

# CSE 408 Project Documentation

## Online Exam

*Prepared By:*

Mohammad Anas( 1505087)

Md. Masud Rana( 1505093)

Arghya Pratim Pal( 1505115)

*Prepared For:*

Md. Shariful Islam Bhuyan

Dr. A. B. M. Alim Al Islam

Dr. Anindya Iqbal

*Under Supervision of-*

Dr. Anindya Iqbal

# ***1) Vision Statement***

## ***a)Project Goals***

The main goal of this project is to create an environment for the users for the participation in the online exam & hence he gets an overall idea of his performance. Usually in the typical examinations, the problem statement is not standard & doesn't contain quality tutorial material from where the student(user) can be benefited. We have resolved this problem where the user can participate in the exams from any place he wants to. Besides, we have developed an interface where the kids can also be benefited & improve their geometrical knowledge.

## ***b)Accomplishments***

We completed all of the major tasks to finalize our project. These tasks include:

- 1)Back-end design
- 2)Database creation
- 3)Set exam module
- 4)User improvement/performance tracking module
- 5)Kids drawing practice module
- 6)Kids geometrical knowledge improvement module
- 7)Show questions module
- 8)Data insertion & manual testing

## ***c)Scope of the project***

We have a set exam module by which user can set an exam. The questions of the exam is set randomly & it is assured that 40% questions difficulty from the total number of questions is easy. Similarly 30% is medium , 20% is hard & 10% is advanced. After an exam the user can see the tutorial of the questions & his performance graph is also updated according to the score of the exam he just participated.

### ***d)Major milestones of the project***

According to CSE 408 Software Development course's outline, we created the milestones of our project shown in the following table:

| Week    | Task  |
|---------|---|
| Week 3  | Project update: Database & ERD implementation   |
| Week 4  | Project update: Basic front end design & implementation                                 |
| Week 5  | Project update: Set model test UI(Basic front end) implementation                       |
| Week 6  | Project update: Set model test UI(Count-down timer) implementation                      |
| Week 7  | Project update: Set model test UI(Determining the Questions of the exam) implementation |
| Week 8  | Project update: Script evaluation system implementation                                 |
| Week 9  | Project update: Admin panels work(Add Questions) implementation                         |
| Week 10 | Project update: Admin panels work(Adding new topic of a new subject) implementation     |
| Week 11 | Project update: Kids drawing subsystem implementation                                   |
| Week 12 | Project update: Kids geometrical knowledge analyzer implementation                      |
| Week 13 | Project update: Overall bug fixing & checking & adding some extra small features        |
| Week 14 | Final Evaluation  |

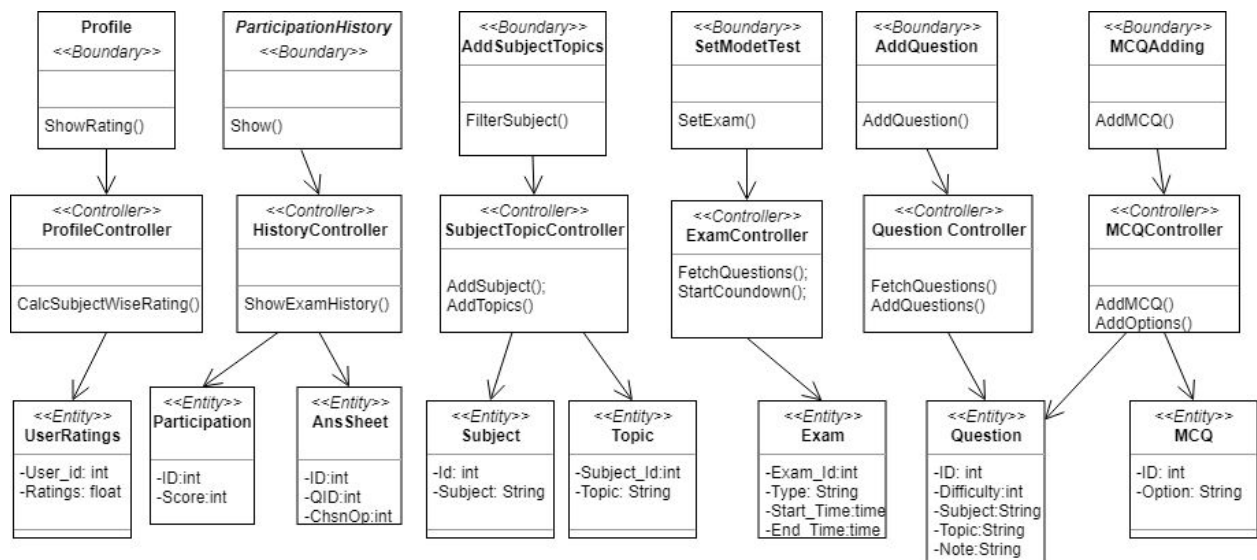
## ***e)Users of the project***

W3MTest is built for 2 types of users:

- 1) Admin
- 2) Normal users(Examinee)

There are three admins of this project. Their job is to add new questions to the database according to their difficulty,topic,subject etc.Another job of the admin is to add new topic to the existing subjects / new subject. The Normal users take part in the exams & they can see their improvements & lackings subject wise.

## ***2) Architecture & Design:***



### ***3)Functional description***

W3MTest provides following features for 3 types of users.

#### ***a)Admin***

An admin is the head of the project & controls everything externally. The responsibilities of the admin is as follows:

- i). Add new mcq
- ii). Add new Topic of the existing subject
- iii). Add new Subject

#### ***b)Examinee***

- i). Set an instance model test
- ii). Participate in the exam
- iii). Going through the wrong & correct answers

after the exam

- iv). Searching questions of any particular subject
- v). Watching the previous participations
- vi). Tracking of improvement & subject wise ratings

according to the result of the exam

An user(Examinee) firstly set an exam. The Questions of the exam is set instaneously & instantaneously. The examinee also has to define the subject & topic he is willing to participate. He has also the right of determining the number of questions. Timer is set for 20 seconds after that for warm up. We have allotted 45 seconds for every question to answer for the participant. So the total time of the exam is:  $(45 * \text{\#of questions})$  sec. A timer is set according to that. After the exam is over (May be time is up / the examinee finishes before the time), his score is evaluated. In the evaluation process, he is shown the questions along with the correct answers & his chosen options. An editorial is also provided after every question. Based on his current score, his subject wise rating (Calculated out of 100) is also updated. If he gets poor marks, his rating also decreases & vice versa.

The user(Examinee) has also the right to see the all the questions of any particular subject & its corresponding topics.

User can also see his profile. In his profile, he can see his all the exams he participated in. Every participation contains a link along with the date he participated in that exam , the subject & topic the exam belonged & the score of the user in that exam.If the user clicks the link,he can see his answer script & his chosen options in that exam.

### ***c)Kids***

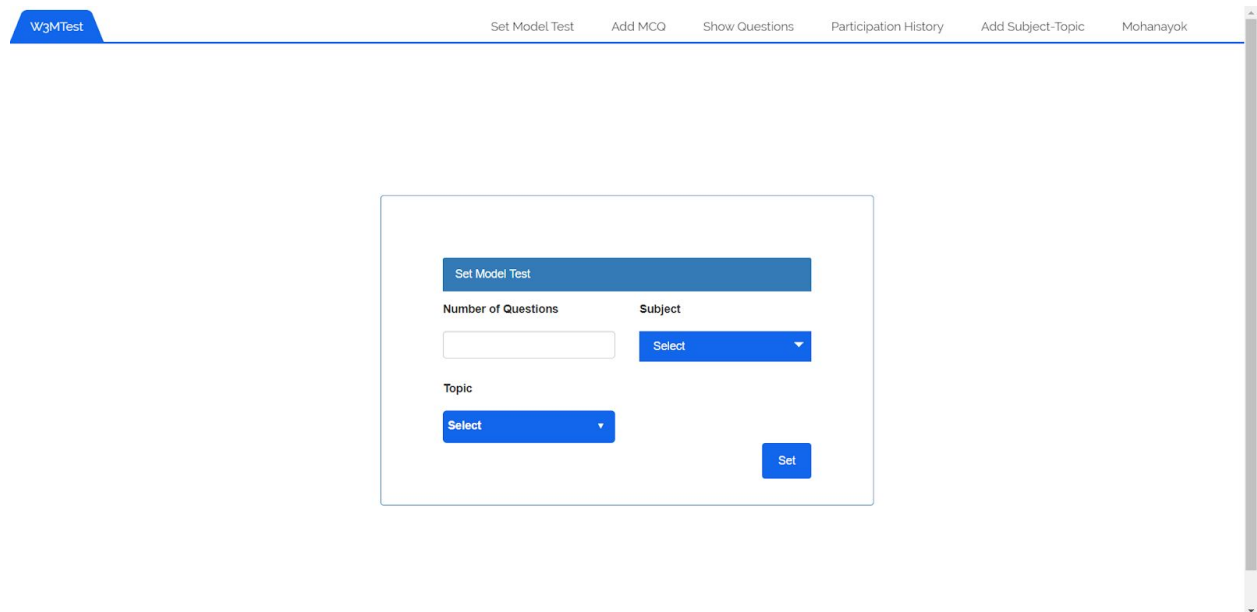
- i). Draw the image of any given figure.
- ii). Develop the geometrical concept of the kids by solving some quiz.

In the first thing , an image is shown to the kid , he tries to draw it by using the painting tool we generated in the website.On the other hand, in the second case , kid is given some shape (i.e. Triangle , Rectangle , Regular Hexagon). He is said to find  $\frac{1}{4}$  th of the triangle / rectangle / hexagon etc. He is judged according to his drawing & an accepted verdict is shown depending on the drawing of the user.

## 4) *User Guide:*

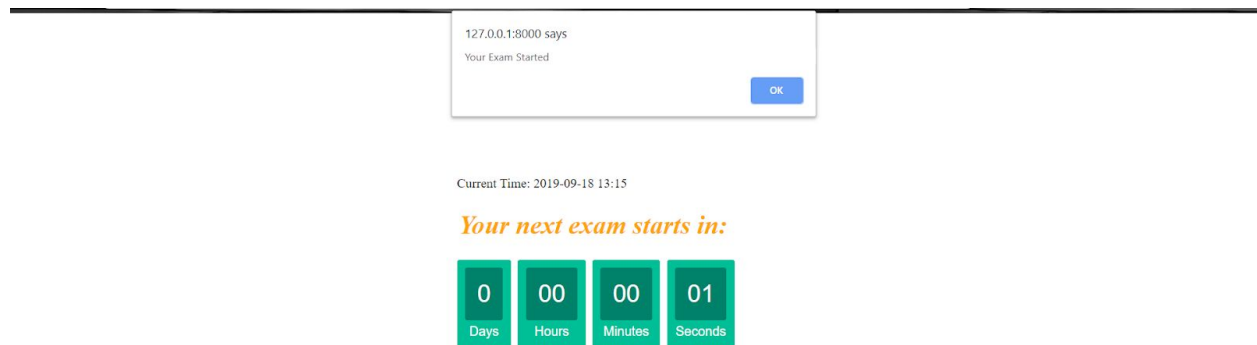
### *a) Set Model Test:*

After pressing Set Model Test button , he will be shown the following page:



The screenshot shows a web application interface for setting a model test. At the top, there is a navigation bar with a blue tab labeled 'W3MTest' and several links: 'Set Model Test', 'Add MCQ', 'Show Questions', 'Participation History', 'Add Subject-Topic', and 'Mohanayok'. The main content area is a light gray box containing a form. The form has a title 'Set Model Test' in a blue header. Below the title, there are three input fields: 'Number of Questions' (a text input), 'Subject' (a dropdown menu with a 'Select' button), and 'Topic' (a dropdown menu with a 'Select' button). A 'Set' button is located at the bottom right of the form.

After filling up the required credentials he is taken to the next page, where a timer is set up for 15 seconds:



After clicking Ok Button,he is taken to the questions page. The page looks like following:

The screenshot shows a questions page with a light blue background. At the top, a large timer displays "3m 22s". Below the timer, there are four questions, each with a text input field and a list of radio button options. Question 1: "Alice's father is a surgeon, he mostly does not have time to spend with his family. (Which word is a singular possessive noun?)" with options: surgeon, time, family, Alice's father, none. Question 2: "I could not recognize her voice on phone. (Which word is an abstract noun?)" with options: phone, could not, her, the, voice. Question 3: "They like swimming. Whenever they have a leisure time. (Which word is a verbal noun?)" with options: leisure, free, like, swimming, none. Question 4: "Australian government will bring together anti-terror laws. (Which word is a nominative noun?)" with no visible options.

3m 22s indicates the remaining time. The user needs to click the submit button.



☒ singer  
☐ Michael  
☐ was  
☐ great  
☐ none

8: Adam's and John's ideas were greatly different. (Which word is a compounded possessive noun?)

☒ Adam's and John's  
☐ ideas  
☐ different  
☐ greatly  
☐ none

9: She is mother, and a great teacher too. (Which word is a predicate noun?)

☒ great  
☐ too  
☐ mother  
☐ teacher  
☐ she

10: I checked all primary classrooms, no one was there. (Which word is a compound noun?)

☒ checked  
☐ classrooms  
☐ primary  
☐ there  
☐ none

submit

After clicking the submit button,he is shown a page containing his score,the short note after every question , his chosen option & the correct option.

W3MTest

Set Model Test Add MCQ Show Questions Participation History Add Subject-Topic Mohanayok

Your Score : 2

Question: Alice's father is a surgeon, he mostly does not have time to spend with his family. (Which word is a singular possessive noun?)

☒ surgeon  
☐ time  
☐ family  
☒ Alice's father  
☐ none

! d-Alice's father

Question: I could not recognize her voice on phone. (Which word is an abstract noun?)

☒ phone  
☐ could not  
☐ her  
☐ the

Here his score is shown 2 . The short note is shown after every question. His chosen option is shown in red mark, which is wrong. Green means correct answer.

### ***b)Add MCQ:***

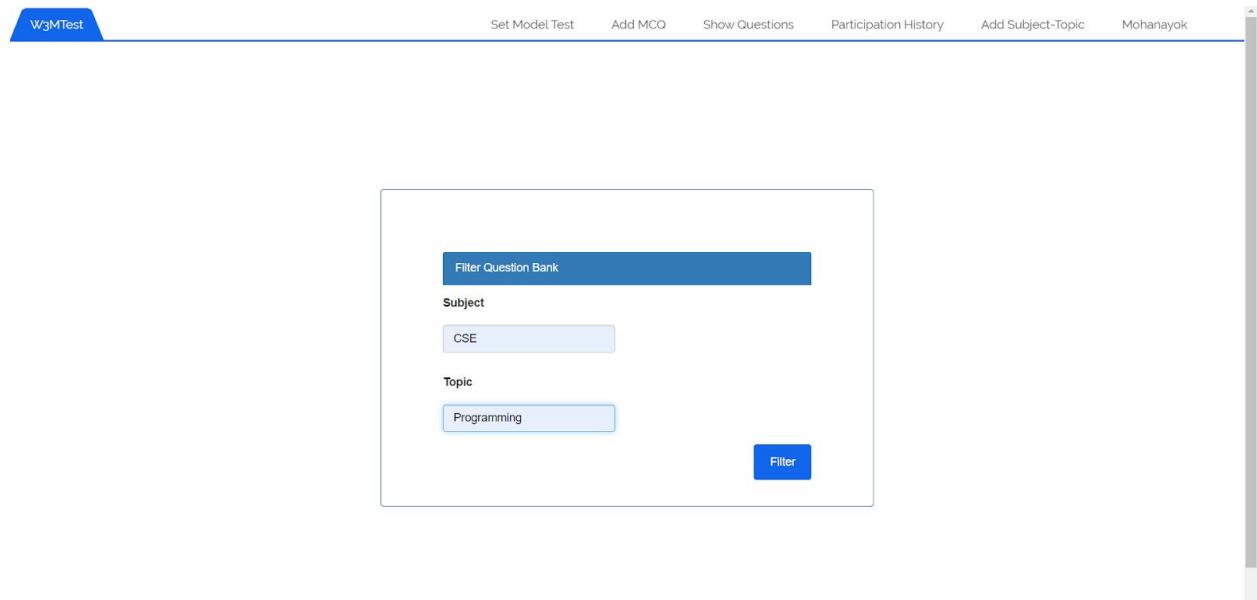
The screenshot shows the 'Add MCQ' form in the W3MTest application. The form is located in the center of the page, below the navigation bar. The navigation bar includes the 'W3MTest' logo and links for 'Set Model Test', 'Add MCQ', 'Show Questions', 'Participation History', 'Add Subject-Topic', and 'Mohanayok'. The form itself has a light gray background and contains several input fields: 'Subject' and 'Topic' (both empty), 'Question' (empty), 'Option A' (empty), 'Option B' (empty), 'Option C' (empty), 'Option D' (empty), 'Option E' (empty), and 'Short Note' (empty). Below these fields are three green buttons: 'Correct Option' (with a dropdown arrow), 'Difficulty' (with a dropdown arrow), and 'Add'.

Add MCQ feature looks like the following after adding the mcq:

The screenshot shows the 'Add MCQ' form in the W3MTest application after a question has been added. The form is located in the center of the page, below the navigation bar. The navigation bar includes the 'W3MTest' logo and links for 'Set Model Test', 'Add MCQ', 'Show Questions', 'Participation History', 'Add Subject-Topic', and 'Mohanayok'. The form itself has a light gray background and contains several input fields: 'Subject' (filled with 'CSE'), 'Topic' (filled with 'Programming'), 'Question' (filled with 'What is the output of the following code : \* x = 3 ; x = ++x; print(x) \*'), 'Option A' (filled with '2'), 'Option B' (filled with '3'), 'Option C' (filled with '4'), 'Option D' (filled with '5'), 'Option E' (filled with '6'), and 'Short Note' (filled with 'The correct answer is : 4'). Below these fields are three green buttons: 'Correct Option' (with a dropdown arrow), 'Difficulty' (with a dropdown arrow), and 'Add'.

### ***c)Filter Question Bank:***

According to the users demand of subject & the topic,he is shown all the questions of the particular topic:




The screenshot displays a web application interface for filtering a question bank. At the top, a navigation bar includes a logo 'W3MTest' and several menu items: 'Set Model Test', 'Add MCQ', 'Show Questions', 'Participation History', 'Add Subject-Topic', and 'Mohanayok'. The main content area features a 'Filter Question Bank' dialog box. Inside this dialog, there are two input fields: 'Subject' with the value 'CSE' and 'Topic' with the value 'Programming'. A blue 'Filter' button is positioned at the bottom right of the dialog box.

After clicking the filter button , the following page is shown to him,containing all the questions from the selected subject & the selected topic:

W3MTest
Set Model Test
Add MCQ
Show Questions
Participation History
Add Subject-Topic
Mohanayok

Question: Which is the popular programming language for competitive programming ?

☐ java  
☐ c#  
☐ c  
☒ C++  
☐ php














C++ is very popular language for competitive programming.

Question: What is the output of the following code : " x = 3 ; x = ++x; print(x) "

☐ 2  
☐ 3  
☒ 4  
☐ 5  
☐ 6

## d)Show the Users Participation History:

After clicking the participation history he is taken into this page. This page contains exam id , exam date , exam subject , topic & his score.

| SELECT  | DATE-TIME           | SUBJECT | TOPICS             | SCORE |
|---|---------------------|---------|--------------------|-------|
|  | 2019-08-27 09:29:00 | English | Verbs, Noun, Voice | 4     |
|  | 2019-08-27 10:01:00 | English | Verbs, Voice, Noun | 0     |
|  | 2019-08-27 10:08:00 | English | Noun, Verbs, Voice | 0     |
|  | 2019-08-27 10:08:00 | English | Noun, Verbs, Voice | 1     |
|  | 2019-08-27 11:16:00 | English | Verbs, Voice, Noun | 4     |
|  | 2019-09-02 14:19:00 | English | Noun, Voice, Verbs | 1     |
|  | 2019-09-02 15:00:00 | English | Voice              | 0     |
|  | 2019-09-02 15:20:00 | English | Voice              | 0     |
|  | 2019-09-02 15:22:00 | English | Voice              | 0     |
|  | 2019-09-02 15:25:00 | English | Noun               | 0     |
|  | 2019-09-02 15:26:00 | English | Verbs              | 0     |
|  | 2019-09-02 15:28:00 | English | Verbs              | 1     |

In the select column , the id is shown correspondingly & the if the id is clicked , he is shown his answer script accordingly, like the following:

Question: Her promises \_\_\_\_\_ always a rope of sand for everyone.

☒ was

☐ is

☐ be

☒ are

☐ being



plural "promises"

Question: I checked all primary classrooms, no one was there. (Which word is a compound noun?)

☐ checked

☒ classrooms

☒ primary

☐ there

☐ none



b-classroom

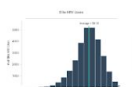
## ***e)User Profile:***

The subject wise strength of the user is shown in his profile:

The subject wise strength is shown in this page.

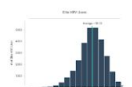


Mohanayok



English

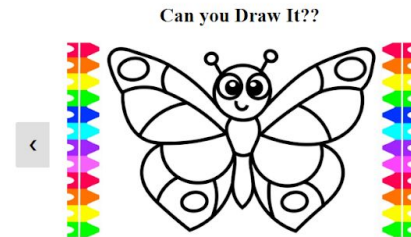
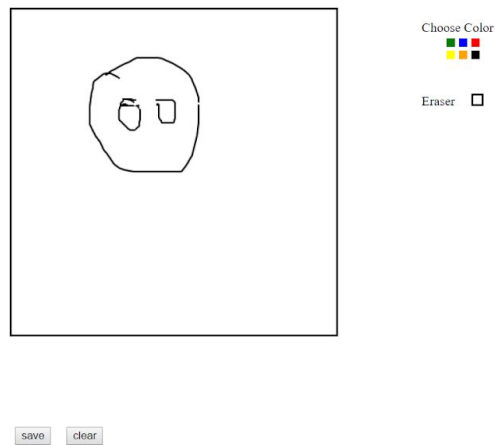
10.89%



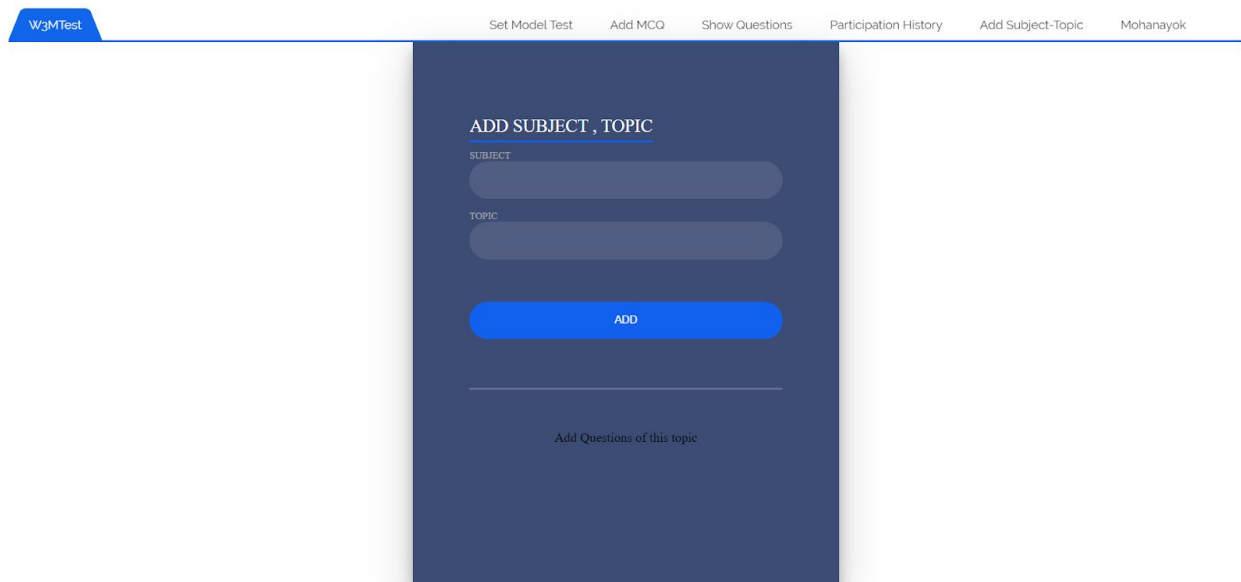
Physics

30%

### ***f) Kids Freehand Drawing:***



### ***g) Add New Subject-Topic:***

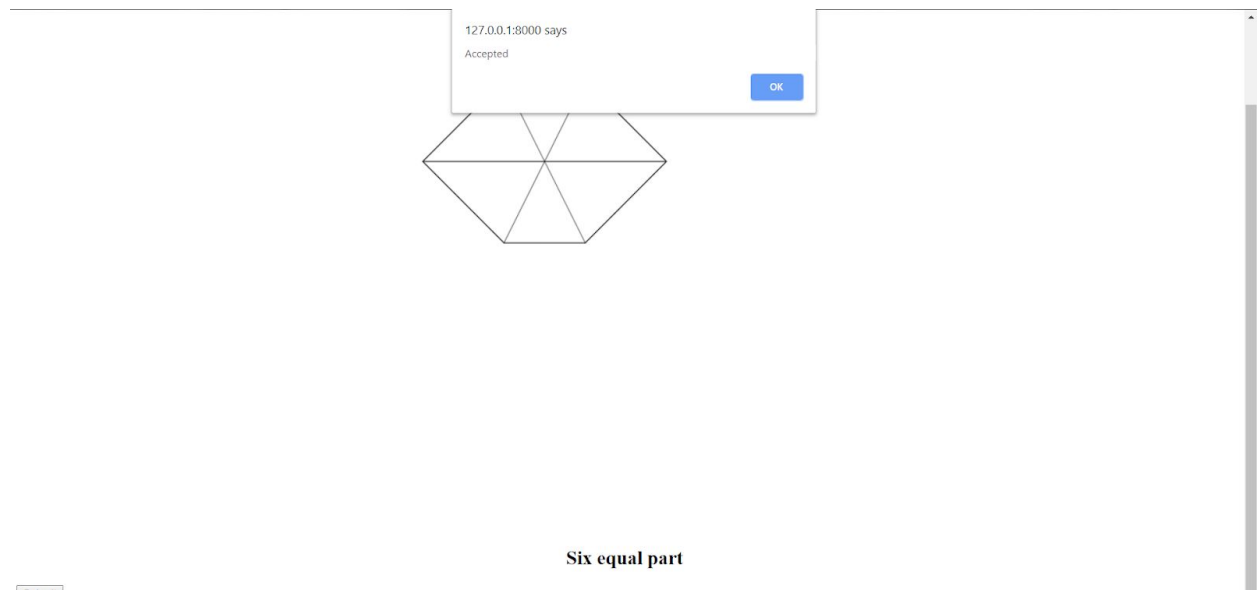


### ***h) All Users List:***

After clicking the users button he is taken into the following page:

Rana  
Anas  
Mohanayok

## ***i) Kids Geometrical Challenge:***



Right click & left clicking of the mouse draws a line from the right clicked point to the left clicked point.

## ***5) API Documentation***

API documentation is attached at the end of this article.



Welcome to the generated API reference. [Get Postman Collection](#)

## Drawing

APIs for drawing images of the given figure.

### Show

shows the front end. It contains the picture to draw & a painting tool to draw it.

Example request:

```
curl -X GET -G "http://localhost/kidszone"
```

Example response (500):

```
{  
  "message": "Server Error"  
}
```

### HTTP Request

GET kidszone

# Exam

APIs for participating in the exams according to the selected subject, topics and the number of questions, which user provides from the front end.

## Participate in the exam

Example request:

```
curl -X POST "http://localhost/myexams" \
  -H "Content-Type: application/json" \
  -d '{"subject":"sed","topics":"sed","NumberOfQuestions":6}'
```

### HTTP Request

POST myexams

#### Body Parameters

| Parameter         | Type    | Status   | Description                                       |
|-------------------|---------|----------|---|
| subject           | string  | required | the subject the examinee is willing to take part. |
| topics            | string  | required | Selected topics from that particular subject.     |
| NumberOfQuestions | integer | required | How many questions will appear in the exam?       |

## Filtering

APIs for filtering questions based on some subject / topics.

## Filter

Example request:

```
curl -X POST "http://localhost/FilterQBank" \  
  -H "Content-Type: application/json" \  
  -d '{"string":"enim"}'
```

### HTTP Request

POST FilterQBank

#### Body Parameters

| Parameter | Type  | Status   | Description  |
|-----------|-------|----------|--|
| string    | topic | optional | which topic's question the user is willing to filter |

## Show questions

shows all the questions in different view(blade)

Example request:

```
curl -X GET -G "http://localhost/ShowQBank"
```

Example response (500):

```
NAV III {  
  "message": "Server Error"  
}
```

## HTTP Request

GET ShowQBank

---

# Geometry

---

It gives some geometrical problems to the kids. Kids try to solve it by drawing & hence their geometrical knowledge is improved.

---

## Hexagone

Tells the user to draw one-sixth th of a hexagone;

Example request:

```
curl -X GET -G "http://localhost/hexagone"
```

Example response (200):

```
null
```

## HTTP Request

GET hexagone

## Triangle

Tells the user to draw one-third of a triangle.

Example request:

```
curl -X GET -G "http://localhost/triangle"
```

Example response (200):

`null`

### HTTP Request

GET triangle

## Triangle0

Tells the user to draw half of a triangle.

Example request:

```
curl -X GET -G "http://localhost/triangle0"
```

Example response (200):



## HTTP Request

GET triangle0

---

# MCQ

---

APIs for inserting mcq for the exam. Each mcq has problem statement , correct option, all other options , difficulty ,from which subject this question belongs to along with the topic.

---

## Index

Shows the application dashboard, most importantly the front UI.

Example request:

```
curl -X GET -G "http://localhost/addmcq"
```

Example response (401):

```
{  
  "message": "Unauthenticated."  
}
```

## HTTP Request

GET addmcq

## InsertMCQ

Example request:

```
curl -X POST "http://localhost/mcqInsert" \
  -H "Content-Type: application/json" \
  -d '{"question":"facere","image":"ea","subject":"inventore","topic":"enim","short_note":"numquam","d
```

### HTTP Request

POST mcqInsert

#### Body Parameters

| Parameter      | Type    | Status   | Description   |
|----------------|---------|----------|---|
| question       | string  | required | the problem statement                                 |
| image          | string  | required | if the question contains any image                    |
| subject        | string  | required | which subjects question the user is willing to filter |
| topic          | string  | required | which topic the question belongs to                   |
| short_note     | string  | required | short tutorial of about the answer                    |
| difficulty     | integer | required | the problem's difficulty in the range from 1 to 4     |
| correct_option | integer | required | correct option of the question                        |
| options        | string  | required | all options of the question                           |

## Participation

APIs for showing all the previous participations of the user. From which he can get an overall overview of his performance.

## Show all participations

shows all the participations of the user along with score, the exam's topic, subject and date-time.

Example request:

```
curl -X GET -G "http://localhost/showParticipations"
```

Example response (500):

```
{  
  "message": "Server Error"  
}
```

### HTTP Request

GET showParticipations

## Testing

For testing purpose

Example request:

```
curl -X GET -G "http://localhost/test1"
```

Example response (200):



null

## HTTP Request

GET test1

---

## Show all participations

shows all the participations of the user along with score, the exam's topic, subject and date-time.

Example request:

```
curl -X GET -G "http://localhost/myScript"
```

Example response (500):

```
{  
  "message": "Server Error"  
}
```

## HTTP Request

GET myScript

---

## Show participations according to the user input topics

shows all the participations of the user along with score, the exam's topic, subject and date-time depending on the selected topic

Example request:

```
curl -X POST "http://localhost/test2" \  
  -H "Content-Type: application/json" \  
  -d '{"string":"ea"}'
```

## HTTP Request

POST test2

### Body Parameters

| Parameter | Type  | Status   | Description   |
|-----------|-------|----------|---|
| string    | Topic | optional | Which topics participation the user is willing to filter? |

# Questions\_Showing

APIs for showing the options of the mcqs.

## Show all options of the question.

shows all the options in different view(blade) along with the corresponding question.

Example request:

```
curl -X GET -G "http://localhost/ShowQuestions"
```

Example response (200):

null

## HTTP Request

GET ShowQuestions

## InsertAnsSheet

Users anssheet is inserted in the database which contains the chosen options(by the user), his calculated score , the questions used in this exam.

Example request:

```
curl -X GET -G "http://localhost/ShowAnsSheet" \  
  -H "Content-Type: application/json" \  
  -d '{"string":"asperiores"}'
```

Example response (500):

```
{  
  "message": "Server Error"  
}
```

## HTTP Request

GET ShowAnsSheet

### Body Parameters

| Parameter | Type      | Status   | Description                                       |
|-----------|-----------|----------|---|
| string    | chosenOps | optional | which options the user has chosen in the mcq exam |

## InsertAnsSheet

Users anssheet is inserted in the database which contains the chosen options(by the user), his calculated score , the questions used in this exam.

Example request:

```
curl -X POST "http://localhost/ansSheetInsert" \  
  -H "Content-Type: application/json" \  
  -d '{"string":"eum"}'
```

### HTTP Request

POST ansSheetInsert

#### Body Parameters

| Parameter | Type      | Status   | Description                                       |
|-----------|-----------|----------|---|
| string    | chosenOps | optional | which options the user has chosen in the mcq exam |

## Script

APIs for showing the editorial page.

## ShowEditorialAndEverything

Shows details of the exam.

Example request:

```
curl -X POST "http://localhost/ShowScript"
```

## HTTP Request

POST ShowScript

---

# SetModelTest

---

APIs for adding new subjects & topics from the admin panel.

---

## Show the application dashboard.

Example request:

```
curl -X GET -G "http://localhost/setModelTest"
```

Example response (401):

```
{  
  "message": "Unauthenticated."  
}
```

## HTTP Request

GET setModelTest

---

## InsertMcqSheet

New exam is inserted in the exam's table.

Example request:

```
curl -X POST "http://localhost/setMTestInsert"
```

## HTTP Request

POST setMTestInsert

---

## Subject\_Topic

APIs for adding new subjects & topics from the admin panel.

---

## ShowSubjectTopic

Shows all the subjects & topics currently available in the database.

Example request:

```
curl -X GET -G "http://localhost/AddSubject"
```

Example response (500):

```
{
  "message": "Server Error"
}
```

HTTP Request

GET AddSubject

AddNewSubject/Topic

From this UI , the admin adds new subject & topic.

Example request:

```
curl -X POST "http://localhost/TestSubject" \
  -H "Content-Type: application/json" \
  -d '{"string":"aut"}'
```

HTTP Request

POST TestSubject

Body Parameters

| Parameter | Type  | Status   | Description               |
|-----------|-------|----------|---------------------------|
| string    | topic | optional | which topic just appeared |

## Show the application's login form.

Example request:

```
curl -X GET -G "http://localhost/login"
```

Example response (200):

```
null
```

### HTTP Request

GET login

## Handle a login request to the application.

Example request:

```
curl -X POST "http://localhost/login"
```

### HTTP Request

POST login



## Log the user out of the application.

Example request:

```
curl -X POST "http://localhost/logout"
```

### HTTP Request

POST logout

## Show the application registration form.

Example request:

```
curl -X GET -G "http://localhost/register"
```

Example response (200):

```
null
```

### HTTP Request

GET register

## Handle a registration request for the application.

Example request:

```
curl -X POST "http://localhost/register"
```

### HTTP Request

POST register

## Display the form to request a password reset link.

Example request:

```
curl -X GET -G "http://localhost/password/reset"
```

Example response (200):

```
null
```

### HTTP Request

GET password/reset

## Send a reset link to the given user.

Example request:

```
curl -X POST "http://localhost/password/email"
```

### HTTP Request

POST password/email

---

## Display the password reset view for the given token.

If no token is present, display the link request form.

Example request:

```
curl -X GET -G "http://localhost/password/reset/1"
```

Example response (200):

```
null
```

### HTTP Request

GET password/reset/{token}



## Reset the given user's password.

Example request:

```
curl -X POST "http://localhost/password/reset"
```

### HTTP Request

POST password/reset

## Show the application dashboard.

Example request:

```
curl -X GET -G "http://localhost/home"
```

Example response (401):

```
{  
  "message": "Unauthenticated."  
}
```

### HTTP Request

GET home

## users

Example request:

```
curl -X GET -G "http://localhost/users"
```

Example response (200):

```
null
```

### HTTP Request

GET users

## showprofile/{name?}

Example request:

```
curl -X GET -G "http://localhost/showprofile/1"
```

Example response (500):

```
{  
  "message": "Server Error"
```



## HTTP Request

GET showprofile/{name?}