DBMS MINI PROJECT REPORT

ONLINE COURSE PORTAL

GROUP NO. 1

GROUP MEMBERS :

Nimish Upadhyay (IIT2019113)

Abhinav (IIT2019098)

Sangam Barnwal (IIT2019094)

Nikil Chouriya (IIT2019120)

Github link

TABLE OF CONTENT

1. Abstract:

1) This project aims at creating a Courses portal for a campus/organization. This allows registered users of the system to join a course available in the site and access the materials published for the course.
2) People can register themselves as students of a course or Faculty for a course. When a person registers himself as a Faculty, an approval mechanism should be triggered which sends an email to the Administrator for approving the person as a Faculty.
3) There will be an admin approval page where admin can approve the faculty members for the course. The course home page should contain the title of the course and a brief description.
4) There will be a discussion board for each course where students can interact, an announcement section, which contains the latest announcements, and a course content section which gives the links for the material available for the course.
5) For faculty members there will be an extra link for uploading the course content in a zip file format.
6) There should be a mechanism for the faculty members to create a test for the course specifying the test title and a set of multiple-choice questions and duration of time of the Test.

2. Technology Used:

1) Front-End Technologies :-

i) **HTML** : HTML provides a framework to all our web pages allowing us to create web pages with ease in our project.

ii) **CSS** : With the help of CSS ,we have added style to our html framework and with help of bootstrap libraries we have used standard and unique classes to add style all our web pages

iii) **JavaScript** : JS helped us to add functionality in our pages like drop down menus ,timer in test page part and button's that tell which question is attempted.

iv) **eJS :** Embedded JS provided us the framework to communicate between Frontend and backend via JSON.

2) Back-End Technologies :-

i) **Node JS :** With the help of node js ,we created our whole Backend server allowing us to host our server in a web browser and manipulate databases using the mongoose package ,we did our all calculations part using this technology . All posts and get requests are handled by node js.

ii) **MongoDB :** As the name suggests it is the database we have used which is fast, flexible and reliant when working with Node js. We can easily manipulate data . Unlike Mysql we don't need to create collections/Schema it will get automatically created, thus easy to handle.

3. AIM:

1) Our project aims at creating an Online course portal In which Teacher/Instructor Can create a course ,In which he can post description about the course ,He can post announcements related to the course, student can post query and doubts regarding the post.
2) Whenever a teacher registers for the course he has to get authenticated by the Administrator ,If the Administrator doesn't authenticate him,he cannot login as a Teacher.
3) Faculty can post exam (MCQ exam) which is given by the student who are enrolled to this course, After the Test ends they can see detailed analysis of their response of the that test like pie chart, percentage and teacher can see the responses of all the students who gave exam and evaluate on the basis of No. of correct Answer.
4) And to reduce cheating by adding functionality like Student cannot change tabs while giving tests.

4. Schema Table,Normalization and ER Diagram

SCHEMA TABLE

1) **ADMIN** :-

   i) Email (primary key) : String

   ii) Password : String

2) **TEACHER** :-

   i) Email (primary key) : String

   ii) Password : String

   iii) First Name : String

   iv) Last Name : String

   v) Address : String

   vi) Zip code : String

3) **STUDENT** : -

   i) Email (primary key) : String

   ii) Password : String

   iii) First Name : String

   iv) Last Name : String

   v) Address : String

   vi) Zip code : String

4) **COURSE** : -

   i) Course_id (primary key) : String

   ii) Name : String

   iii) Teacher_id : String

5) **ANNOUNCEMENT** :-

   i) ID (primary key) : String

   ii) Course_ID : String

   iii) Head : String

   iv) Body : String

6) **TEST** :-

   i) ID (primary key) : String

   ii) Course_ID : String

   iii) Question_set : String

   iv) Answer : String

7) RESPONSE :-

   i) Test_id : String

   ii) Student_id : String

   iii) Total Marks : Number

   iv) Correct_answers : Number

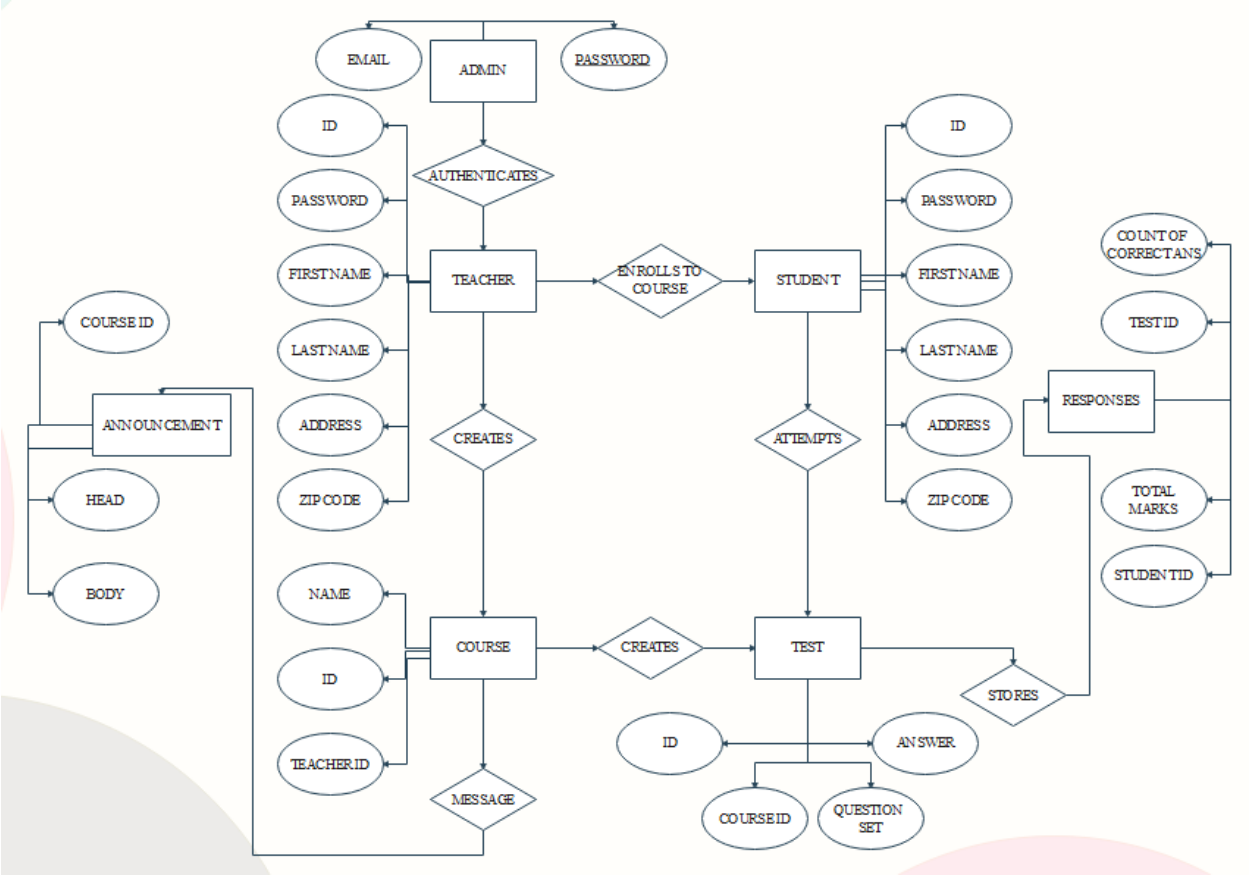
   (primary key is Test_id,Student_id)

NORMALIZATION :-

1) In our schema all attributes are single valued attributes which makes all the tables in 1NF.
2) As all the tables have single attributes as primary key which makes them in 2NF ,as their are partial dependencies not possible.
3) As all Tables have single attributes as primary key that rule out the situation of transitive dependencies ,thus making them 3NF.

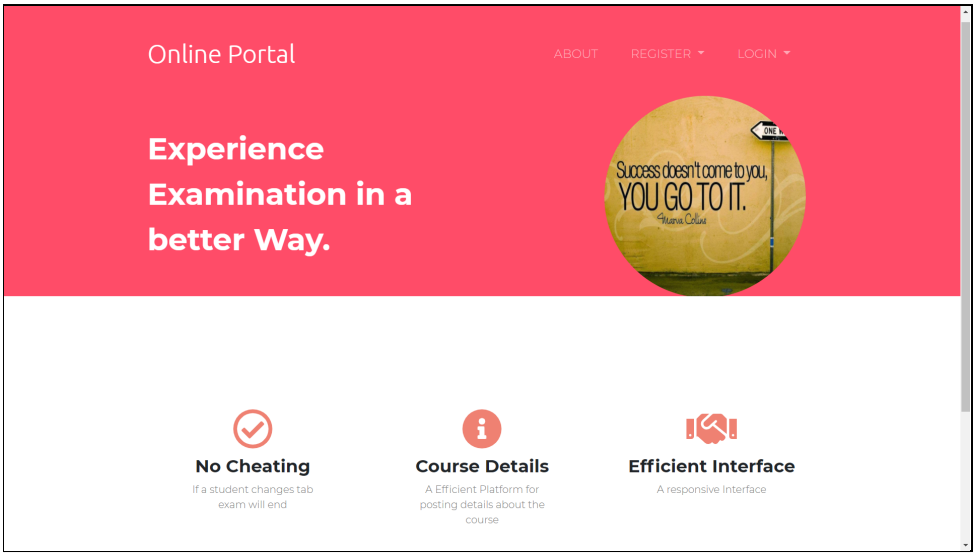
   Thus all our tables are in 3NF.

E-R DIAGRAM


EMAIL
ADMIN
PASSWORD
ID
AUTHENTICATES
PASSWORD
FIRSTNAME
TEACHER
ENROLLS TO COURSE
STUDENT
LASTNAME
ADDRESS
CREATES
ATTEMPTS
ZIPCODE
COURSE ID
ANNOUNCEMENT
HEAD
BODY
NAME
ID
TEACHER ID
COURSE
CREATES
TEST
MESSAGE
ID
PASSWORD
FIRSTNAME
LASTNAME
ADDRESS
ZIPCODE
COUNT OF CORRECT ANS
TEST ID
RESPONSES
TOTAL MARKS
STUDENT ID
STORES
ID
ANSWER
COURSE ID
QUESTION SET

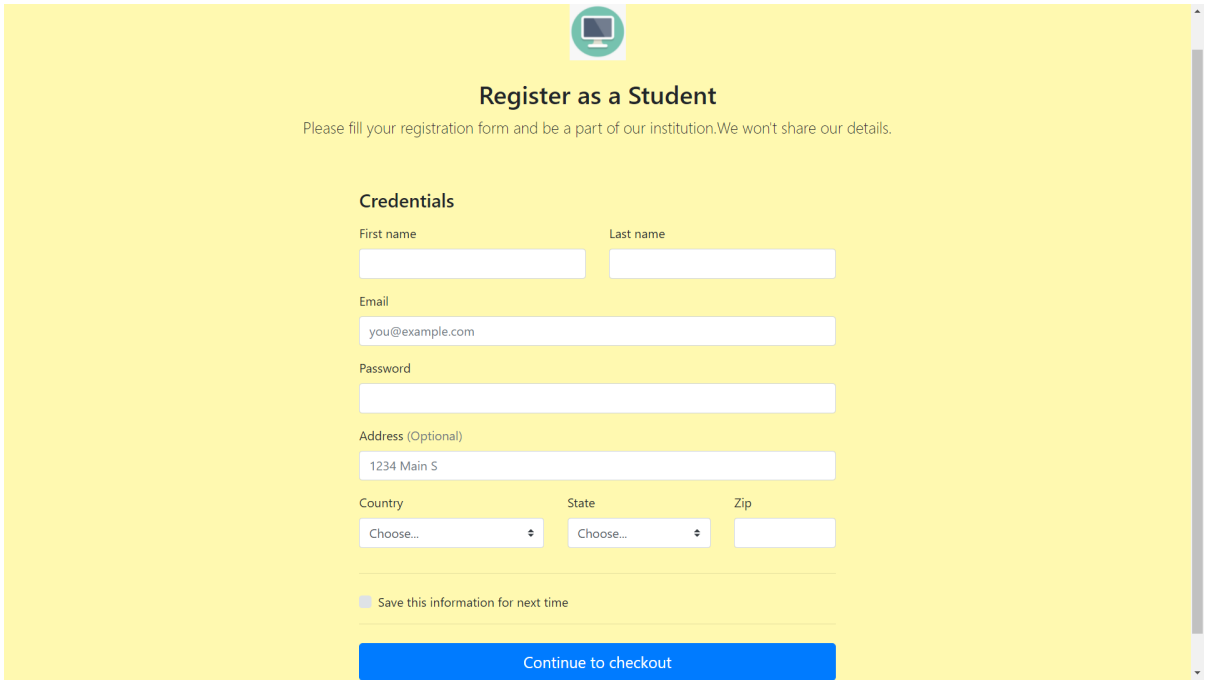
5. Screens:

1) **Intro Page:** it is the first page of our server which tells the user weather he wants to register ,login or see information about.

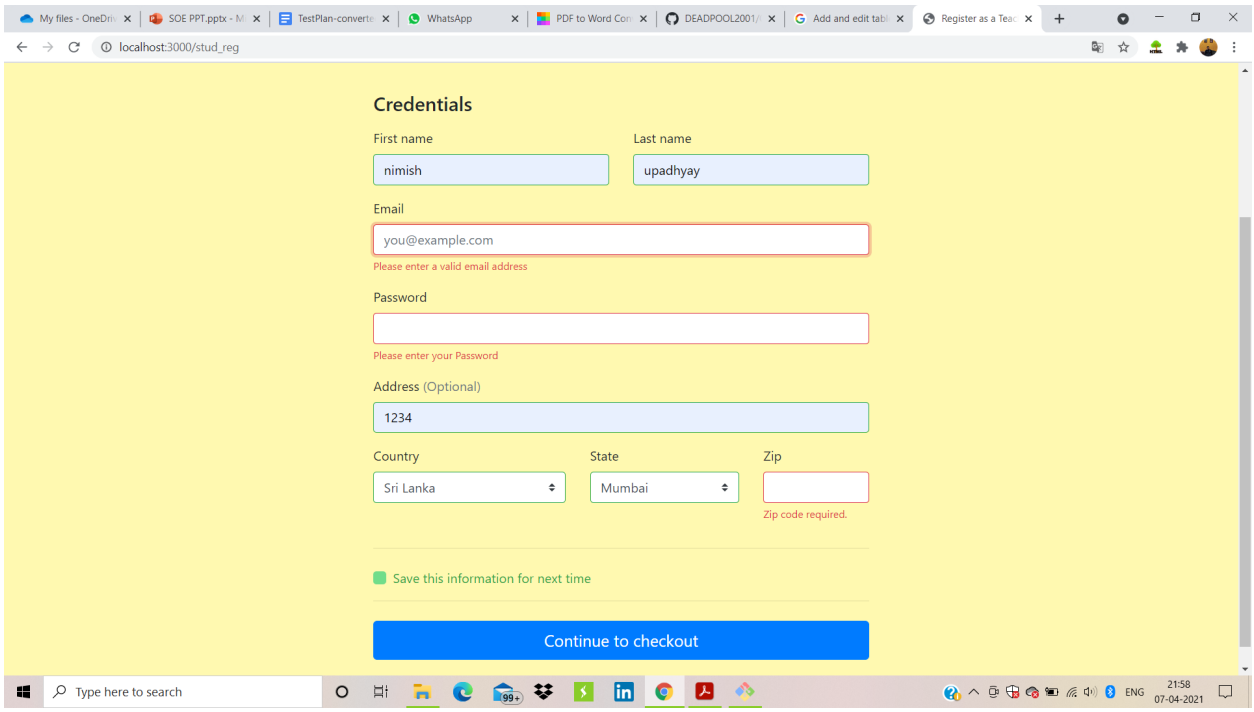


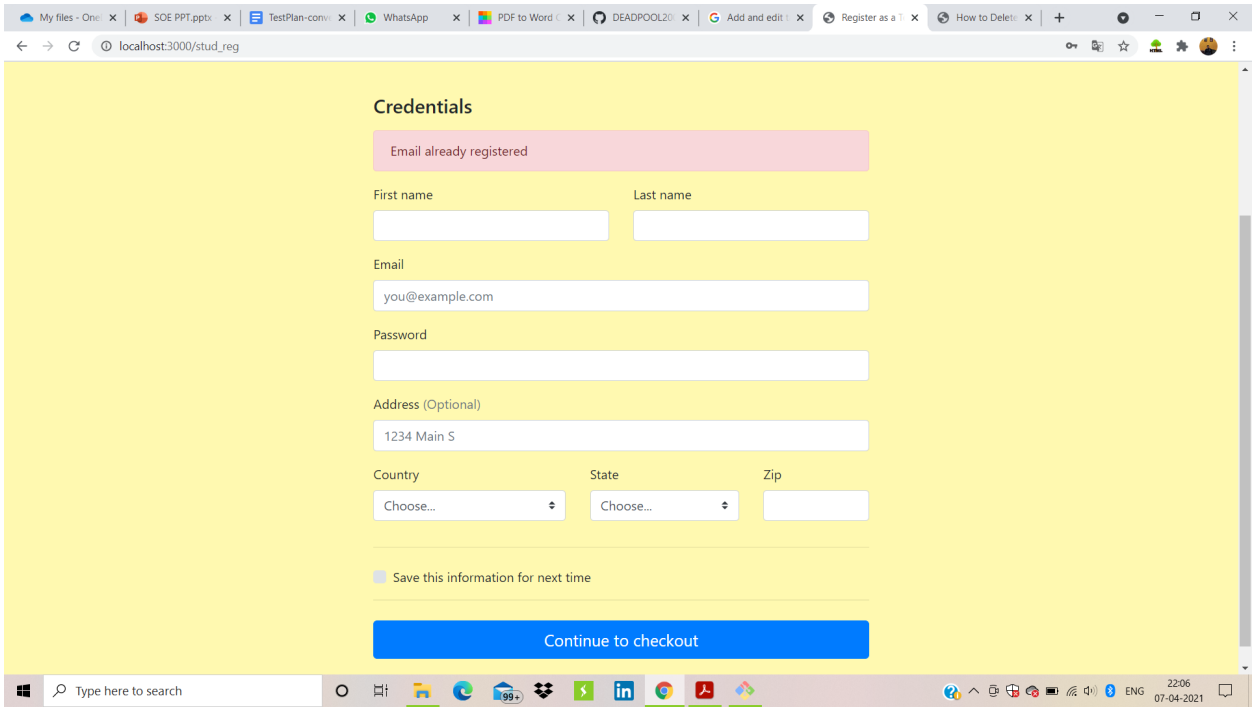
2) **Registration Page:** When we click on register, it shows us two options, to register as a student, or teacher. Let's take a look at each page.

Student registration Page:



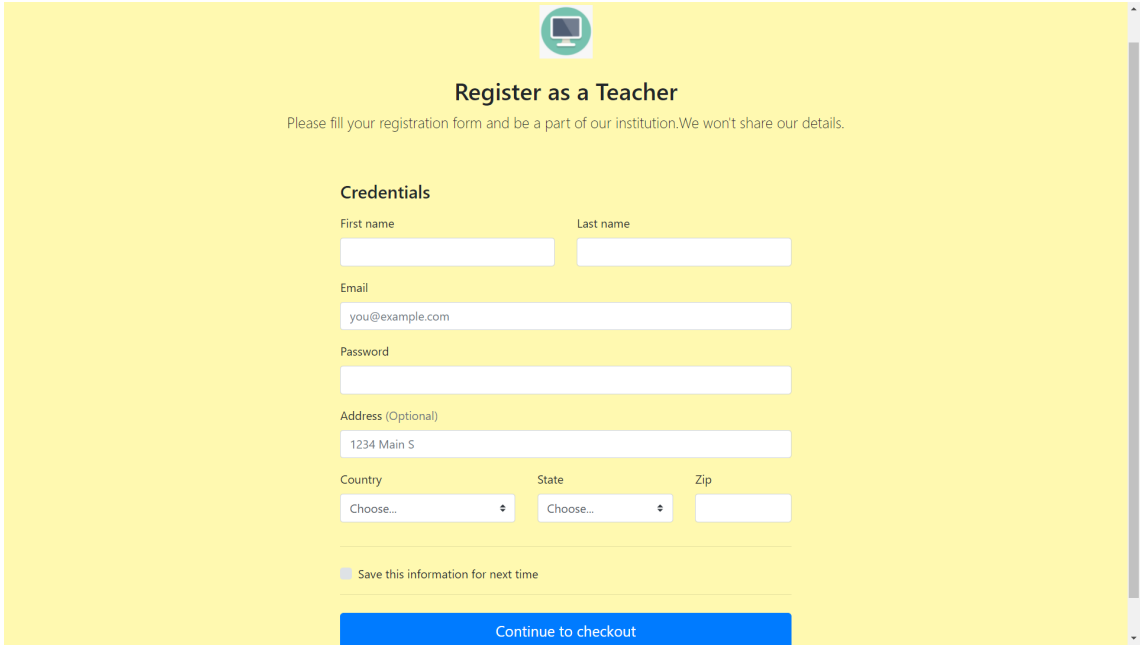
If we end an invalid email address or an already registered email, it displays us a message:





If we enter valid data then our gets saved in student database.

Teacher Registration Page:

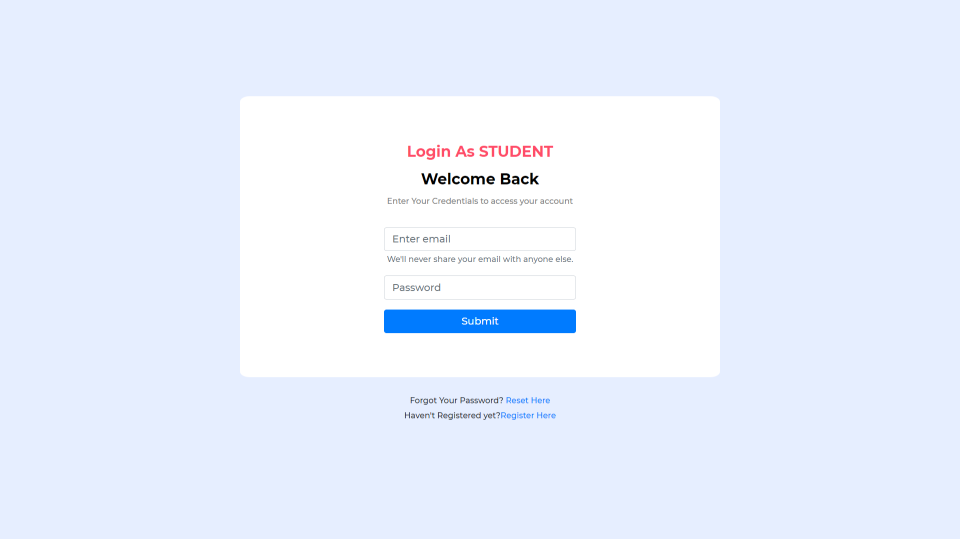


In case of incorrect or already registered email, it displays the message the same way it did in student registration. else if we enter valid data then data gets saved in the waiting list of the teacher schema.

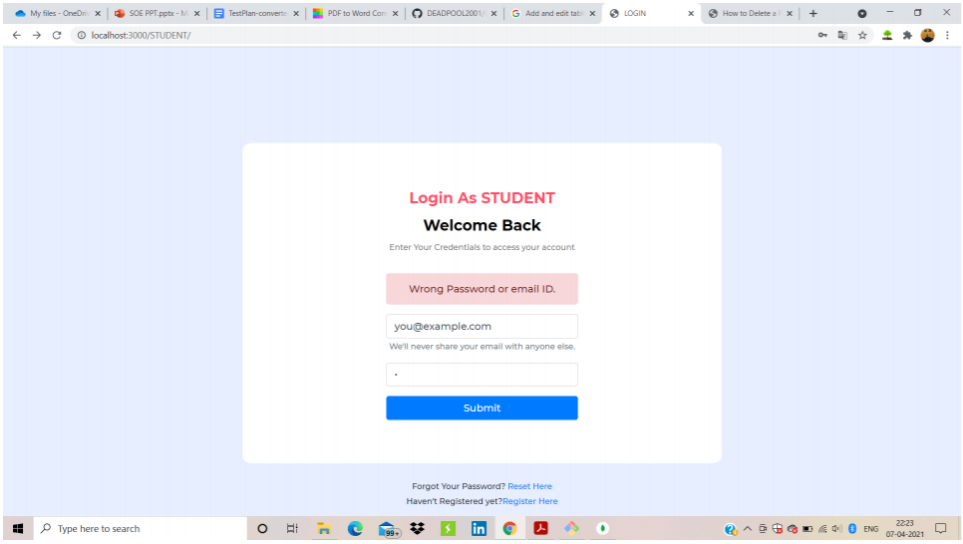
3) Login Page:

When we click on login in our welcome page, it shows us three options, to login as student, teacher or admin.
Let's take a look at each page.

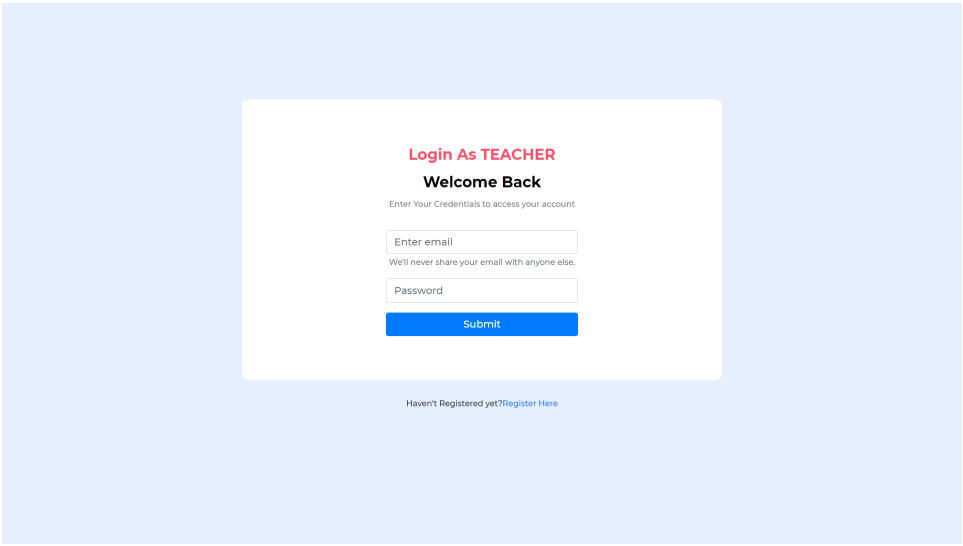
Student Login Page:



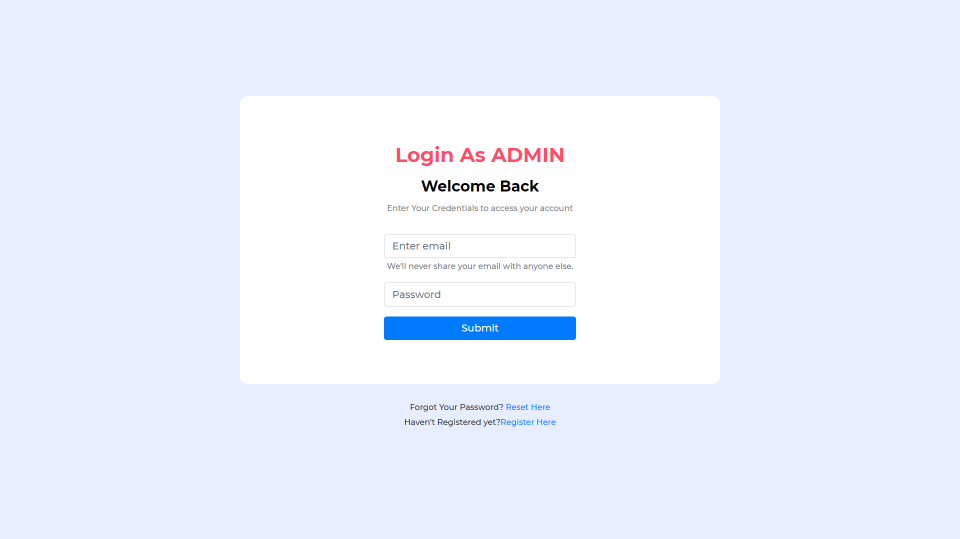
In case of incorrect email or password it informs us about it. (It does the same thing in other login tab)



Teacher Login Page:

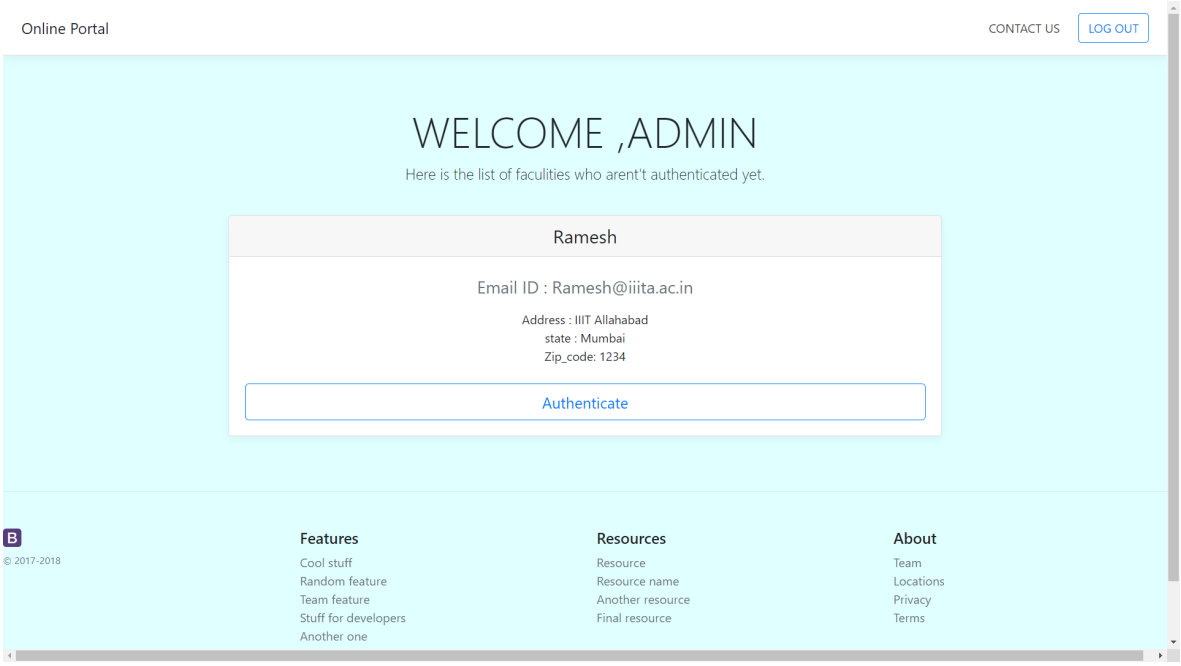


Admin Login Page:



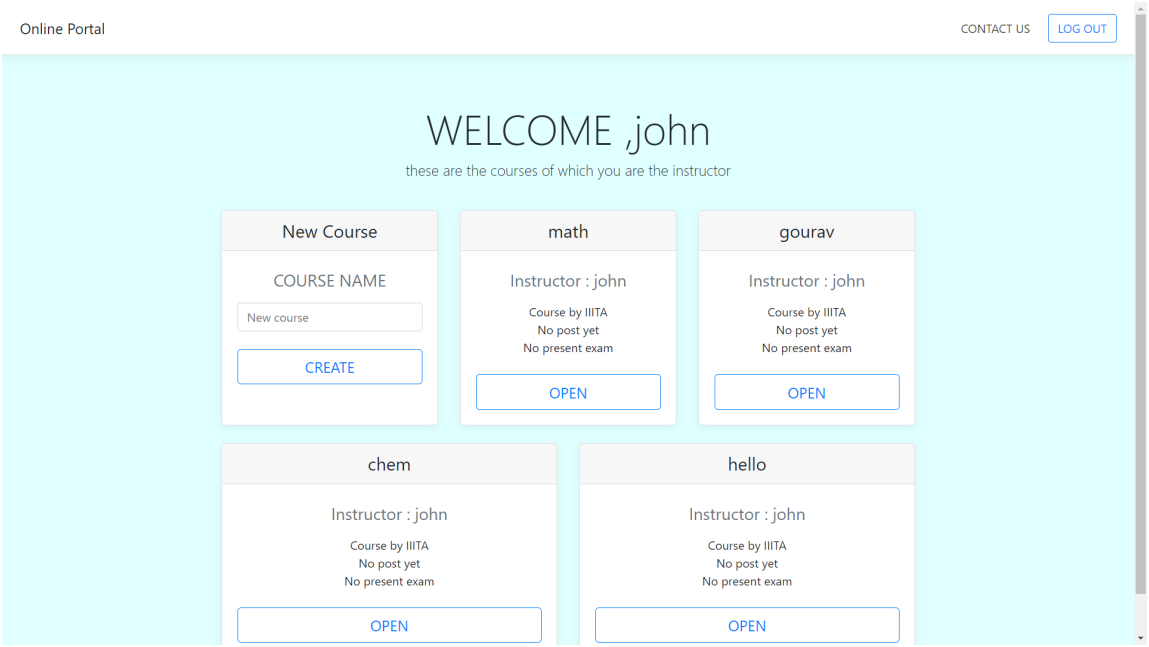
4) **Admin Authentication**

When Someone register himself/herself as a teacher, then admin must authenticate it, whether he/she actually is a teacher or not, let's look at that page:



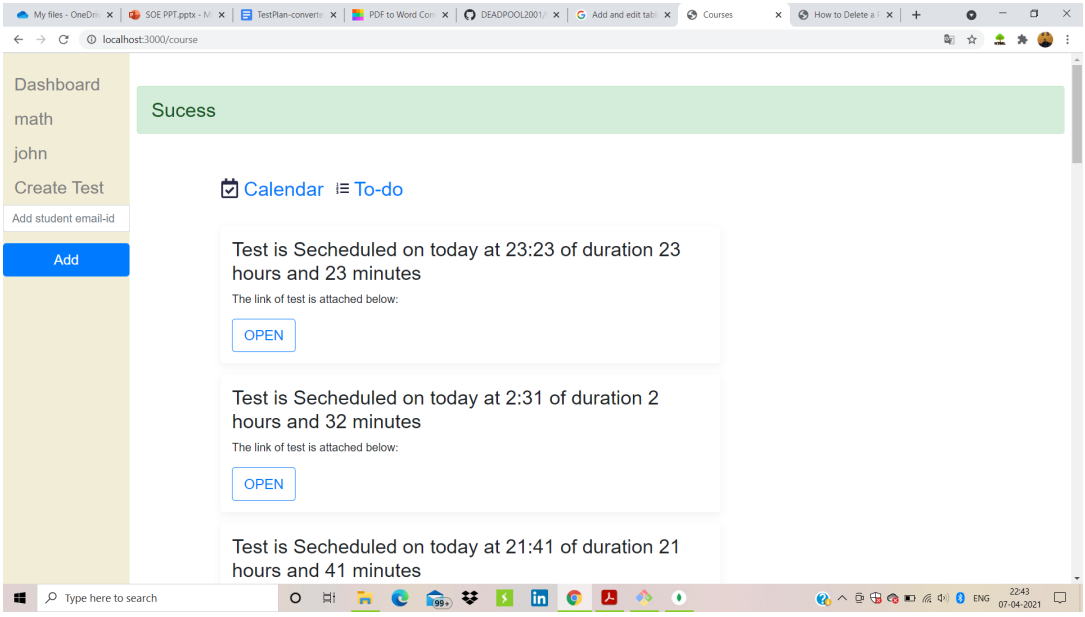
5) Creating Course

After successfully logging in as a Teacher can create a new course whenever they want.



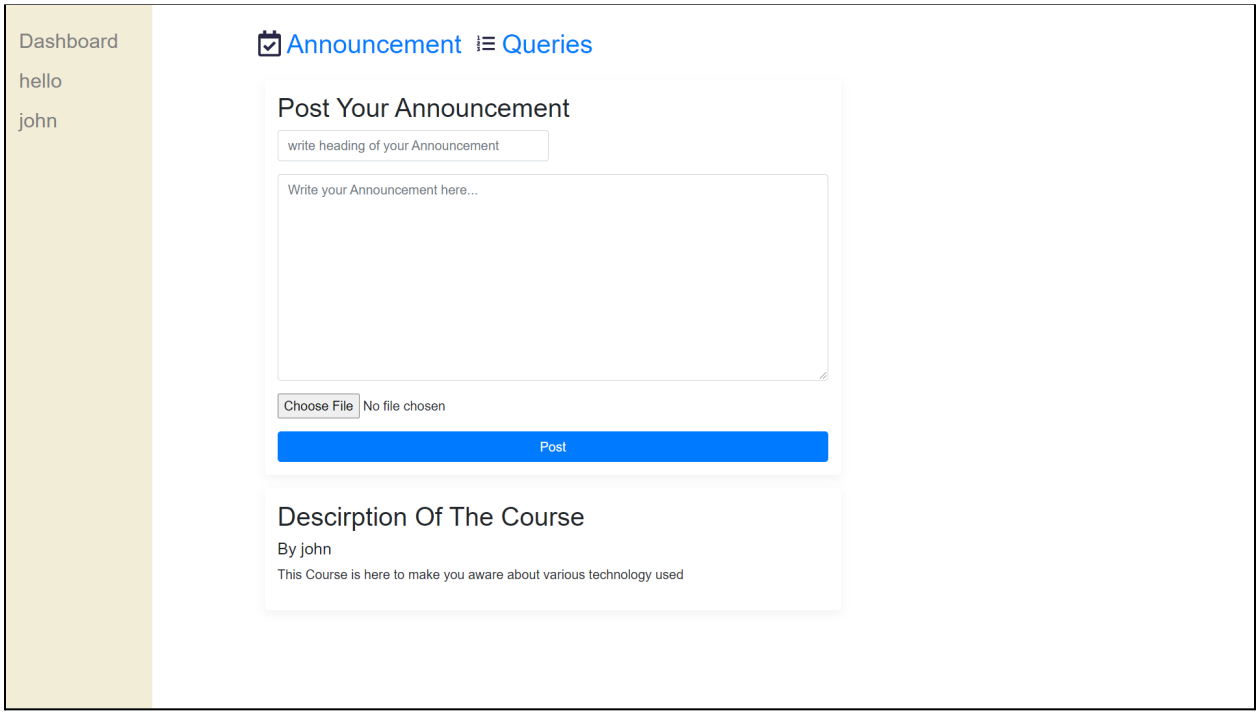
6) Adding student to course

a teacher can add students using their email ID , if it is present in the student database that particular student gets added to the course.



7) Adding Announcement in the course

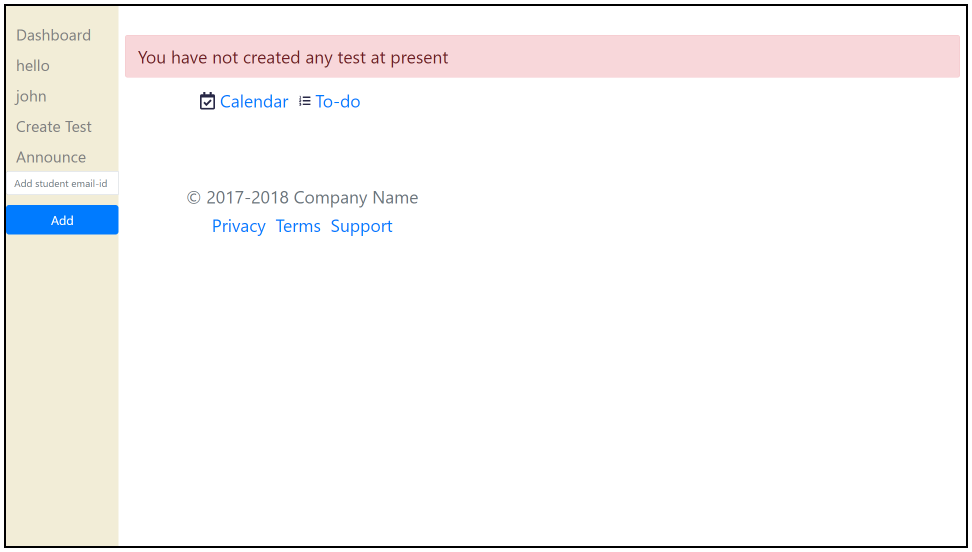
Teachers can post announcements regarding the course like description, or any other announcement like about the google meet link and when there will be class.



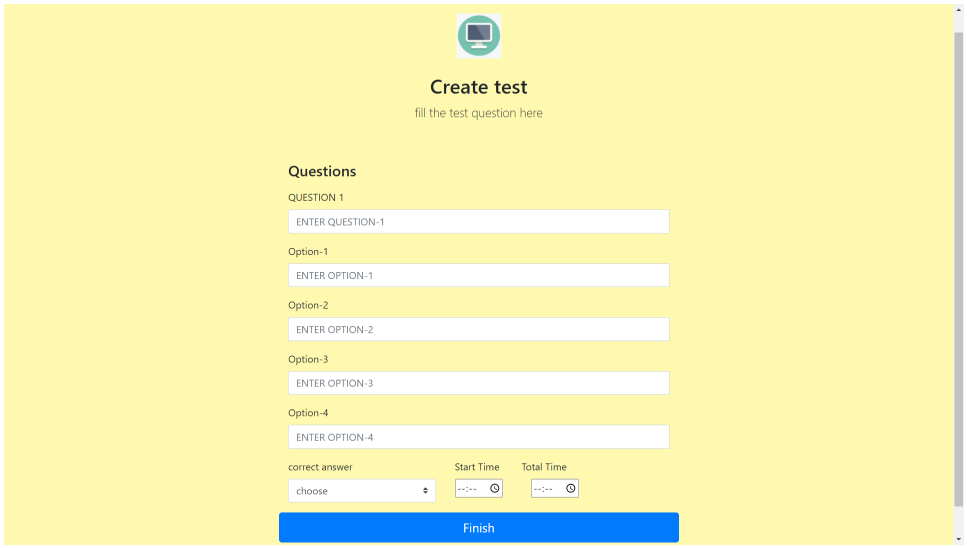
8) Creating Test

Teacher can also create a test whenever they want:

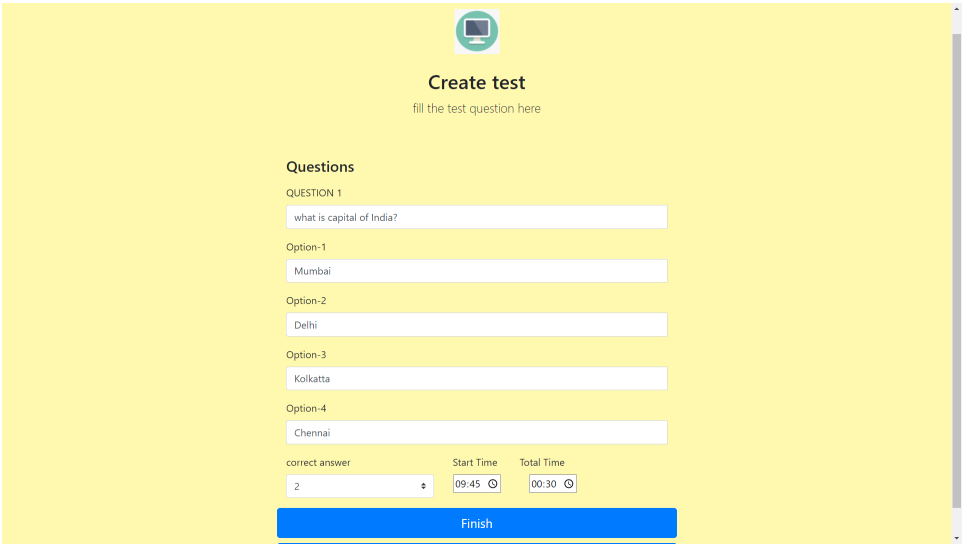
If there are no tests, it will display : You have not created any test at present.
To Create a New Test, we can click on Create a new test at the left side of the screen.



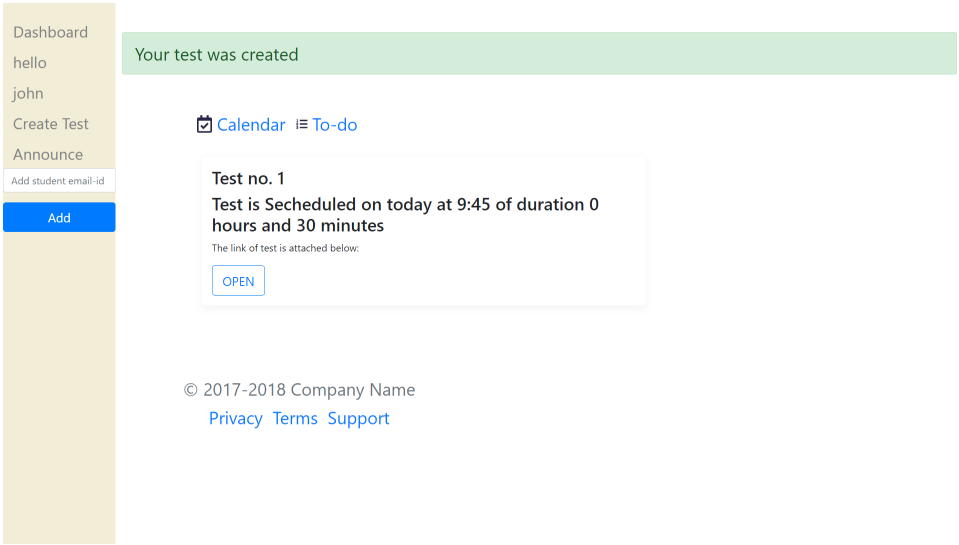
Test Creation Page:



We can enter the details of the questions listed above, let's look at an example

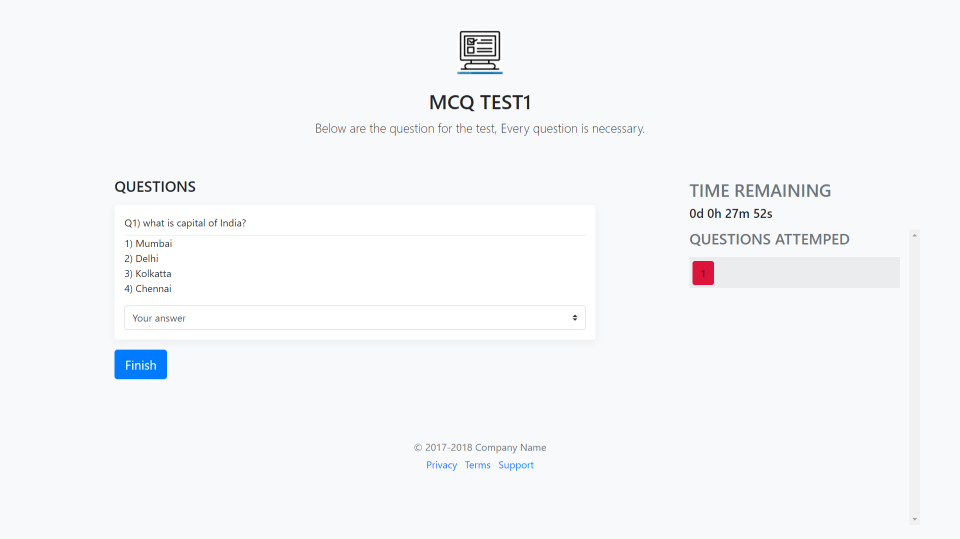


We can also add new questions to it, finally after creating the test, our screen would look like this:

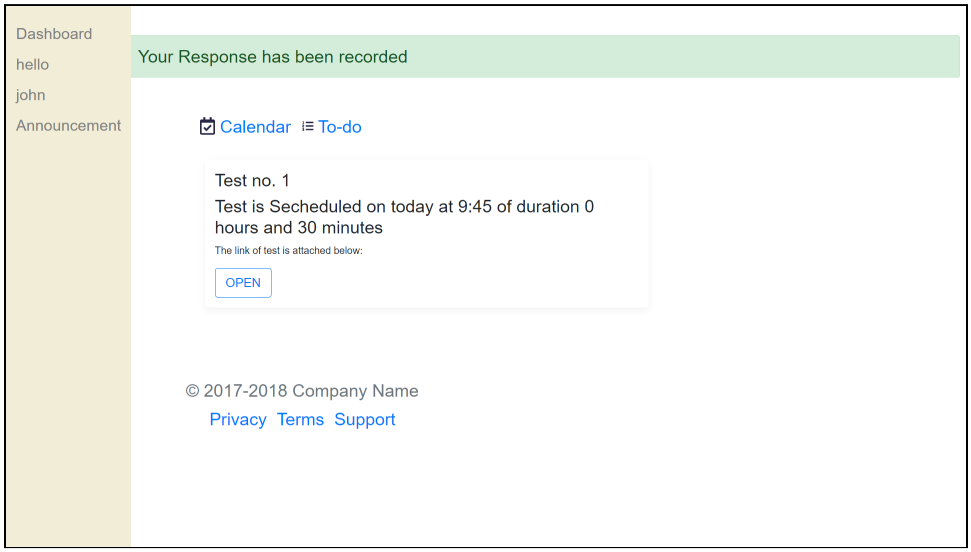


9) Attempting test

While attending the test, student's screen will look like:

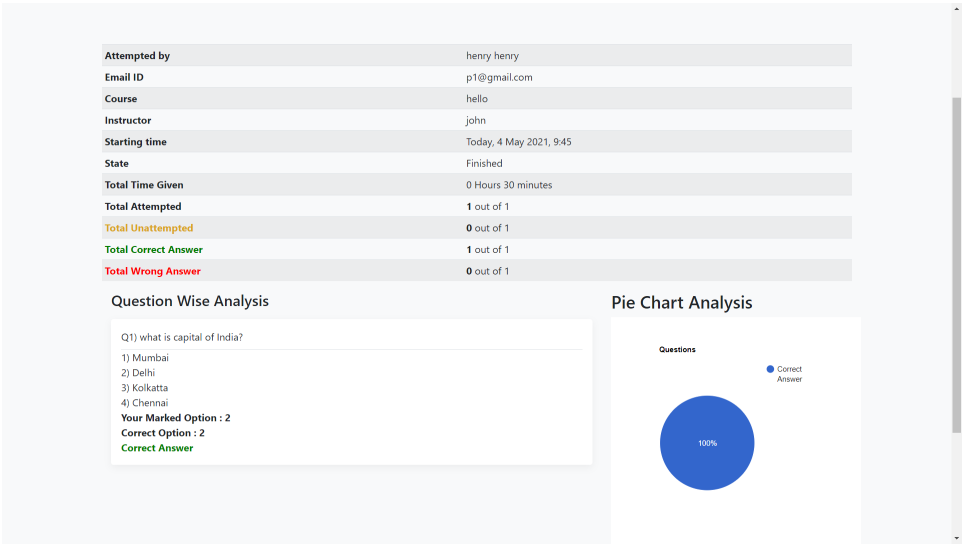


When a student finally submits the test (or opens a new tab), the response gets recorded.



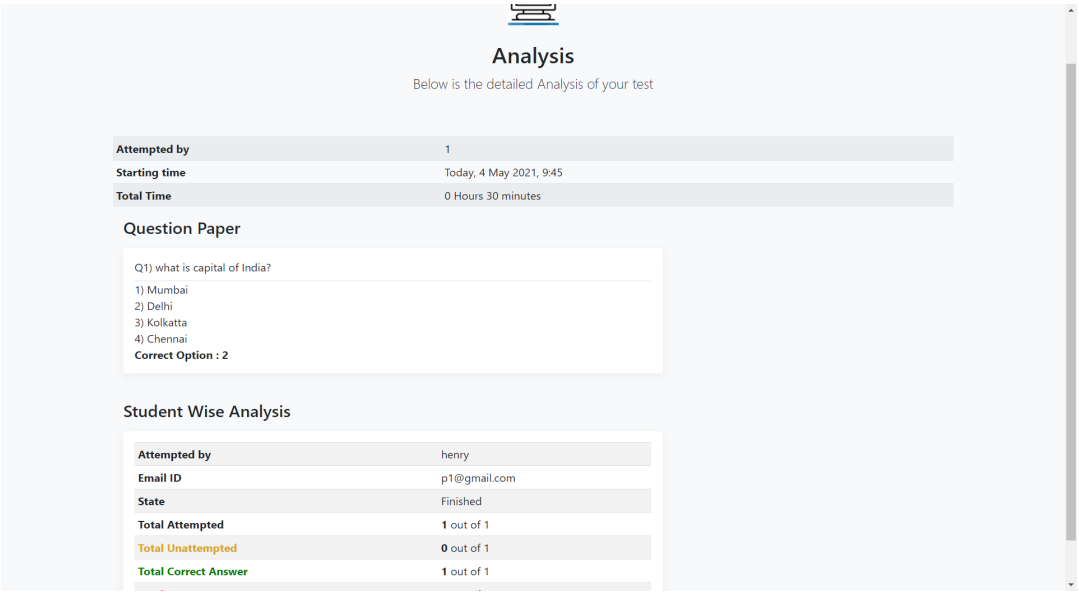
10) Result Analysis for student

Students can also see the analysis of the test ,after submitting the test they can see how many questions have been answered correctly and incorrectly and how many were unanswered , with pie chart analysis and also an editorial which shows which options were the correct answer.



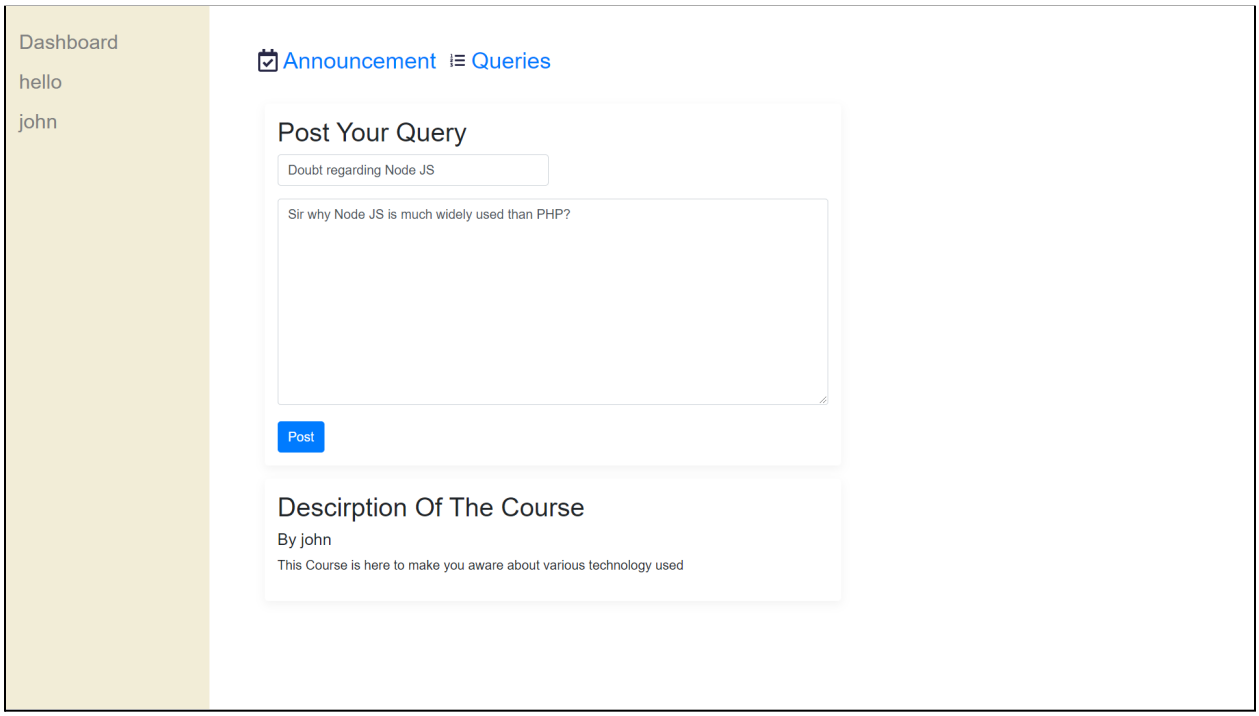
11) Result Analysis for teacher

Teacher can see how many student attempted their paper and how many marks they have got checking done in the backend he don't have to check manually



12) **Asking Queries in the course**

Students can ask various questions about the course ,which he can post in our query system.



---------- THE END ----------