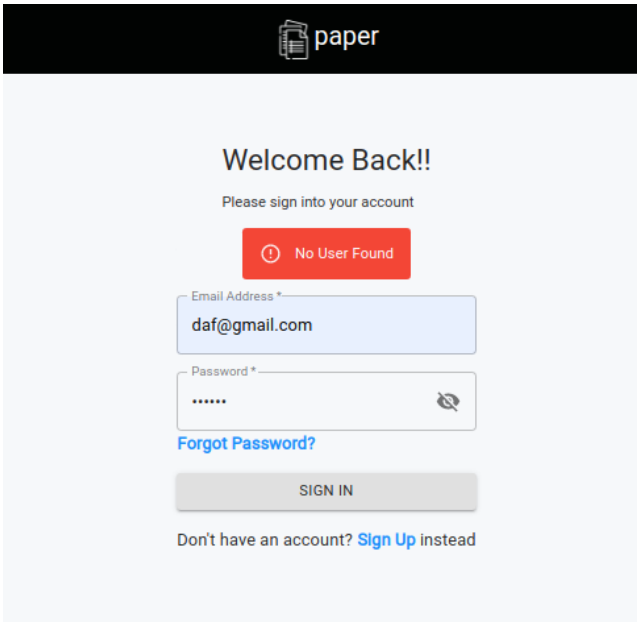

Figure 5: Login Page : Invalid email address error

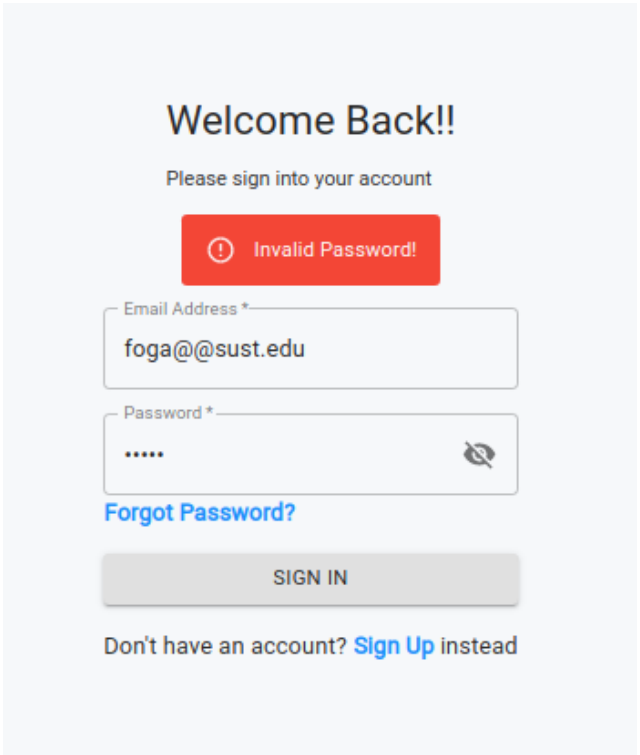

Figure 6: Login Page : wrong password error

4.2 Home Page

It is a common page for all users.In home page an authorized user will show three sections.They are -

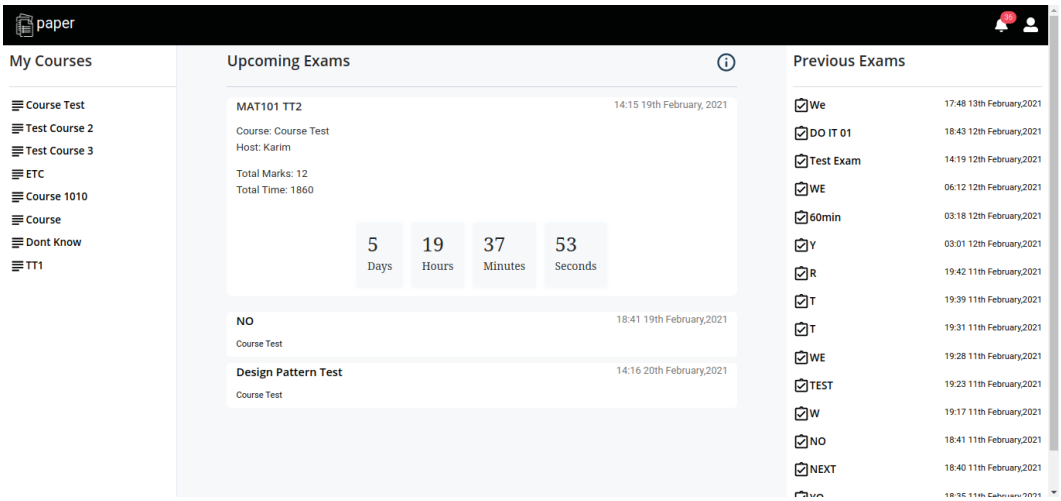

Figure 7: Home Page View

4.2.1 Courses

In course section, user interfaces will be different for different roles users. The students will see a list of all joined courses where the teacher will see the courses he created and a button to create more courses.

4.2.2 Course Creation

Teachers are able to create new courses for the students.

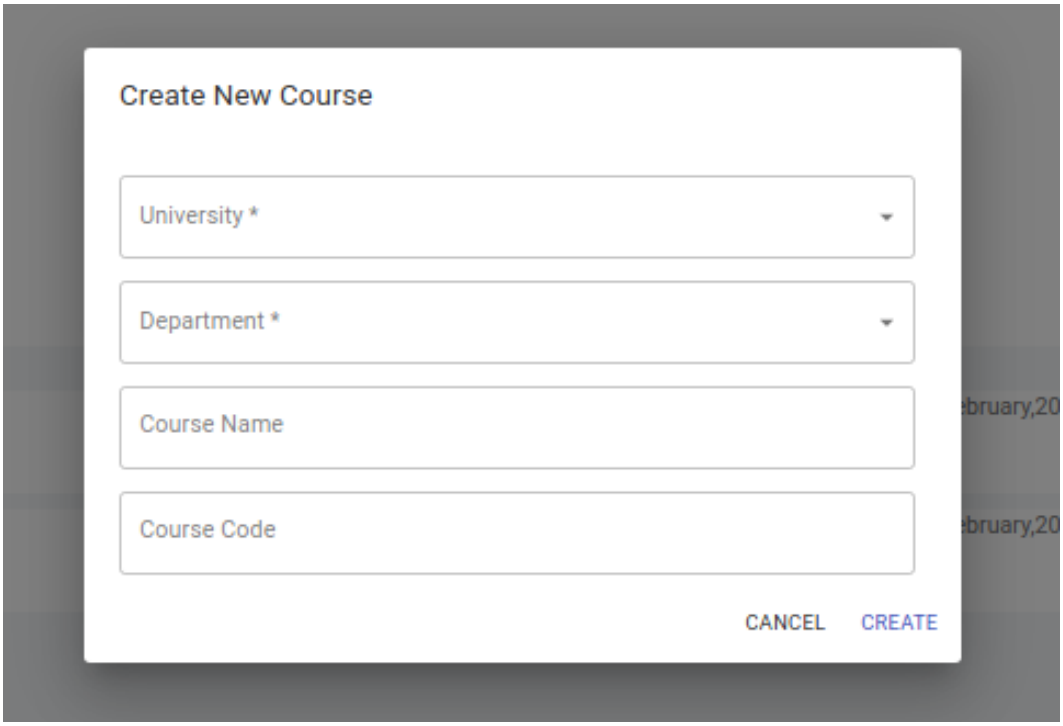

Figure 8: Teacher View: Course Creation

4.2.3 Countdown Timer

At the focus of the home page, there is a Countdown clock to chase the time of most upcoming exam.

4.2.4 Upcoming Exams

There is a list of all upcoming exams at the home page.

4.2.5 Previous Exams

At the left side of the screen there is a section for all previous exams.

4.3 Upcoming Exam Page

Detail information for upcoming exams can be checked in this page.

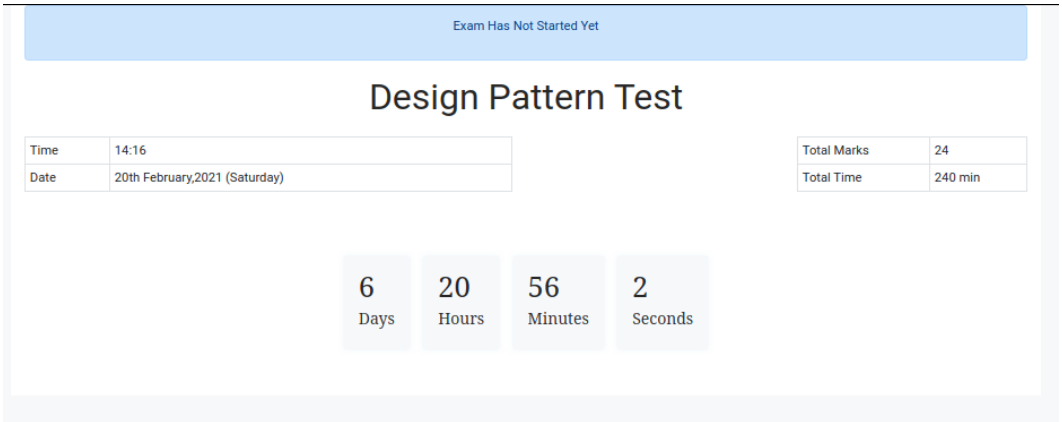

Figure 9: Upcoming Exam Page

4.4 Previous Exam Page

The user interface of previous exam page is shown differently for a teacher and a student.

4.4.1 Exam Detail

Student will be able to see the exam detail info such as

- Exam name
- Time
- Date
- Mark
- Obtained mark
- Participated or not

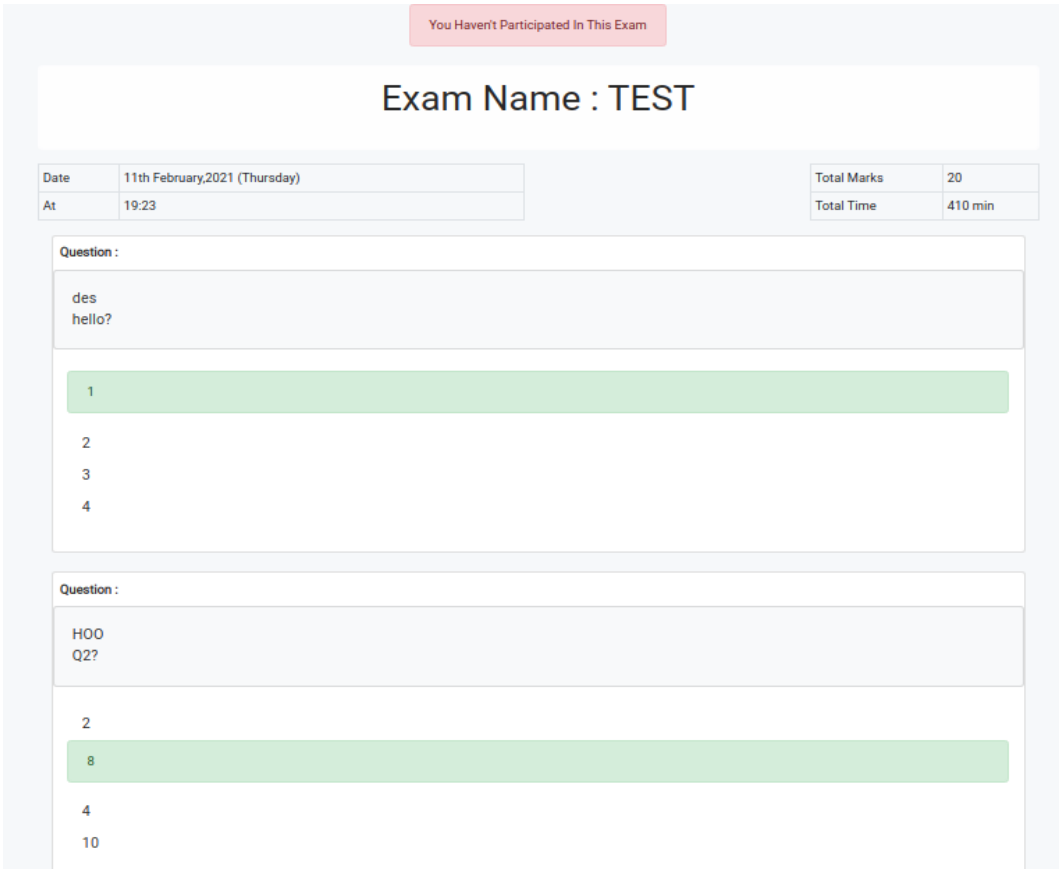

Figure 10: Student View : Previous MCQ Exam

4.4.2 My Answers

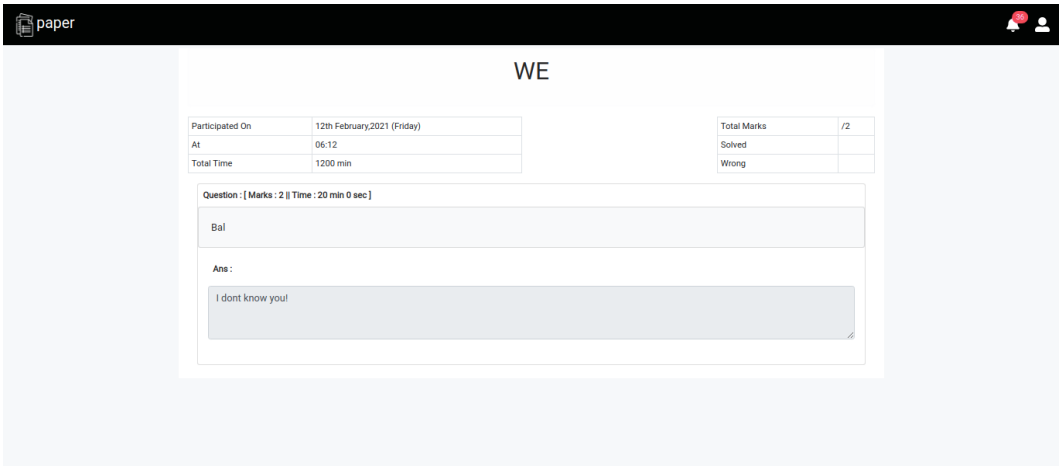

Figure 11: Student View : Previous CQ Exam

4.4.3 Marksheet

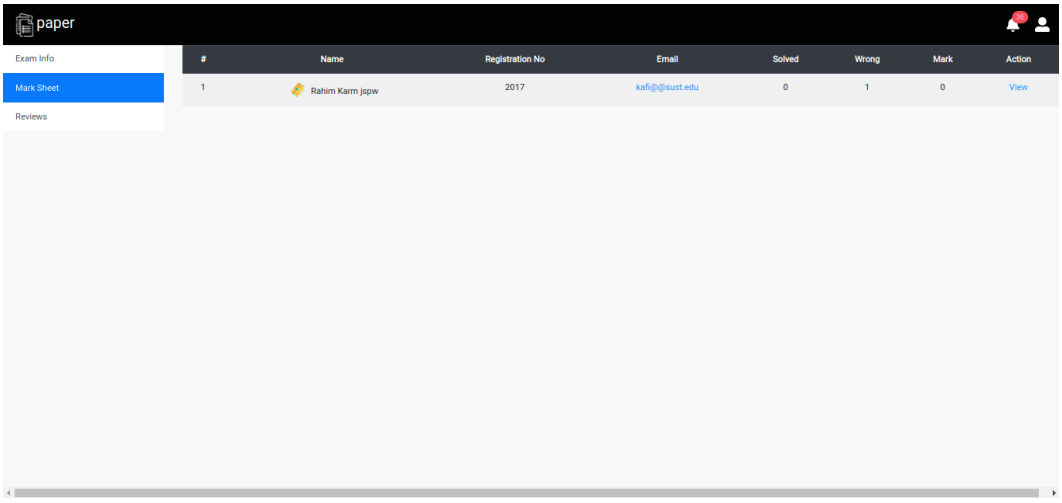

Figure 12: Teacher View: Marksheet of Exam

4.4.4 Reports

Students are able to report or give feedback about questions while exam. The teacher will be able to see the reports so that he can improve these types of problems.

4.5 Course Page

4.5.1 Course Info

A Detail information of the course is shown here for the user.

4.5.2 Exam Creation

Only the teacher is able to create exam for courses. There are two types of exams a teacher can create.

- MCQ
- CQ

First of all, teacher has to make questions for the exam. Then he set name, time and date for the exam and confirm it.

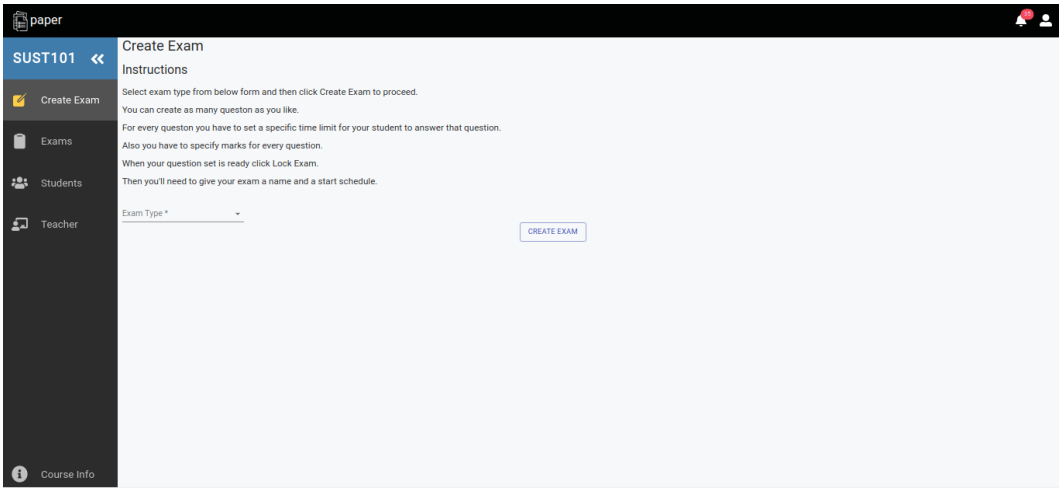

Figure 13: Teacher View: Exam Type Select

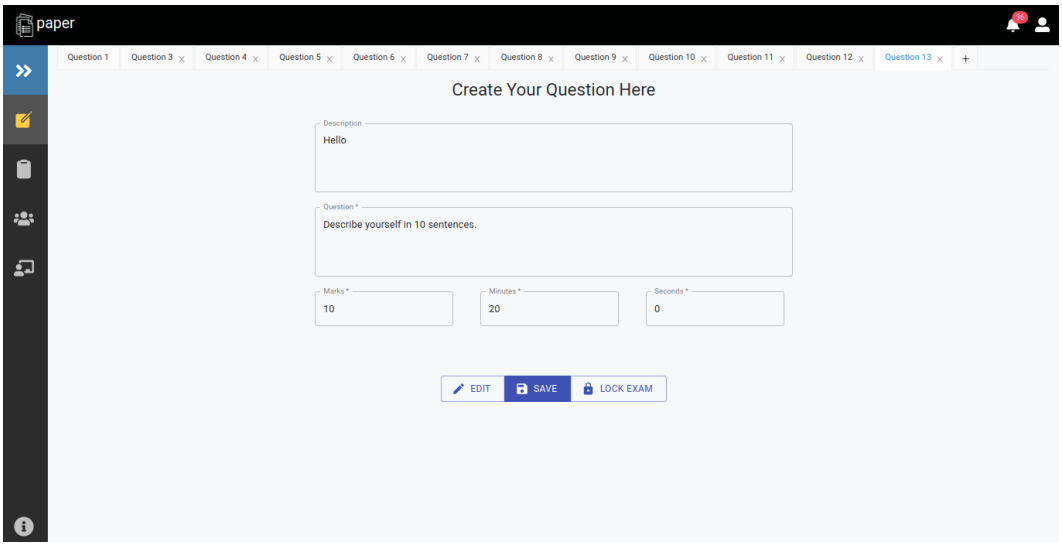

Figure 14: Teacher View: CQ Question Create

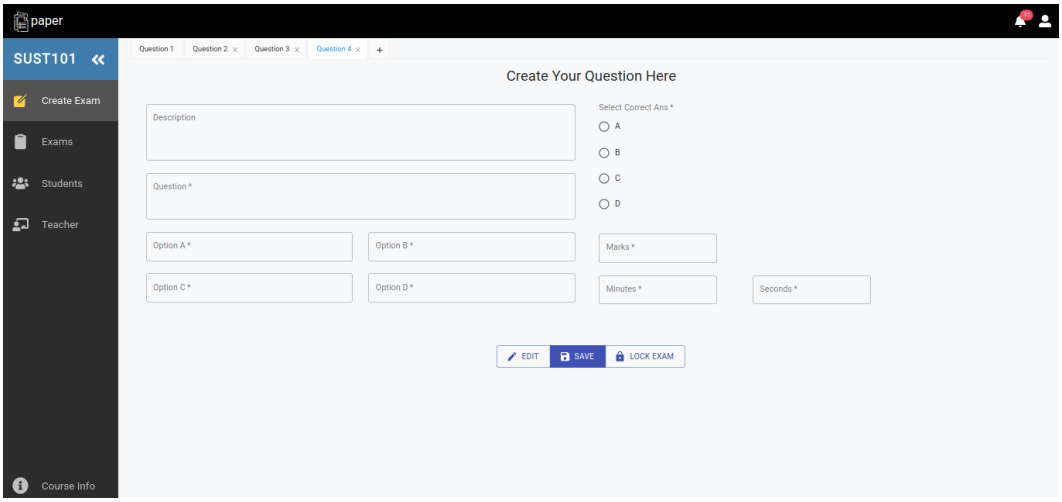

Figure 15: Teacher View: MCQ Question Create

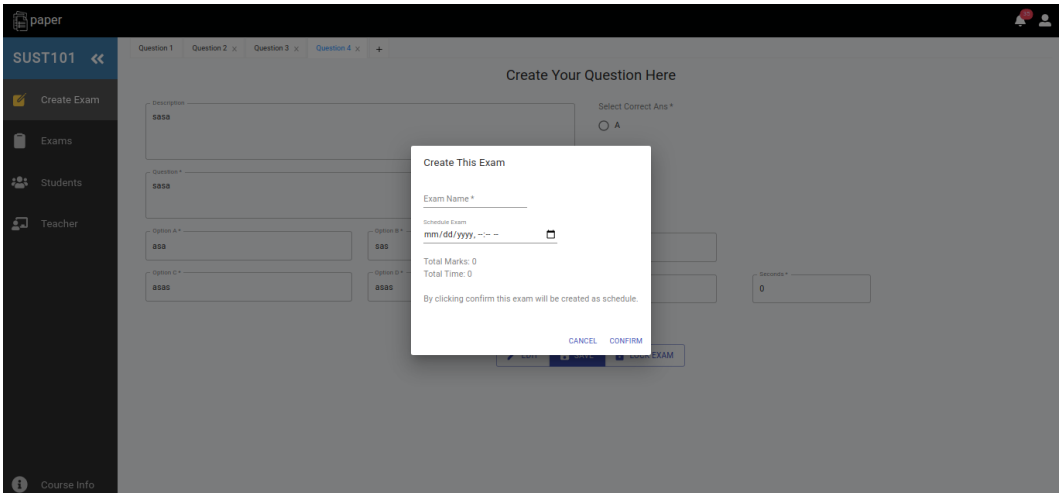

Figure 16: Teacher View: Exam Creation

4.5.3 All Exams

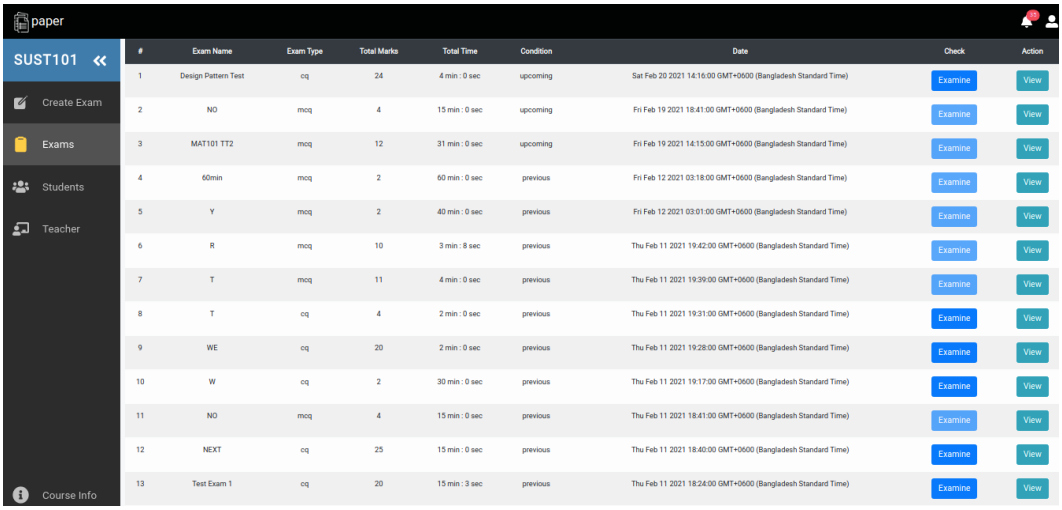

Figure 17: Course Page : All Exams in the course

4.5.4 Students

A table of all students information assigned in the course are show here so that teacher can overview the students performance.

4.5.5 Teacher Info

Students will be able to check the detail information of the teacher who is taking the course.

4.6 Real-time Exam

This is an interactive user interface for live exam.

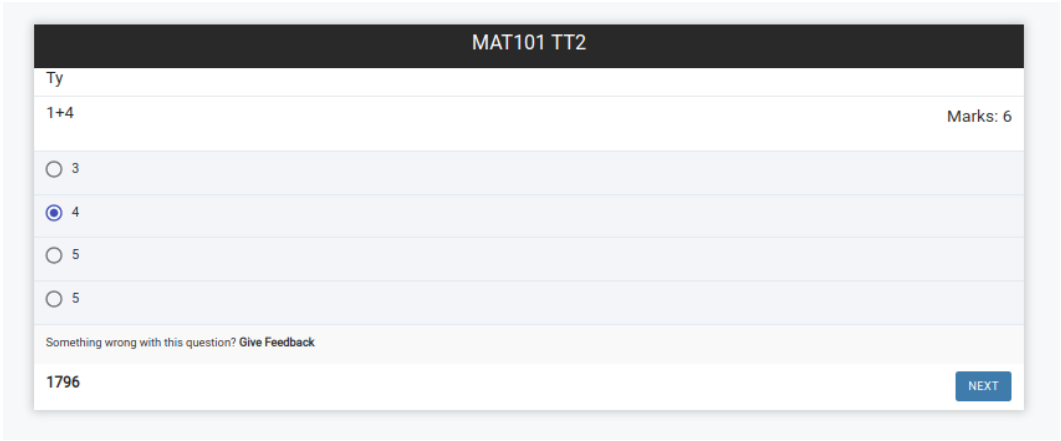

Figure 18: Student View: Realtime MCQ Exam

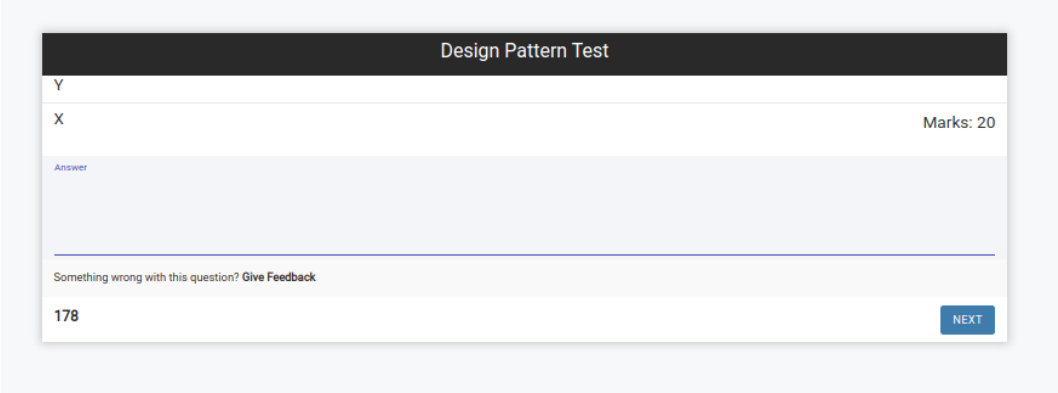

Figure 19: Student View: Realtime CQ Exam

4.7 Examine CQ

Teacher can check cq exams manually and give mark on them.

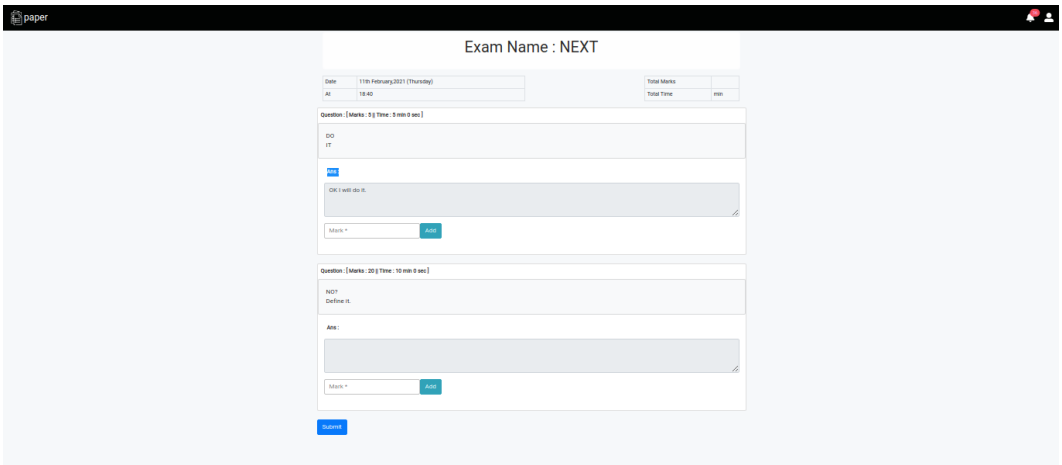

Figure 20: Teacher View: CQ Examine

4.8 Notifications

Users are able to see notifications of -

- Course Creation
- Exam notifications
- CQ result publish

Figure 21: Realtime Notification : Course Invitation

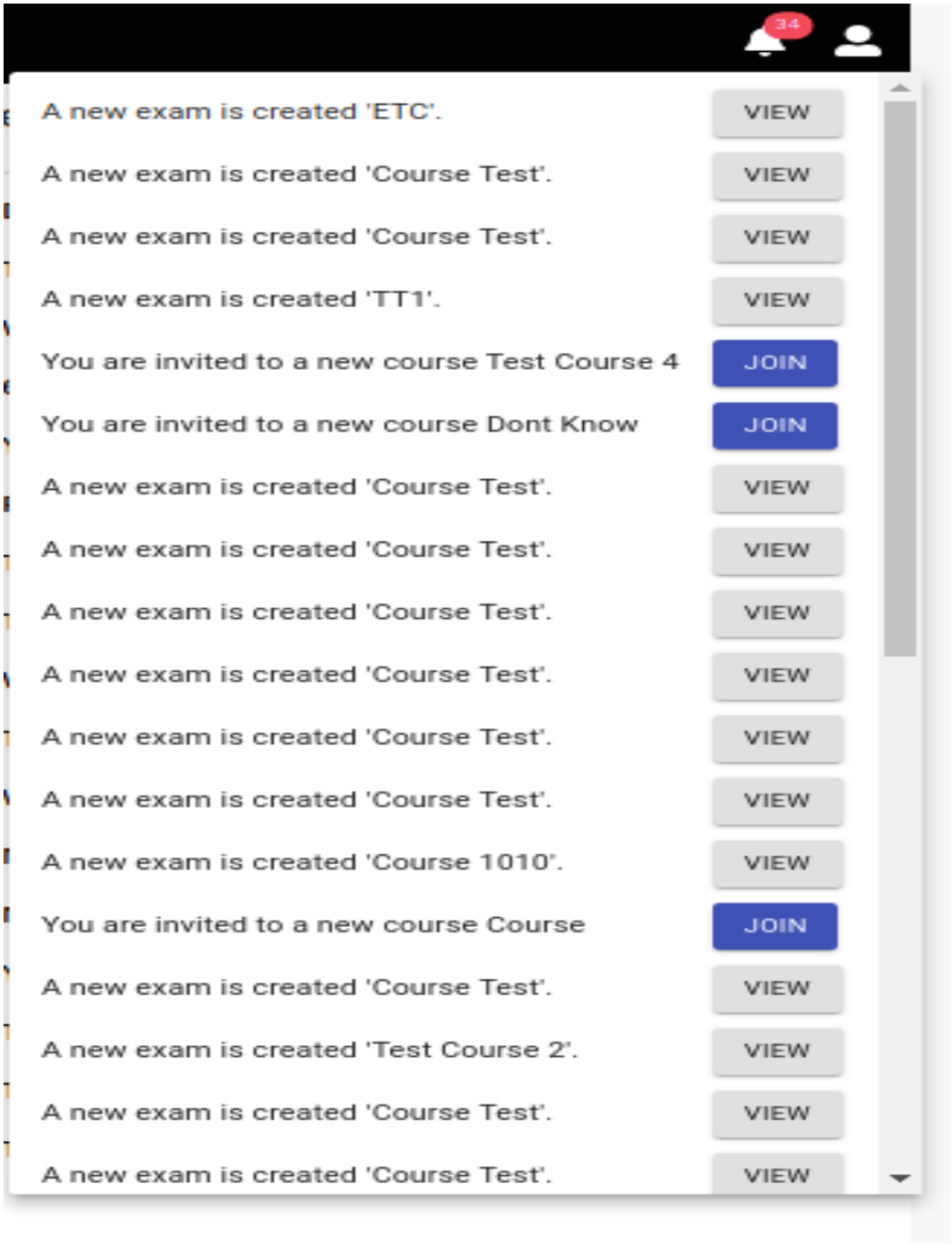

Figure 22: Student View: All Notifications

5 System Design

5.1 Use Case Diagram

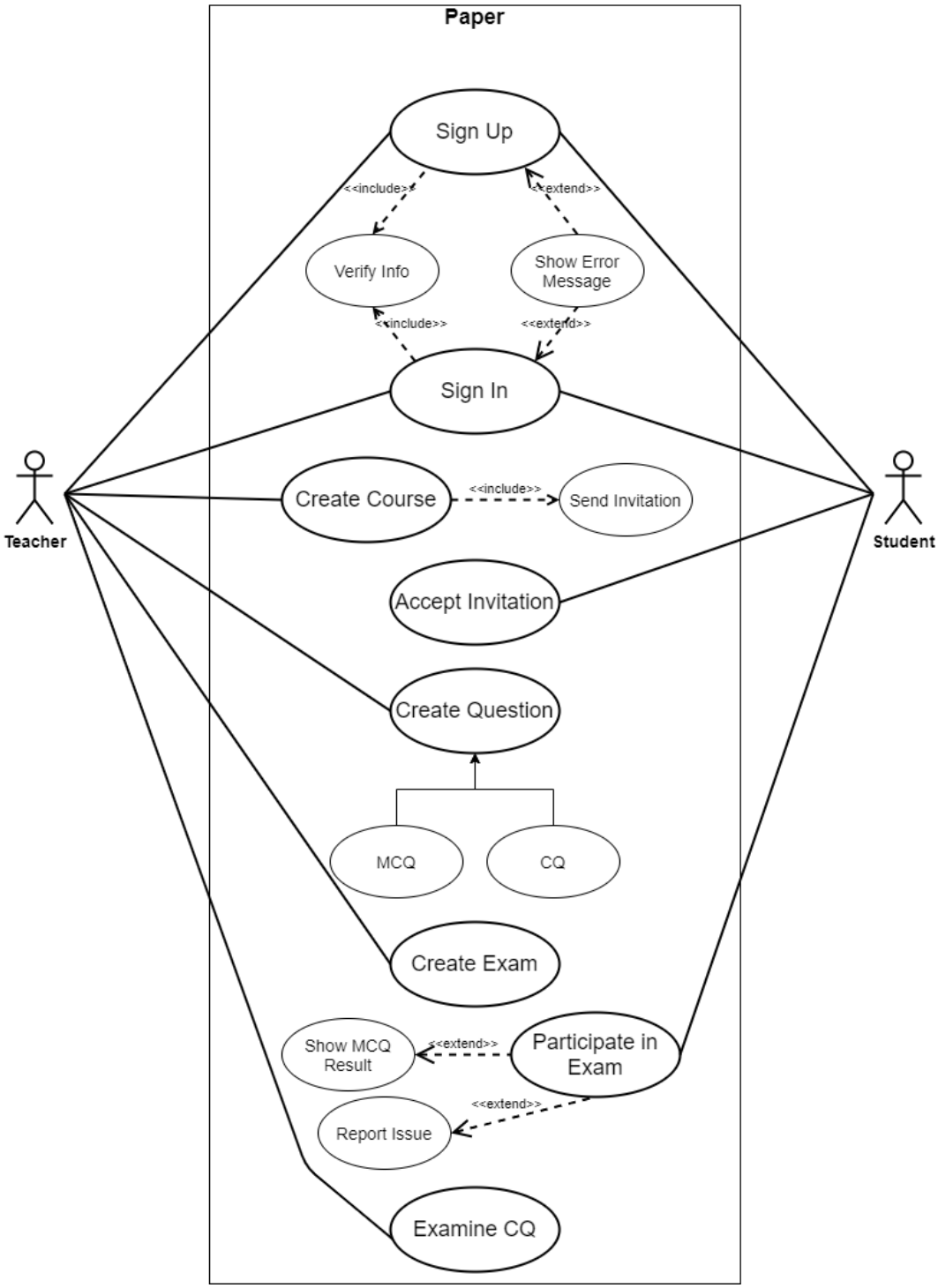

Figure 23: Use Case Diagram

5.2 Context Diagram

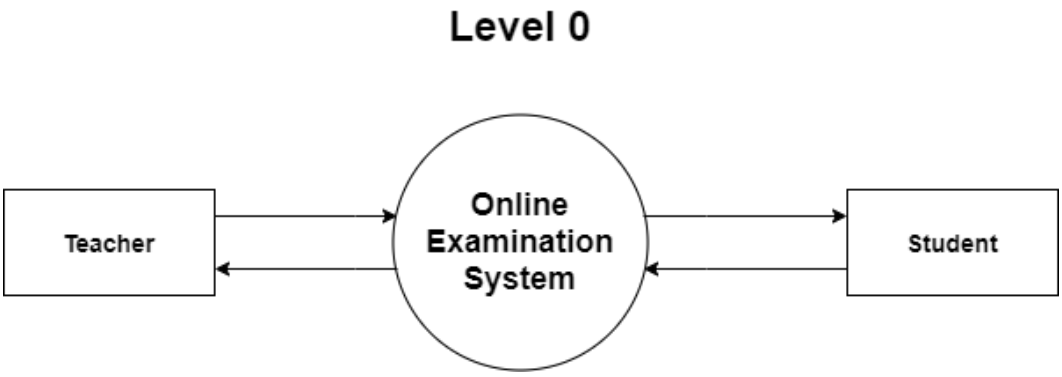

Figure 24: Context Diagram

5.3 Data Flow Diagram

5.4 Entity Relationship Diagram

6 System Architecture

6.1 Front-End

In frontend we have used reactjs which is a framework of javascript to design user interfaces.Here we have followed Component Based Architecture.

6.2 Back-end

In backend development we created restful api using javascript backend framework Express. We followed Model View Controller (MVC) here.

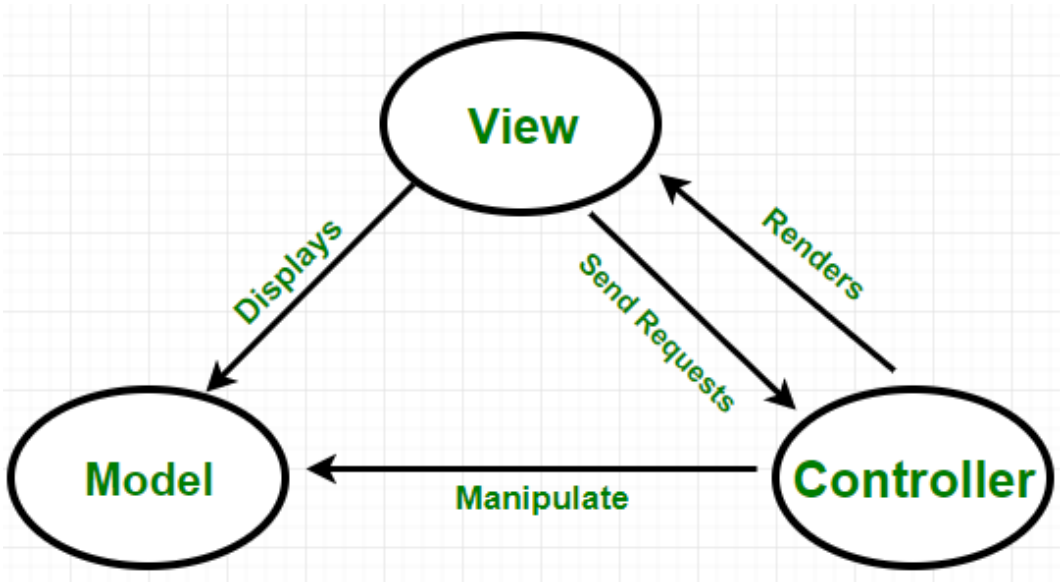

Figure 25: Model View Controller Architecture

7 System Implementation

We can separate the whole system Implementation into two sections.

- Frontend
- Backend

We have used ExpressJs as server side framework for backend-server,React for frontend, MongoDB for database. The frontend and backend are developed separately and merged through Restful api. We have designed api end-points to fetch data from database to frontend. We have also implemented Sokcet-IO for realtime notification integration.

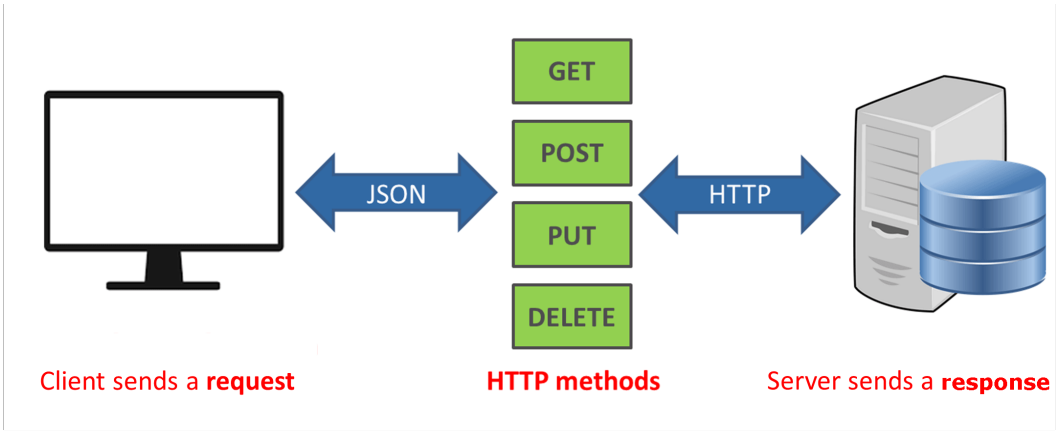

Figure 26: Backend-Frontend Merged Through API

7.1 Authentication

For authentication we have used Javascript Web Token (jwt). The backend send a token for every users when they login or signup in the frontend. The frontend saves the token and use it everytime it sends request to the server. That's how the system becomes more secure.

```
axios.post(url,data, {
    headers: {
        'authorization': your_token,
        'Accept' : 'application/json',
        'Content-Type': 'application/json'
    }
})
.then(response => {
    // return  response;
})
.catch((error) => {
    //return  error;
});
```

Figure 27: Frontend : React Api call with jwt token

7.2 Pattern Of Code

7.2.1 Frontend

We have divided the whole system into small components. First we define the components and them merge them with the system.Our component based system :

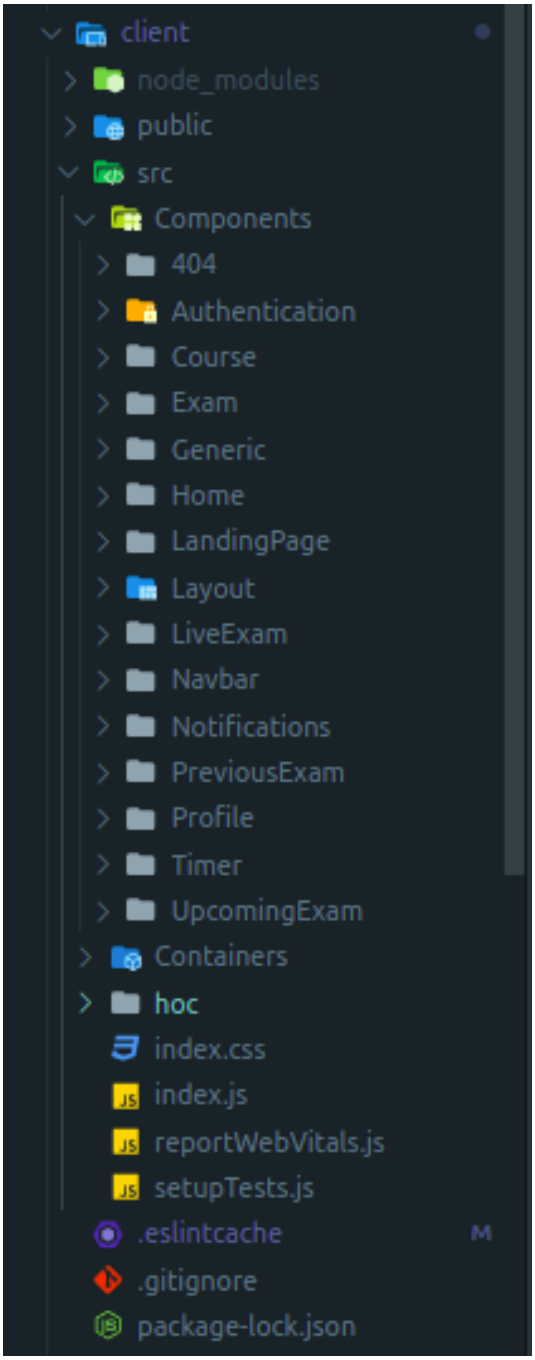

Figure 28: Frontend components based System Structure

In frontend we have called api to fetch data from database.

```
if (userdata) {
  axios
    .get(`${userdata.role}/user/${userdata.id}`)
    .then((result) => {
      setloginStatus(result.data.status);
      setUserInfo(result.data.result.data);
      localStorage.setItem("data", JSON.stringify(result.data.result.data));
      console.log("UserInfo api call", result);      You, 19 days ago • api course and exam info ro
    })
    .catch((error) => {
      console.log(error);
      setloginStatus("Failed");
      console.log("Error api call", error);
    });
} else setloginStatus("Failed");
```

Figure 29: Frontend Api calling

7.2.2 Backend

In backend we have separated the controllers, routes and models.All the end-points and routers are defined in routes folder.The models folders are for database models and controller controls the post and get method and send responses to the frontend.

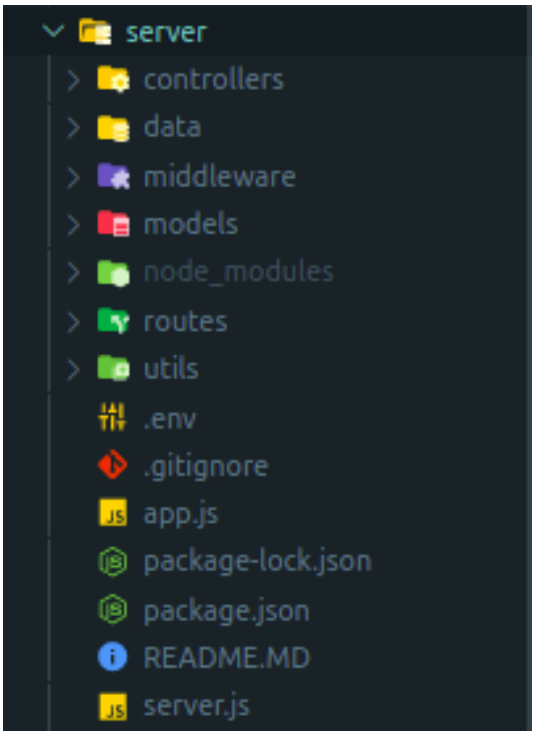

Figure 30: Backend File Structure

```
You, 22 days ago | 1 author (You)
const mongoose = require("mongoose");
const validator = require("validator");
const Schema = mongoose.Schema;
const universitySchema = new Schema({
  name: {
    type: String,
    require: true,
    unique: true,
  },

  shortform: {
    type: String,
  },

  emails: [
    {
      email: {
        type: String,
        unique: true,
        lowercase: true,
        unique: true,
      },
    },
  ],
  departments: [
    {
      name: {
        type: String,
        unique: false,
      },
      shortform: {
        type: String,
        unique: false,
      },
    },
  ],
});
universitySchema.plugin(require('mongoose-autopopulate'));
module.exports = mongoose.model("University", universitySchema);
```

Figure 31: Backend : Model Sample

```
const express = require("express");      You, 22 days ago * courses added in studen
const studentController = require("../controllers/student");
const authenticateJWT = require("../middleware/authenticateJWS");

const router = express.Router();

router.route("/user/:id").get(authenticateJWT, studentController.getStudent);

router
  .route("/user/edit/:id")
  .get(authenticateJWT, studentController.getEditStudent)
  .post(authenticateJWT, studentController.postEditStudent);

router
  .route("/course/add")
  .post(authenticateJWT, studentController.postCourseAdd);

router.route("/exam/:id").get(authenticateJWT, studentController.getExam);

router.route("/course/:id").get(authenticateJWT, studentController.getCourse);

router
  .route("/exam/mcq/submit/:id")
  .post(authenticateJWT, studentController.postMcqSubmit);

router
  .route("/exam/cq/submit/:id")
  .post(authenticateJWT, studentController.postCqSubmit);

router
  .route("/exam/mcq/submit/:id")
  .get(authenticateJWT, studentController.getMcqSubmit);

router
  .route("/exam/cq/submit/:id")
  .get(authenticateJWT, studentController.getCqSubmit);

module.exports = router;
```

Figure 32: Backend : Model Route

8 Testing And validation

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

Before releasing the software we have gone through some testing methods to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. We have classified our testing system into two catagories. They are -

8.1 Functional Testing

8.1.1 Unit Testing :

This is the first stage of testing our system where we divided our system into several units, functions and components. The purpose is to validate that each unit of the software code performs as expected. Unit Testing was done during the development (coding phase) of an application by our developers. It isolated a section of code and verify its correctness. Thats how we finished our unit testing.

8.1.2 Integration testing :

As our project consists of multiple software modules, coded by different programmers so we integrated the modules logically and tested as a group. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. We focused on checking data communication amongst these modules.

8.1.3 System testing:

After Integration testing we validated the complete and fully integrated software product.The purpose of system test is to evaluate the end-to-end system specifications.

In system testing we focused on -

- **Usability Testing** - mainly focused on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.
- **Load Testing** - we tested real-life loading scenarios such as

  - Home screen loading
  - Login, Signup loading
  - Next pages Loading
  - Course Join

- **Migration testing** - is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

8.1.4 Acceptance Testing :

This is the final level of testing. In Acceptance testing we determined that this application is ready to release as it meets all the requirements and validations.

8.2 Non-Functional Testing

In non functional testing we were testing the non functional requirements of the system such as security, performance, usability, scalability.

8.2.1 Load Testing

Our load testing involves testing the system's loading capacity. Loading capacity means more and more people can work on this system simultaneously.

8.2.2 Security Testing

The most focus on our system is the security as it is a system to take exams and we ensure the security most. We tested the account creation system so that no unauthorized user will be able to signup in our application. We also tested the exam time verification system too.

8.2.3 Reliability Testing

In reliability testing we assure that the system is running without fail under specified conditions and it must be run for a specific time and number of processes.

8.2.4 Efficiency Testing

We tested the Efficiency of our system if it becomes slow or not when more users are interacting with our system.

9 Future Enhancement

As our project is in a beginning stage, we have plans to add several features in future. Our plan includes features like -

- ### 9.1 Combined Exams

  Currently in our application MCQ and CQ exams are two seperate types of exam. But in future exams can have both CQ and MCQ types of questions.

- ### 9.2 Lab Exams

  For now our system is able to take theoretical exams but we are eager to implement lab exams such as coding related course exams online.

- ### 9.3 Realtime Chat

  Teachers and students will be able to chat for necessary information. This will reduce communication gaps between them.

- ### 9.4 Mobile Version

  We will release our application for mobile users as it will be more easy for students to participate in exam online.

10 Conclusion

This web based application will make robust changes in our education system.It will help the education system to continute our educational progress during lockdown and COVID-19 pandemic.It will save students from Session jam.

11 References

- Influence of COVID-19 confinement on students' performance in higher education
- The Effects of COVID-19 on Academic Activities and Surgical Education in Italy