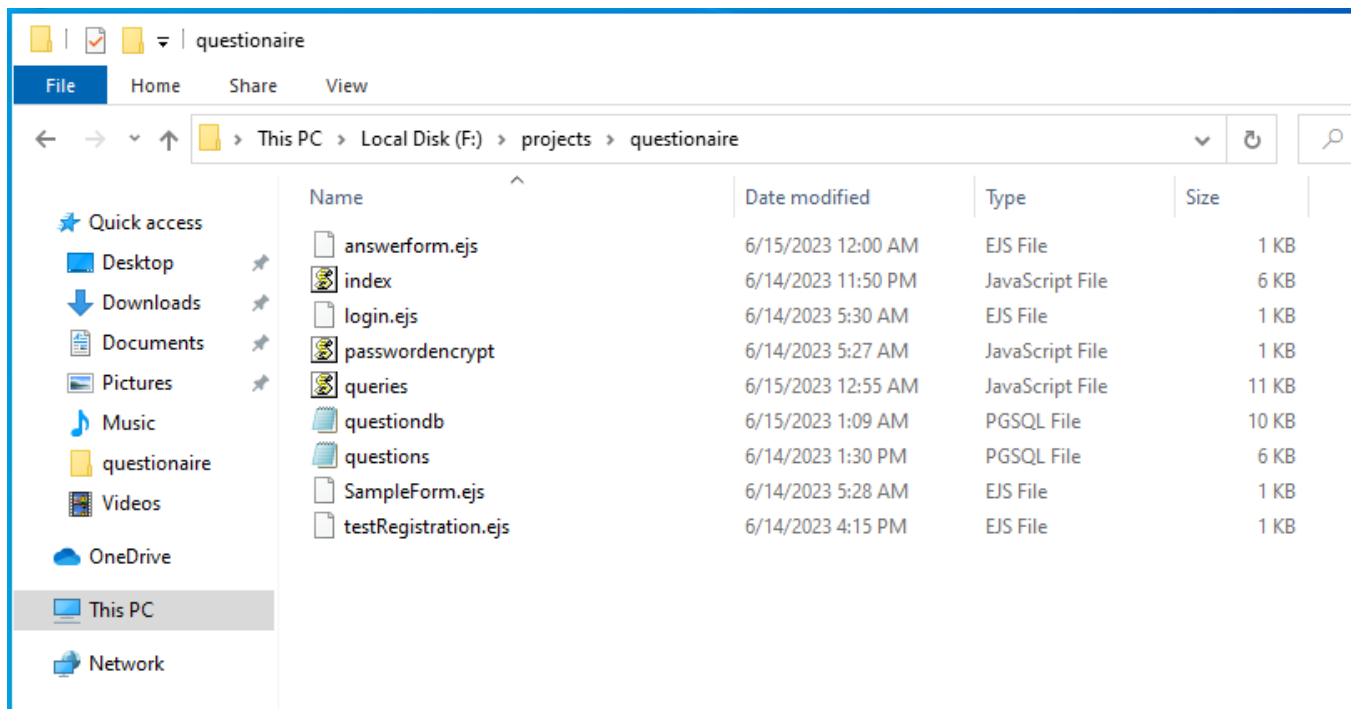
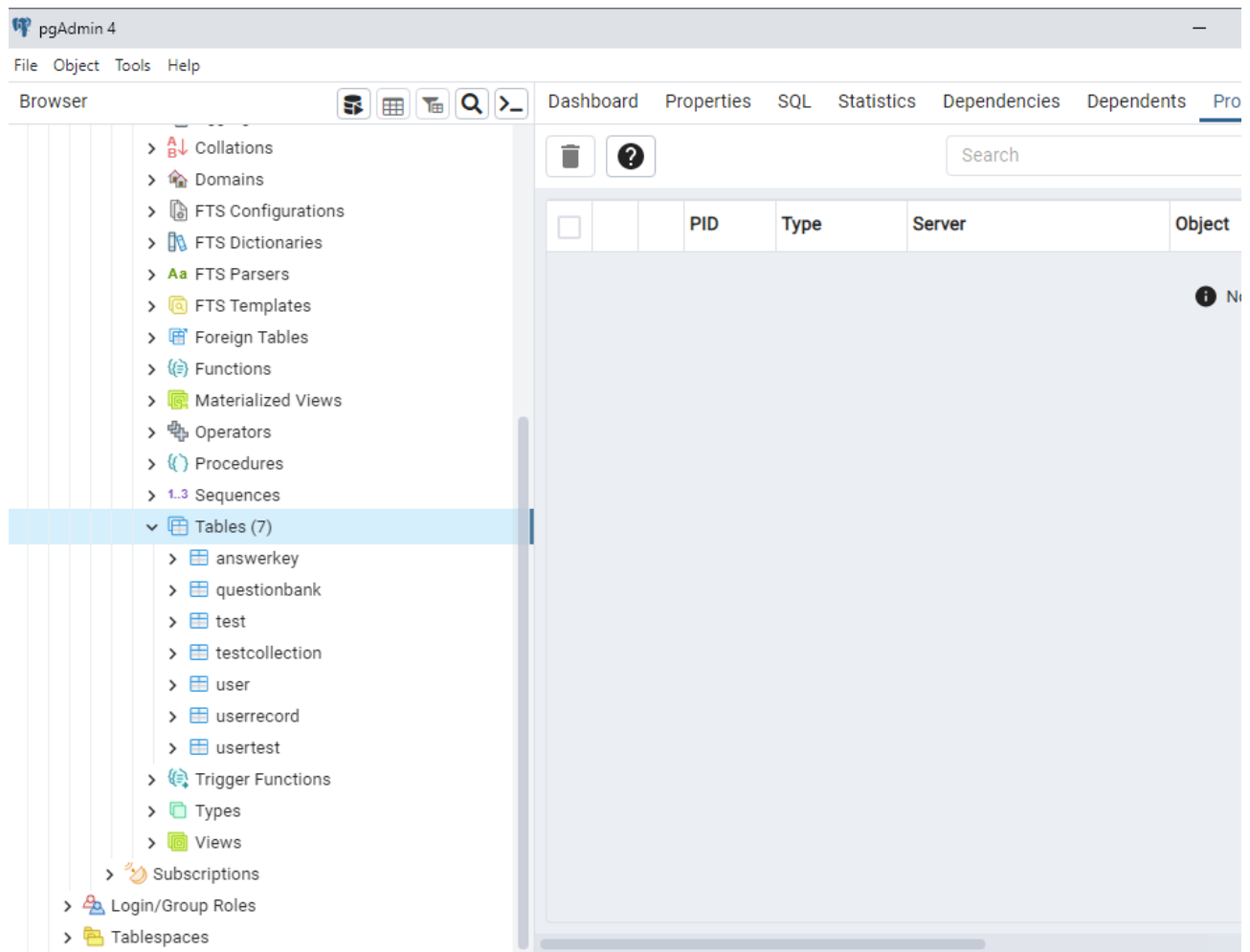


Below is the snapshot of the project code.



Below is the set of tables designed in postgres. The schema can be referred to in the questiondb.pgsql file along with initial data used during testing.



Observe that the user password is stored in an encrypted format in the DB. Bcrypt hash been used to mangle the password.

| | username [PK] character varying (100) | email character varying (100) | mobilen bigint | password character varying (200) |
|----|--|----------------------------------|-------------------|---|
| 11 | Student41 KIIT | student41@gmail.com | 1234554321 | [null] |
| 12 | Student42 KIIT | student42@gmail.com | 1234554321 | [null] |
| 13 | Student43 KIIT | student43@gmail.com | 1234554321 | \$2a\$10\$sekKaJo.mUEupZdveMzQWv.xxw0o6AVtc |
| 14 | Student44 KIIT | student44@gmail.com | 1234554321 | \$2a\$10\$qrD13mDiMHfyPtIAye8/F.NohlrJ4fESqpi |
| 15 | Student45 KIIT | student45@gmail.com | 1234554321 | \$2a\$10\$JkBPcjJJW.byet98aZm7ux9j0xrWvpx4z |

Below are the sequence of screen shots for the node js startup and rest api invocation.

```
C:\> Administrator: Command Prompt - node index

Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

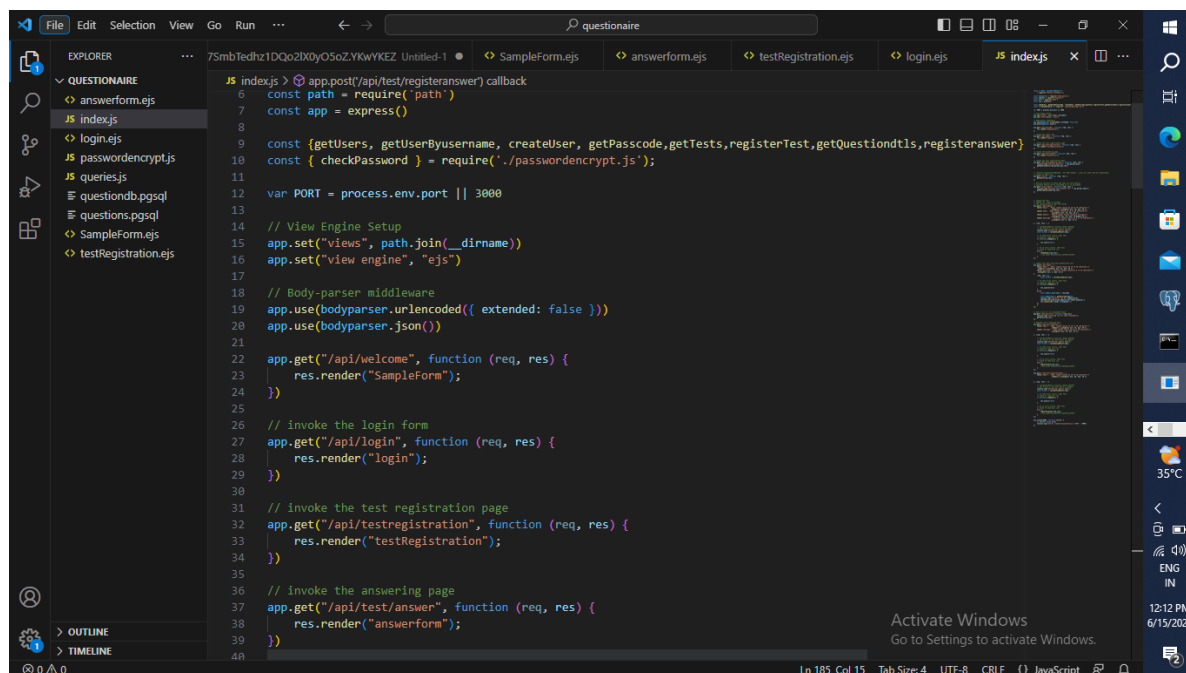
C:\WINDOWS\system32>f:

F:\>cd projects

F:\projects>cd questionnaire

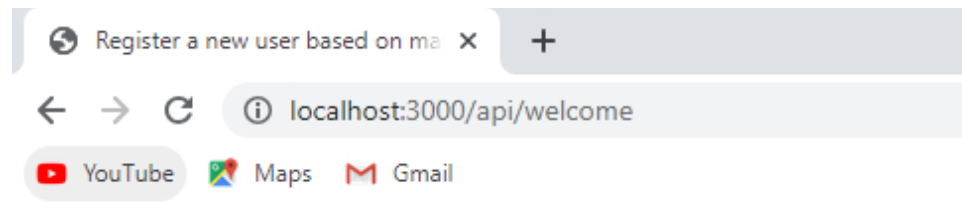
F:\projects\questionnaire>node index
Server created Successfully on PORT 3000
```

Below is the repository of end points exposed using node express.



```
File Edit Selection View Go Run ... questionnaire
EXPLORER
QUESTIONNAIRE
  answerform.ejs
  index.js
  login.ejs
  passwordencrypt.js
  queries.js
  questiondb.pgsql
  questions.pgsql
  SampleForm.ejs
  testRegistration.ejs
7SmbTcdhz1DQo2X0yO5oZ.YKwYKEZ. Untitled-1 • SampleForm.ejs answerform.ejs testRegistration.ejs login.ejs JS index.js x
JS index.js
6 const path = require('path')
7 const app = express()
8
9 const { getUsers, getUserByUsername, createUser, getPasscode, getTests, registerTest, getQuestionDtls, registerAnswer }
10 const { checkPassword } = require('./passwordencrypt.js');
11
12 var PORT = process.env.port || 3000
13
14 // View Engine Setup
15 app.set('views', path.join(__dirname))
16 app.set('view engine', 'ejs')
17
18 // Body-parser middleware
19 app.use(bodyParser.urlencoded({ extended: false }))
20 app.use(bodyParser.json())
21
22 app.get('/api/welcome', function (req, res) {
23   res.render('SampleForm');
24 })
25
26 // invoke the login form
27 app.get('/api/login', function (req, res) {
28   res.render('login');
29 })
30
31 // invoke the test registration page
32 app.get('/api/testregistration', function (req, res) {
33   res.render('testRegistration');
34 })
35
36 // invoke the answering page
37 app.get('/api/test/answer', function (req, res) {
38   res.render('answerform');
39 })
40
Ln 185, Col 15 Tab Size: 4 UTF-8 CRLF () JavaScript
```

When the service is started the welcome page allows for display of a user signup form.



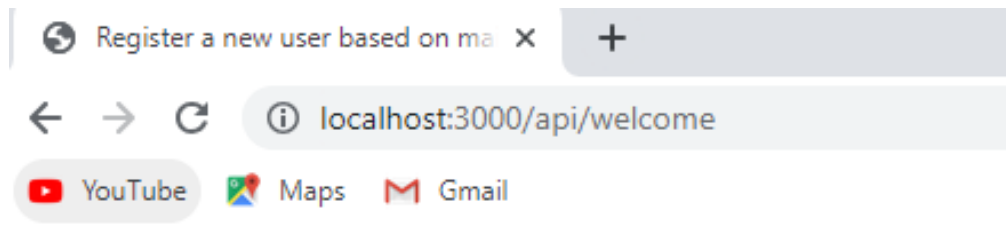
User Signup Form

Enter your Email :

Enter your Name :

Enter your Number :

Enter your Password :



User Signup Form

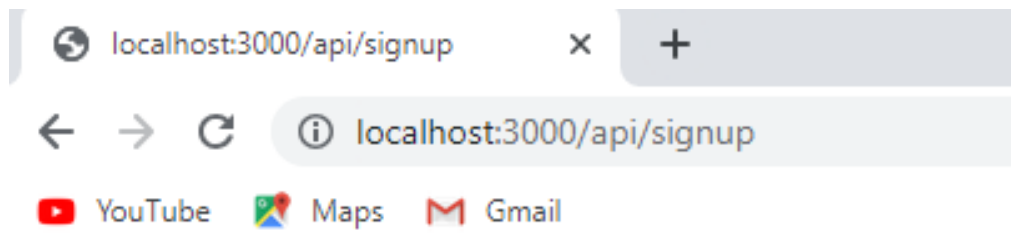
Enter your Email :

Enter your Name :

Enter your Number :

Enter your Password :

On submission the rest end point api/signup is called and response returned as below.



```
{success: true,message: "Signed up successfully"}
```

DB entry made as below:

Data Output
Messages
Notifications

| | username [PK] character varying (100) | email character varying (100) | mobileno bigint | password character varying (200) |
|---|--|----------------------------------|--------------------|--|
| 1 | atulagrax | atulagrawal1901@gmail.com | 7008251350 | demo#01 |
| 2 | Student 117 Agrawal | student117@KIIT.CO.IN | 9437050215 | \$2a\$10\$B4rA/r53DHs6yX3wM42v.nr392RJCSol |

User Login

localhost:3000/api/login

YouTube
Maps
Gmail

User Login Form

Enter your Email :

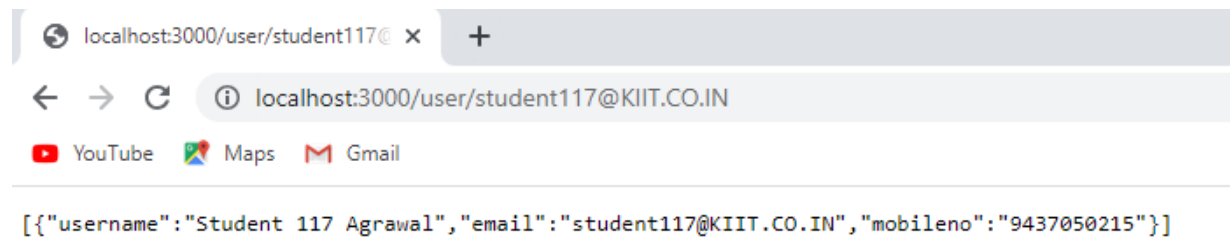
Enter your Password :

Login is currently failing because there seems to be an issue with password retrieval and validation using the hash. I will solve this EOD.

```
F:\projects\questionnaire>node index
Server created Successfully on PORT 3000
checking for email:student117@KIIT.CO.IN
Password from db undefined
F:\projects\questionnaire\passwordencrypt.js:12
    res.json({success: false, message: 'passwords do not match'});
    ^
TypeError: Cannot read properties of undefined (reading 'json')
    at F:\projects\questionnaire\passwordencrypt.js:12:11
    at process.processImmediate (node:internal/timers:476:21)

Node.js v18.16.0
F:\projects\questionnaire>
```

The following is the display of user details by providing the email id. API is /user/:mailid

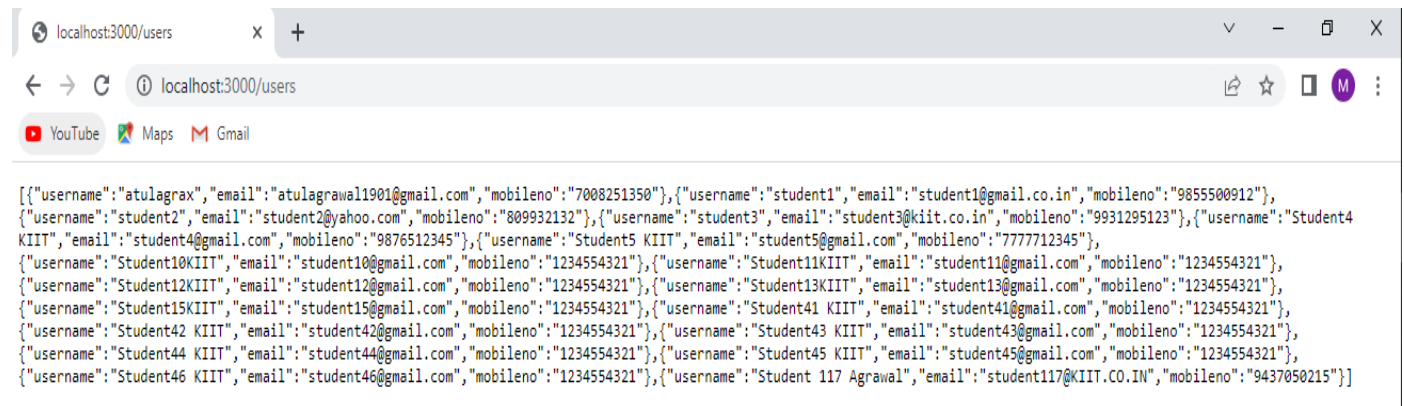


A screenshot of a web browser window. The address bar shows 'localhost:3000/user/student117@KIIT.CO.IN'. Below the address bar, there are icons for YouTube, Maps, and Gmail. The main content area displays a JSON array with one object: [{"username": "Student 117 Agrawal", "email": "student117@KIIT.CO.IN", "mobilen": "9437050215"}].

```
[{"username": "Student 117 Agrawal", "email": "student117@KIIT.CO.IN", "mobilen": "9437050215"}]
```

I have not yet implemented the user details modification api. Will soon implement it.

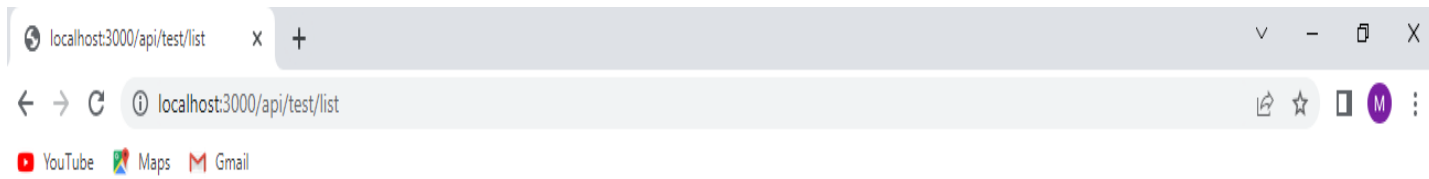
The below API is exposed for admin purposes (I.e to enable an admin to provide test access to users if tests are to be access controlled) “/users”



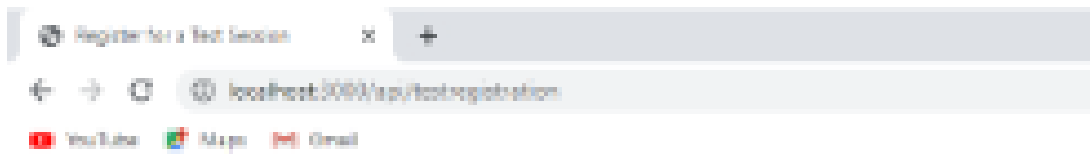
A screenshot of a web browser window. The address bar shows 'localhost:3000/users'. Below the address bar, there are icons for YouTube, Maps, and Gmail. The main content area displays a large JSON array containing 17 user objects, each with fields for username, email, and mobilen.

```
[{"username": "atulagrax", "email": "atulagraval1901@gmail.com", "mobilen": "7008251350"}, {"username": "student1", "email": "student1@gmail.co.in", "mobilen": "9855500912"}, {"username": "student2", "email": "student2@yahoo.com", "mobilen": "809932132"}, {"username": "student3", "email": "student3@kiit.co.in", "mobilen": "9931295123"}, {"username": "Student4 KIIT", "email": "student4@gmail.com", "mobilen": "9876512345"}, {"username": "Student5 KIIT", "email": "student5@gmail.com", "mobilen": "7777712345"}, {"username": "Student10KIIT", "email": "student10@gmail.com", "mobilen": "1234554321"}, {"username": "Student11KIIT", "email": "student11@gmail.com", "mobilen": "1234554321"}, {"username": "Student12KIIT", "email": "student12@gmail.com", "mobilen": "1234554321"}, {"username": "Student13KIIT", "email": "student13@gmail.com", "mobilen": "1234554321"}, {"username": "Student15KIIT", "email": "student15@gmail.com", "mobilen": "1234554321"}, {"username": "Student41 KIIT", "email": "student41@gmail.com", "mobilen": "1234554321"}, {"username": "Student42 KIIT", "email": "student42@gmail.com", "mobilen": "1234554321"}, {"username": "Student43 KIIT", "email": "student43@gmail.com", "mobilen": "1234554321"}, {"username": "Student44 KIIT", "email": "student44@gmail.com", "mobilen": "1234554321"}, {"username": "Student45 KIIT", "email": "student45@gmail.com", "mobilen": "1234554321"}, {"username": "Student46 KIIT", "email": "student46@gmail.com", "mobilen": "1234554321"}, {"username": "Student 117 Agrawal", "email": "student117@KIIT.CO.IN", "mobilen": "9437050215"}]
```

A user can see all tests which are available using the api “/api/test/list”



A user can invoke the registration page `/api/testregistration` to input test registration intent.



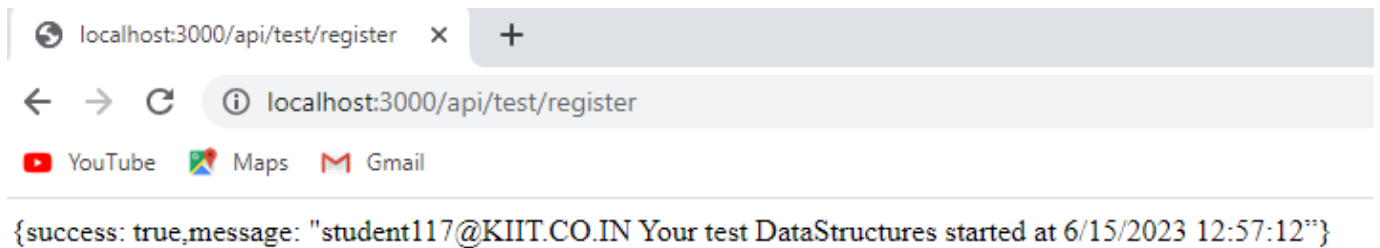
Test Selection

Enter your Email:

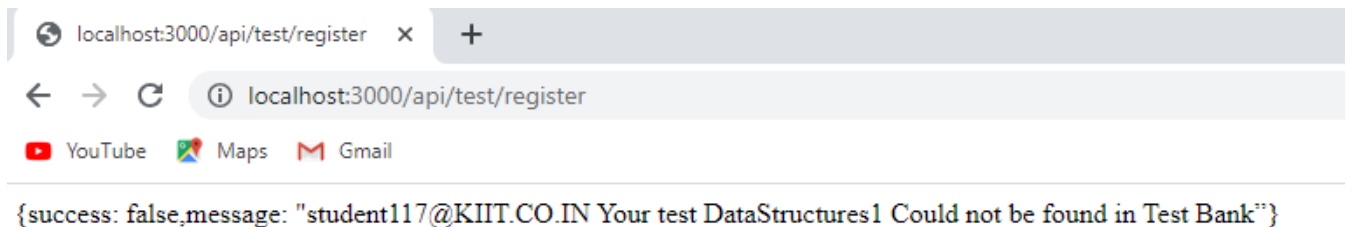
Type in the test name to attempt:

- DataStructures
- Language-Java
- Language-Java
- DataStructures

Below is a successful test registration



Below is an incorrect test registration submission (user submitted for DataStructures1 instead of Data Structures)

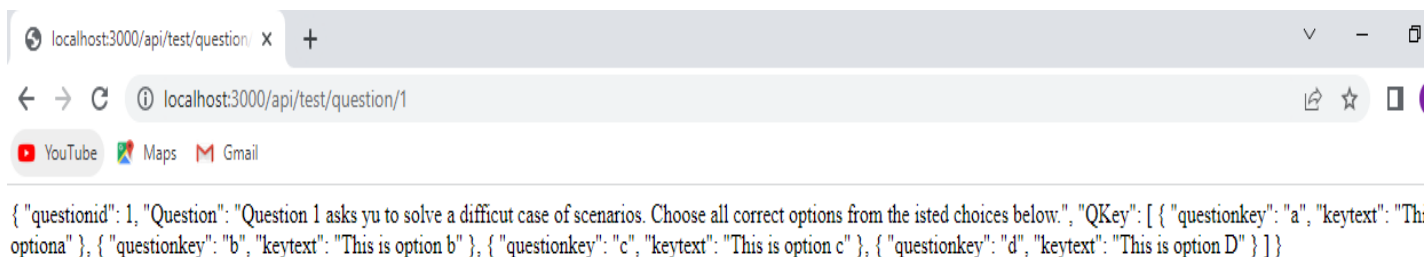


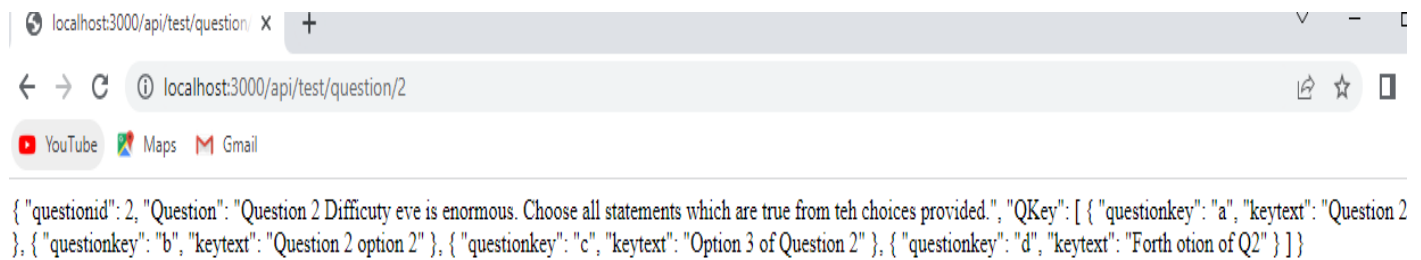
User will be allowed to make multiple attempts. Database will record timestamp and attempt sequence number for each test registration.

User can start a test and submit the request for the question number for a specific test. [This will actually be handled by front end code which will load the question ids corresponding to a test and present to the user by invoking the rest api for each question belonging to the test.]

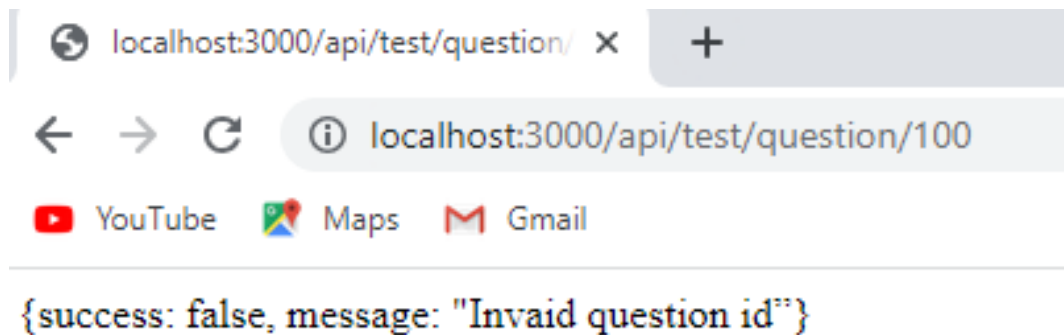
At this time tests and questions and test-question mapping is done directly into database. Later some admin screens and apis will be provided to create test bank and exam schedules.

The api `/api/test/question/:qno` (qno can be question id such as 1, 2, 3 etc.) It is assumed question ids are unique across the question bank. They are delinked from test collections so that any test can collect questions as they wish. The question ids will have a sequence running into few hundred thousand entries. The list of questions mapped to a test is provided in a table called testcollection. This mapping setup will be done by admin.





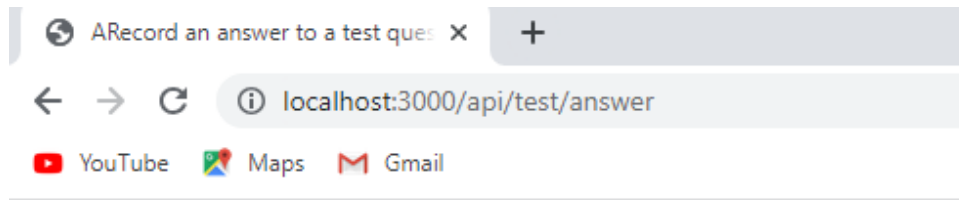
Invalid question ids cannot be submitted by the user or the frontend.



Each question has the question text and the accompanying Multiple choices.

The user or front end app can submit an answer to the portal for an ongoing test by submitting the choices for a question. (The question options are already specified as in the previous step)

The page `/api/test/answer` is created for testing the answer submission.



Answer to a question

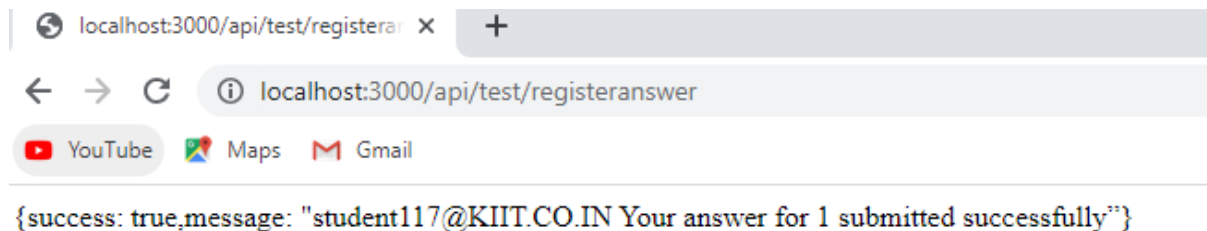
Enter your Email :

Enter the testid :

Enter the questionid :

Enter the selections :

All choices are sequenced with a comma separation. In the backend the comma is removed and the response is recorded as the option sequence chosen by the user.



Below table shows the responses for each candidate. The column locked if set to true will not allow for answer modification. Later a facility will be implemented to make this false initially on answer submission so that until the test session ends, modification will be allowed.

| | username [PK] character varying (50) | testid [PK] bigint | questionid [PK] bigint | date date | locked boolean | keysequence character varying |
|----|---|-----------------------|---------------------------|--------------|-------------------|----------------------------------|
| 1 | atulagrawal1901@gmail.com | 1 | 1 | 2023-06-15 | true | abc |
| 2 | atulagrawal1901@gmail.com | 1 | 2 | 2023-06-15 | true | a |
| 3 | student10@gmail.com | 1 | 1 | 2023-06-15 | true | a |
| 4 | student10@gmail.com | 1 | 2 | 2023-06-15 | true | a |
| 5 | student10@gmail.com | 2 | 1 | 2023-06-15 | true | a |
| 6 | student10@gmail.com | 3 | 1 | 2023-06-15 | true | a |
| 7 | student10@gmail.com | 4 | 1 | 2023-06-15 | true | a |
| 8 | student10@gmail.com | 5 | 1 | 2023-06-15 | true | bc |
| 9 | student117@KIIT.CO.IN | 1 | 1 | 2023-06-15 | true | abc |
| 10 | student4@gmail.com | 1 | 1 | 2023-06-15 | true | a |
| 11 | student4@gmail.com | 1 | 2 | 2023-06-15 | true | a |

At this moment the evaluation of test score is not implemented. This will be implemented by selecting all answers from the userrecord table (as shown above) and matching each questionid against the answer keys from the question table as below. All question keys which are marked valid as below will be sequenced for the question and compared to the answer sequence submitted by the user. If that matches, the relevant marks will be added to the score. (Example for q1 ac will be the right answer)

| | questionid [PK] bigint | questionkey [PK] "char" (1) | keytext character varying | invalid boolean |
|---|---------------------------|--------------------------------|------------------------------|--------------------|
| 1 | 1 | a | This is optiona | true |
| 2 | 1 | b | This is option b | false |
| 3 | 1 | c | This is option c | false |
| 4 | 1 | d | This is option D | true |
| 5 | 2 | a | Question 2 Otion 1 | false |
| 6 | 2 | b | Question 2 option 2 | true |
| 7 | 2 | c | Option 3 of Question 2 | false |
| 8 | 2 | d | Forth otion of Q2 | false |

The number of questions and distribution of marks for each question (equally) is determined by the test table. (total marks / no of questions will give marks for each question).

When user submits closure of the test the test end time will be updated and score evaluation algorithm will kick in and below table populated with the total marks and percentage score. [This end point is not yet created. This will be closed asap]

The entire code of this project will be in open-source domain as I develop further on this.

| | username [PK] character varying (100) | testid [PK] bigint | attemptno [PK] bigint | attemptedon date | score bigint | percent numeric | starttime time with time zone | endtime time with time zone |
|----|--|-----------------------|--------------------------|---------------------|-----------------|--------------------|----------------------------------|--------------------------------|
| 1 | atulagrawal1901@gmail.com | 4 | 1 | 2023-05-03 | 0 | 0 | 19:21:01-07:00 | [null] |
| 2 | atulagrawal1901@gmail.com | 4 | 3 | 2023-05-03 | 0 | 0 | 19:27:30-07:00 | [null] |
| 3 | atulagrawal1901@gmail.com | 4 | 4 | 2023-05-03 | 0 | 0 | 19:55:37-07:00 | [null] |
| 4 | atulagrawal1901@gmail.com | 4 | 5 | 2023-05-03 | 0 | 0 | 19:59:18-07:00 | [null] |
| 5 | atulagrawal1901@gmail.com | 4 | 6 | 2023-06-14 | 0 | 0 | 20:01:05-07:00 | [null] |
| 6 | atulagrawal1901@gmail.com | 4 | 7 | 2023-06-14 | 0 | 0 | 20:04:45-07:00 | [null] |
| 7 | atulagrawal1901@gmail.com | 4 | 11 | 2023-05-03 | 0 | 0 | 19:26:01-07:00 | [null] |
| 8 | student117@KIIT.CO.IN | 4 | 1 | 2023-06-15 | 0 | 0 | 12:57:12-07:00 | [null] |
| 9 | student117@KIIT.CO.IN | 4 | 2 | 2023-06-15 | 0 | 0 | 12:59:38-07:00 | [null] |
| 10 | student4@gmail.com | 4 | 1 | 2023-06-14 | 0 | 0 | 20:05:57-07:00 | [null] |
| 11 | student43@gmail.com | 4 | 1 | 2023-06-14 | 0 | 0 | 20:18:50-07:00 | [null] |

| | testid [PK] bigint | testname [PK] character varying (100) | testdescription character varying (250) | questioncount bigint | totalmarks bigint |
|---|-----------------------|--|---|-------------------------|----------------------|
| 1 | 1 | SoftSkills-Presentation | To Test Presentation Skills of an Engineering Student | 20 | 20 |
| 2 | 2 | OSExpertise | This Test assesses the operating system appreciation by a com... | 15 | 30 |
| 3 | 3 | CISCOCertification | A CISCO engineers requires Edge devices skill sets to be succe... | 20 | 50 |
| 4 | 4 | DataStructures | Can a candidate solve data representation and analysis chalen... | 15 | 20 |
| 5 | 5 | LanguageSkills-Java | Java being a very popular programming language for modern I... | 25 | 25 |