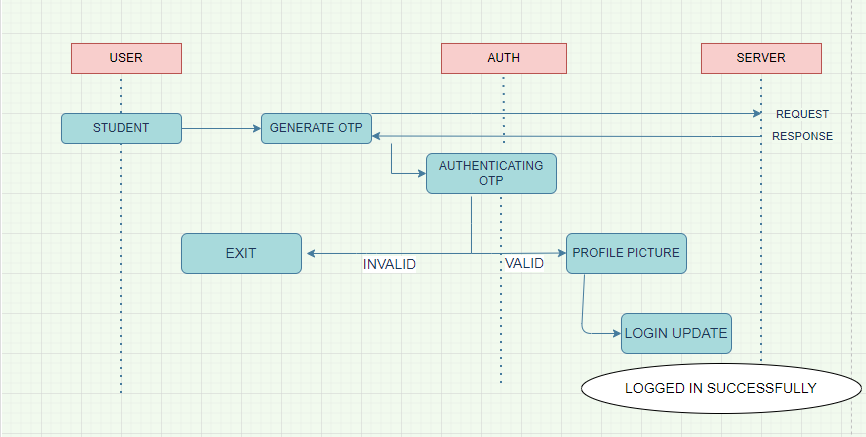
Up School Project Report

**LOGIN FLOWCHART**



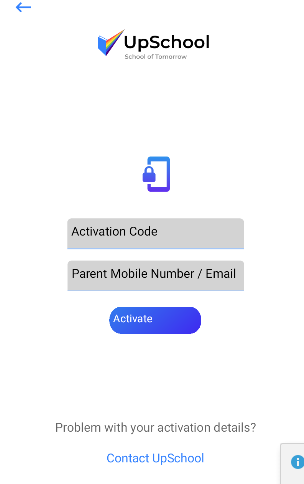
# **Student Android App – Generate OTP**

1. **Entering phone number to generate OTP**

Here the students have to login to the app by entering his phone number, and once the student enters the phone number the request goes to the backend, and the below Api is triggered, here this Api is a post Api, where we send the request and that request send an OTP as response to that phone number and the student has to enter the OTP to login.

**REQUEST -** post("/OTP/generate/")

**RESPONSE -** OTP sent successfully



# **Student Android App – Authenticating by OTP**

1. **Entering the OTP for authentication**

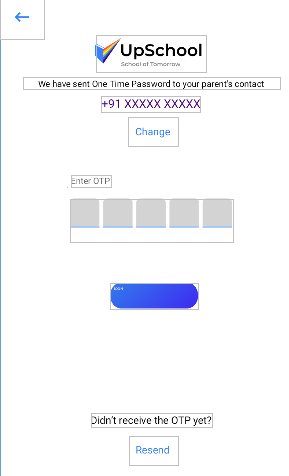
Here the student will receive the OTP to the number which he had entered in the previous page, this section is used to authenticate the OTP entered by student is right or not,

It is used for validating that this number belongs to that particular student, in this phrase the student will be logged In only if he enters the correct OTP that is sent to his number,

After checking the entered OTP is right then the student is rendered to next page that is the main page

**REQUEST** - POST("/activation-code/validate/")

**RESPONSE -** "message": "Device activated successfully!","statusCode":200



# **Student Android App – UPLOAD PROFILE PICTURE**

1. **Uploading the profile picture**

After generating the OTP, the student has to take his profile picture to continue with login process. Without uploading the profile picture, the student cannot move to the next page. After uploading the profile pic the student will be logged in successfully.

**REQUEST** - POST("/upload-picture/")

**RESPONSE -** data":{"name":"1629692882087.jpg","path":"users/3\_Sirish\_Student/profile\_picture/1629692889775\_1629692882087.jpg" "message" :"Image uploaded successfully!","status":"SUCCESS","statusCode":200 }

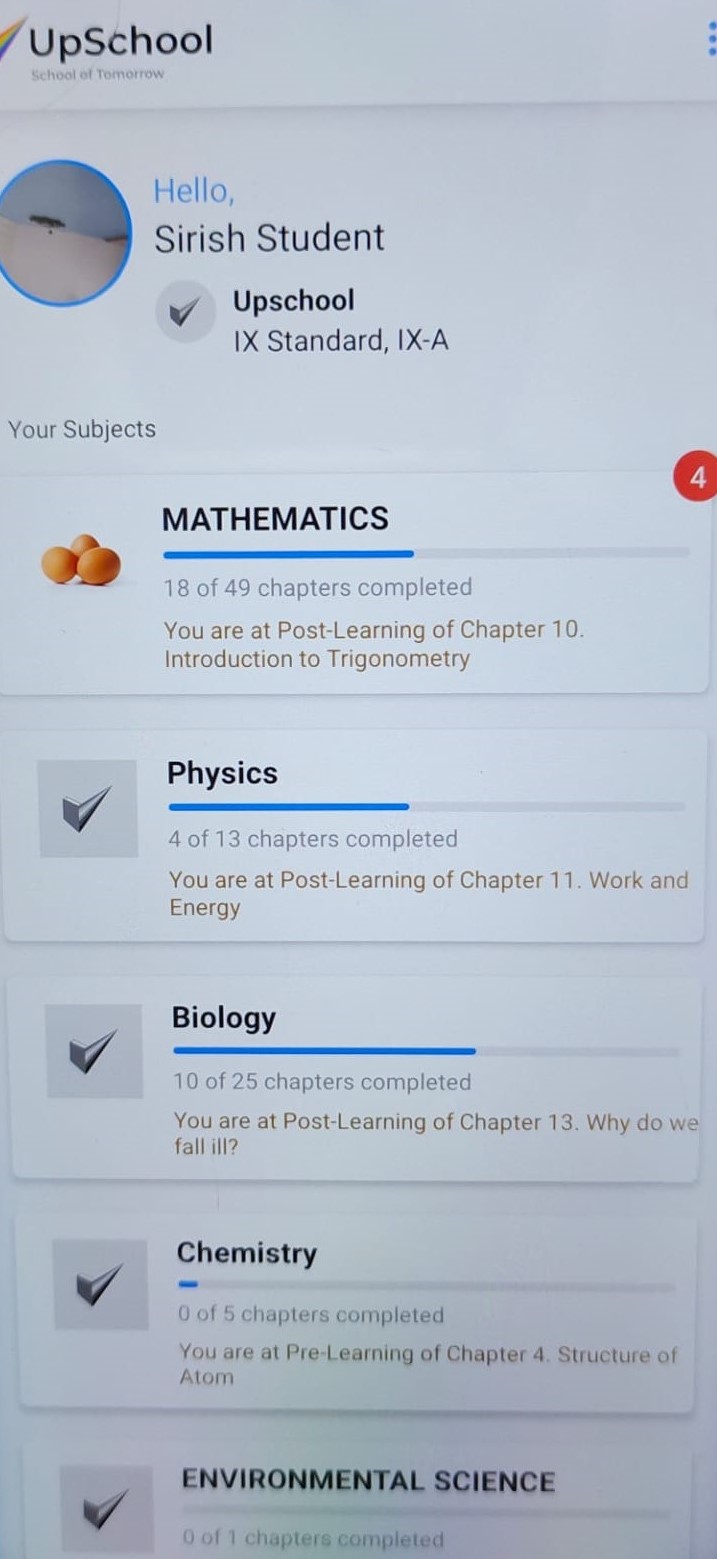
# **Student Android App – LOGIN UPDATE**

1. **Student Login Update**

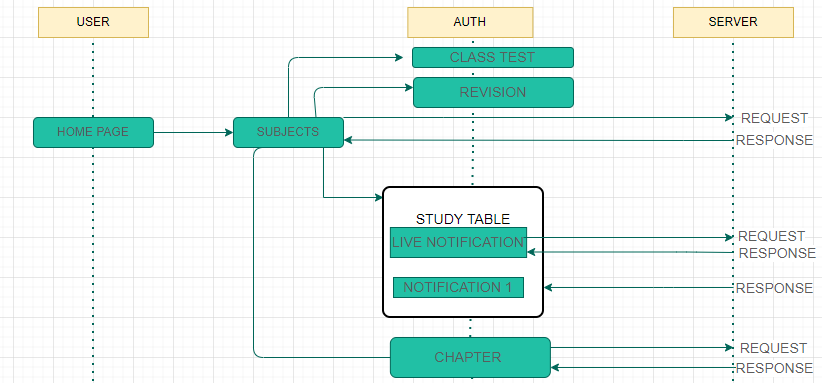
This page is rendered after the student logged In successfully, once the student enters OTP and after the validation of the OTP is done the student gets the current page that has the information of his class, with section, and the List of Subjects he has to attend

**REQUEST -** POST("/student/login-update/")

**RESPONSE -** "message": "User Login Updated Successfully!"



**FLOWCHART - 2**



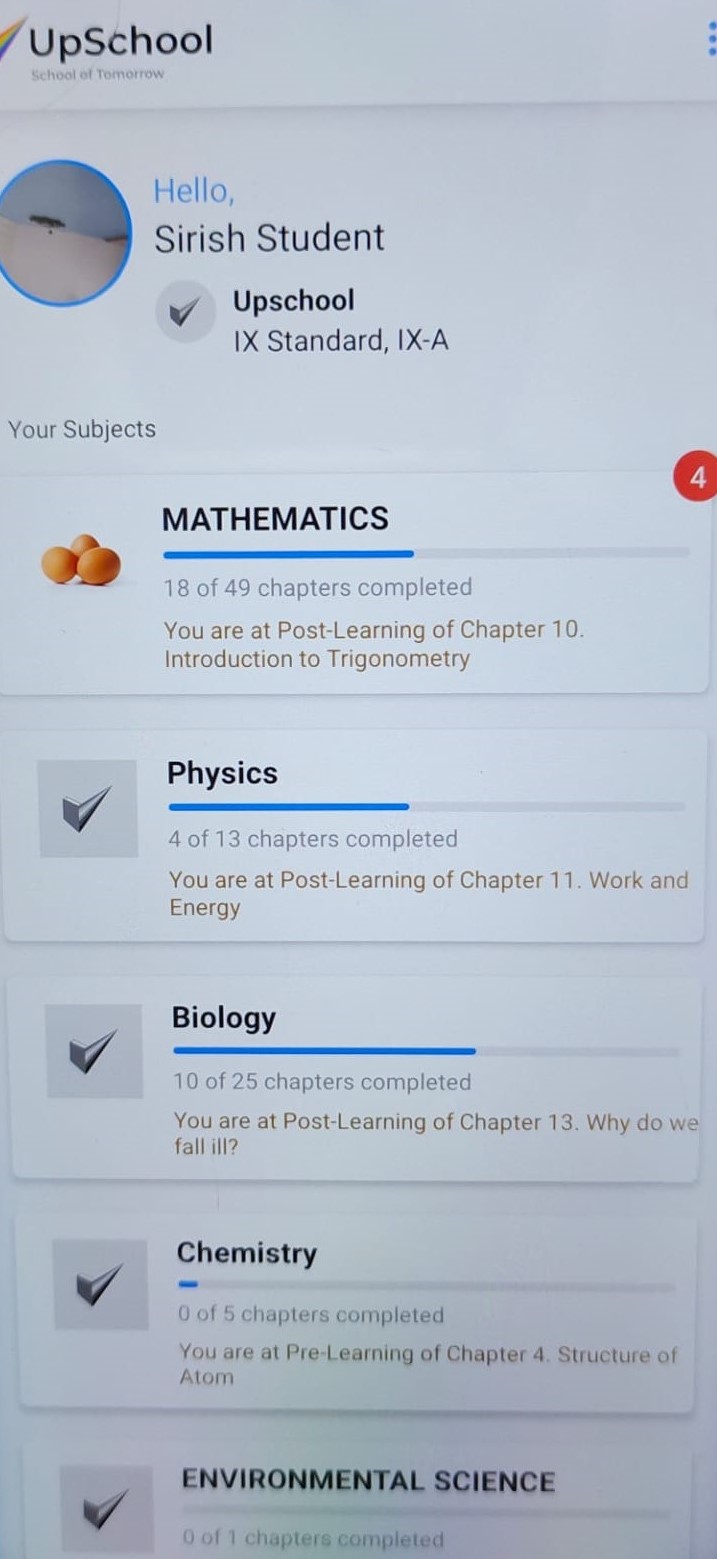
**Student Android App – GET SUBJECTS**

1. **Displaying the subjects of that Particular Student**

This Api is used to fetch the number of subjects of a particular student, once the student logs in to the home page, he is rendered with the number of subjects for the particular class he is studying, each subject has a particular Subject id. Each subject contains chapters in it.

**REQUEST** - GET("/student/subjects/")

**RESPONSE** - "id": 5, "name": "Physics", "icon": null, "status": 0, "total\_chapters": 13, "completed\_chapters": 4, "current\_status": "You are at Post-Learning of Chapter 10. Gravitation", "notification\_count": 0, "progress": 47,” statusCode”:**200**,” message”:” **Details fetched successfully”**



**Student Android App – GET NOTIFICATION**

1. **Displaying the Notification in study table**

When the student Clicks on the particular chapter he will be directed to Study table pagehere in this page the student will get the latest notification of that particular chapter, the recent updates that teacher has made will be seen in the study table, the student can go through the notification, Here the student can get Notification of chapter that related to pre-requisites(pre-learning) or post learning

In the Api we are passing the particular subject id because this gives us the notification of that particular Subject

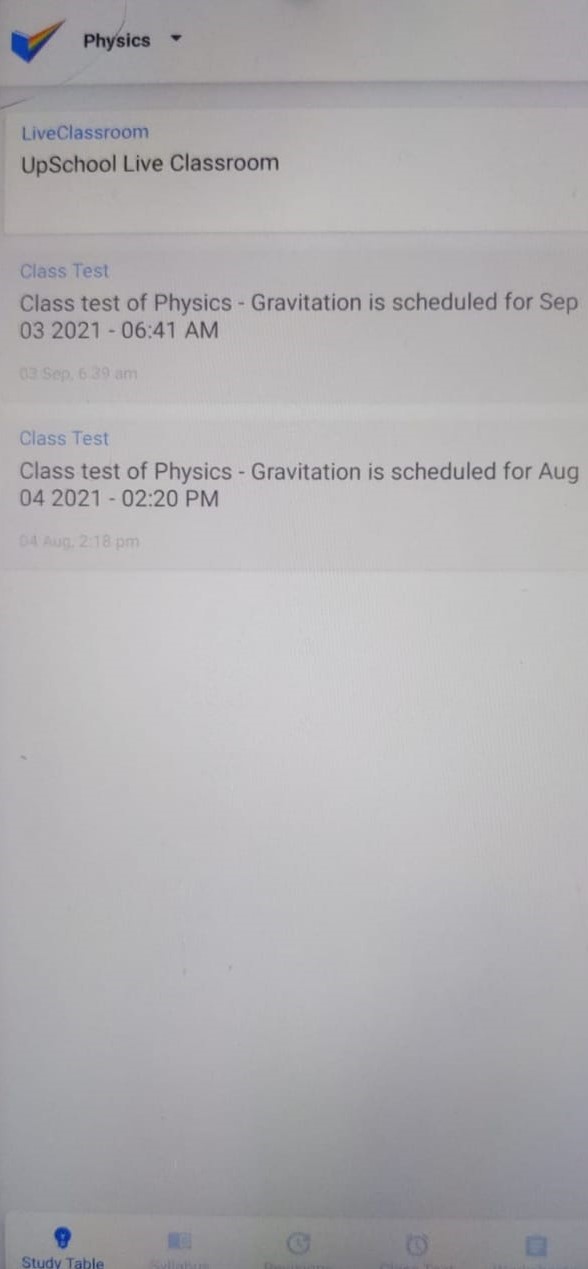
**REQUEST** - GET("/student/subject/{subject\_id} /notifications/?page=1")

**RESPONSE** - "notifications": [

{ "id": 7258,"status": 0,"chapter\_id": 520,"created\_at": "2021-12-14T10:42:19.346823Z","topic\_id": 1132,"action": 28658,"section\_id": 1,"work\_sheet\_id": null,"class\_test\_id": null,"action\_at": "2021-12-14 10:42:19","message": "Unlocked pushtopostlearn","target\_content\_type": "uptopic","tag": "Post-Learning","subject\_id": 0,"subject\_name": "","user\_id": 0,"meeting\_id": "default","notification\_type": "default","url": "default","expiry\_at": "","client\_id": "","client\_secret": ""

},

"statusCode": 200,"status": "SUCCESS", "message": "Notifications fetched successfully" ]



**Student Android App – NOTIFICATION COUNT**

1. **Displaying the recent 10 notifications**

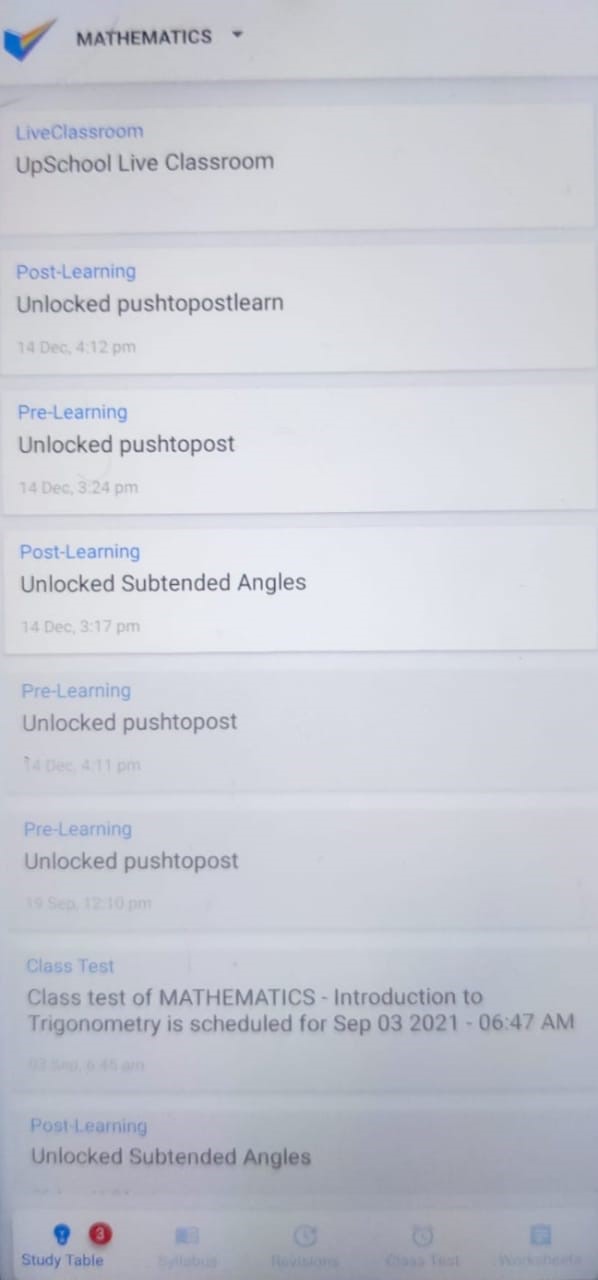
This Api renders the student with the recent 10 notifications, In the first part of notification it displays the Live classroom notification and the next 9 will be the recent updated notification

The maximum notification that the study table will displa**y**

**REQUEST** - GET("student/subject/{subject\_id}/notification-count/")

**RESPONSE** - "data": { "notification\_count": 2},

{ message": "Notification count fetched successfully!","status":"SUCCESS","statusCode":200}



**Student Android App – NOTIFICATION READ**

1. **Checks the Notification is Read or not**

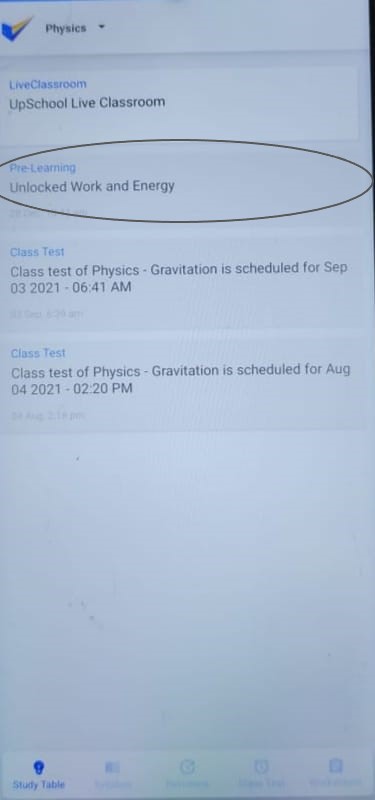
**This Api is used to Check whether the student has received and seen the notification, all the notifications sent by teacher is present in study table, the notification that not opened by the student will be in light grey, as soon as the student opens the notification this Api is triggered and sends the response that the student has seen the notification of particular subject**

**Each notification has a particular notification id and once the notification is read by the student the request goes to the backend stating that the student has seen the notification of that particular subject and in response will get the message notification read successfully**

**REQUEST -** POST("/student/subject/{subject\_id}/notification-read/{notification\_id}/")

**RESPONSE -** "statusCode": 200, "status": "SUCCESS", "message": "Notification read successfully!"

**Notification before read Notification after read**

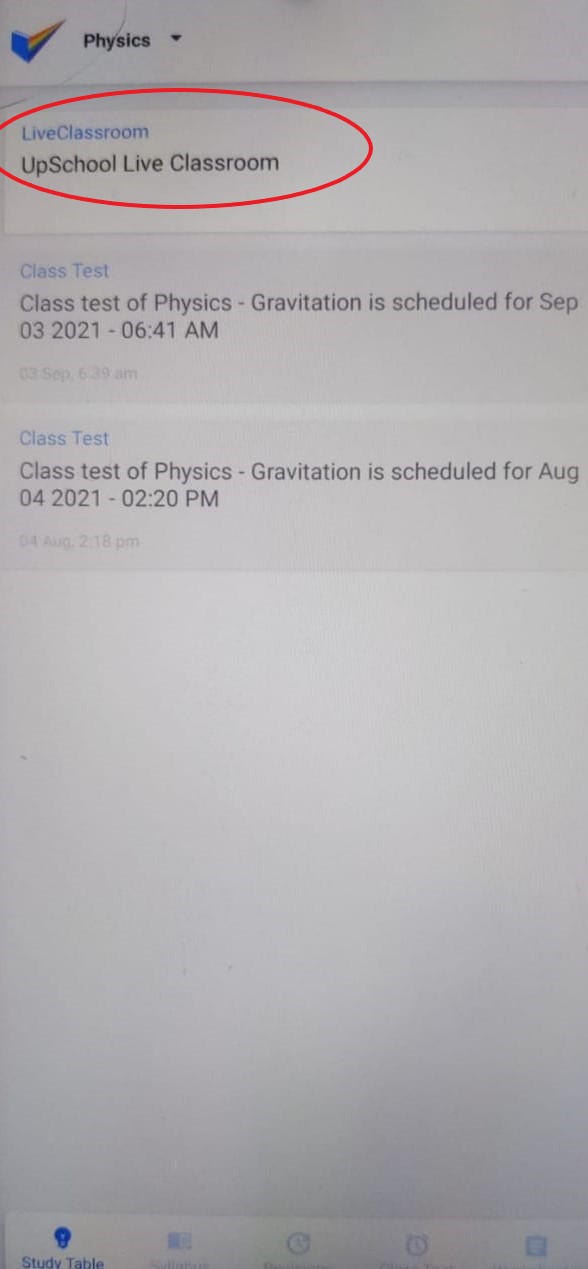


**Student Android App – LIVE NOTIFICATION**

1. **Displaying live notification using Airlearn**

In the upschool android app the student can interact with teachers using the live notification here the student and the teacher will have a live video conversation, and the teacher can take class remotely for the students. The live notification is always placed in the top of the study table module

**REQUEST -** GET("/student/subject/{subject\_id} /user\_id/{user\_id}/livenotifications/ “)

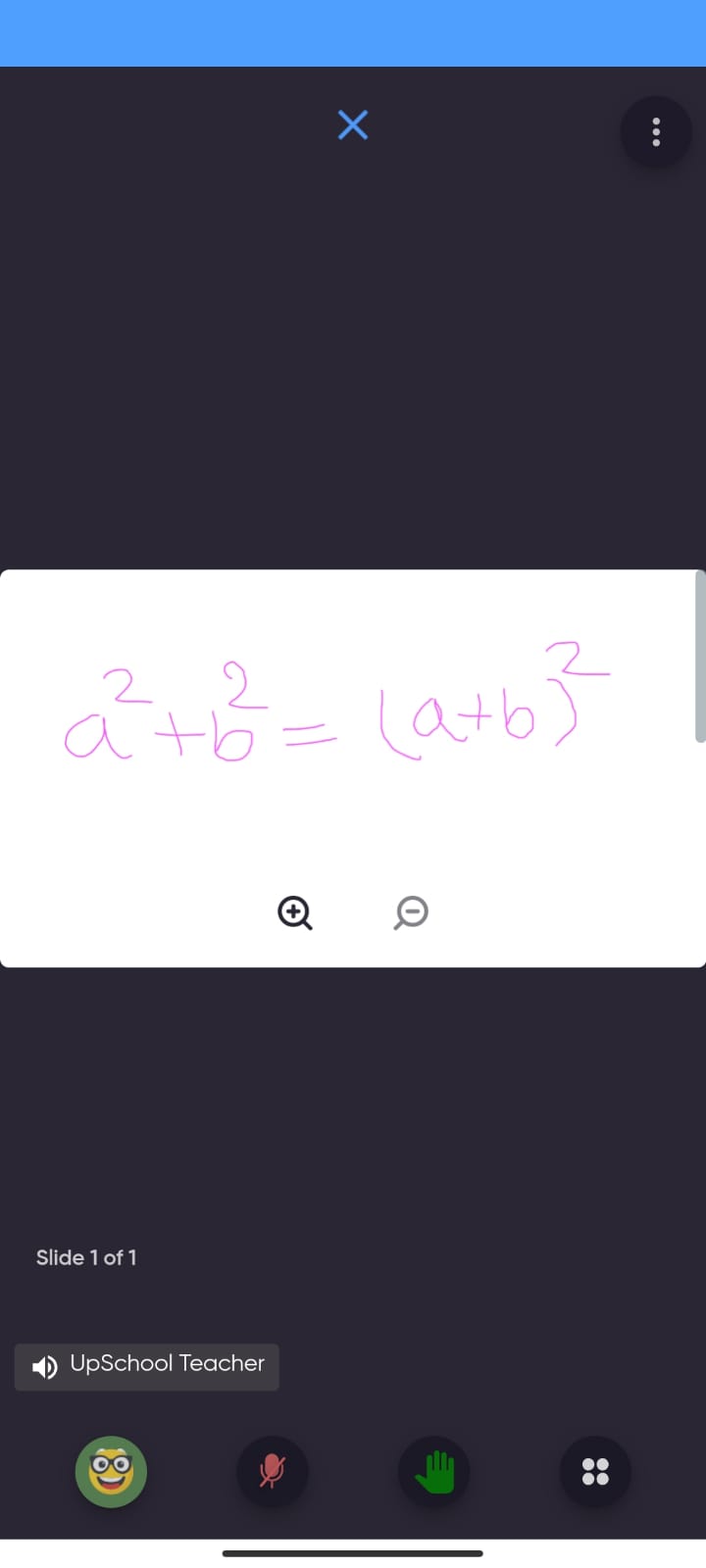
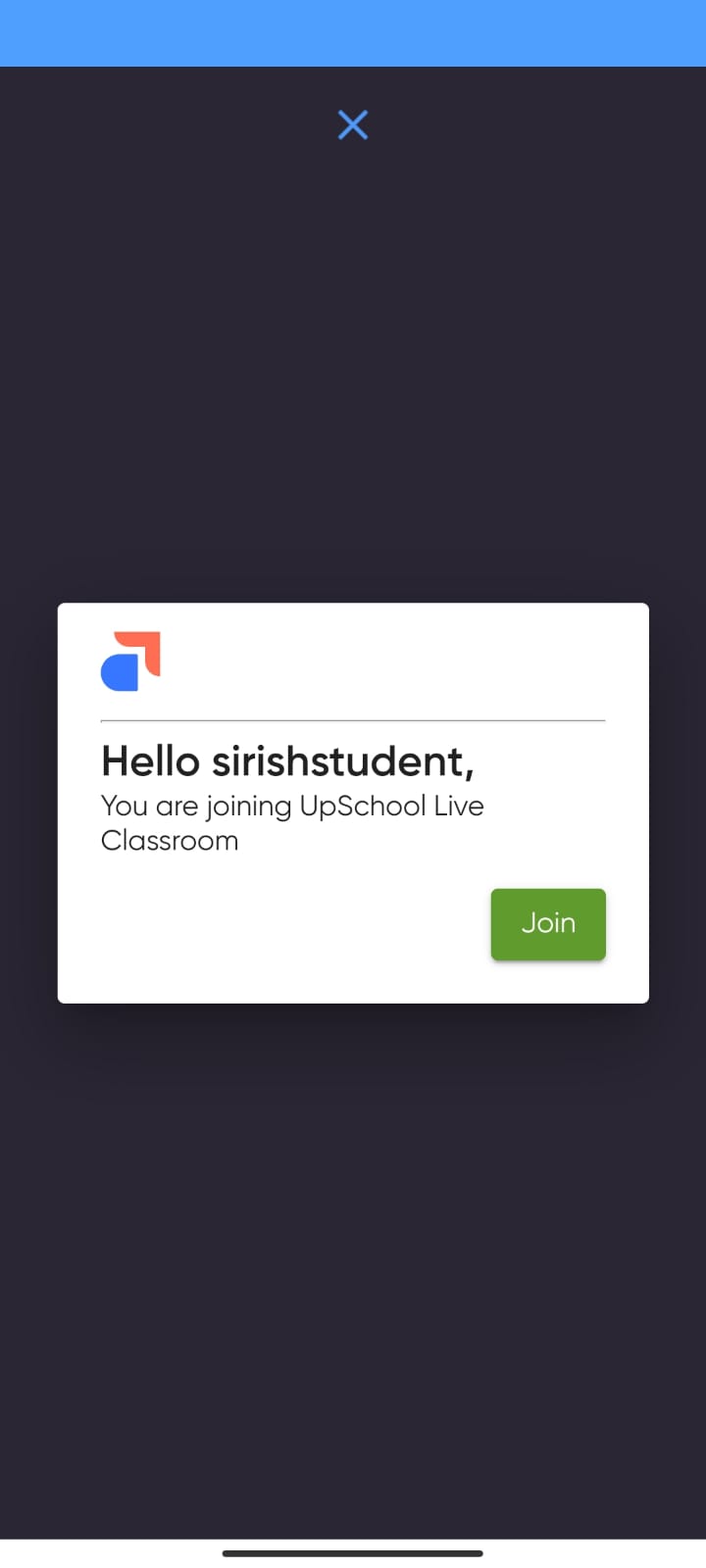


**Student Android App – AIR LEARN**

1. **LIVE NOTIFICATION USING AIRLEARN**

For this live class room, we have used an external third-party exposed API service for livestreaming called Airlearn. Here the student gets the notification and the student logs into the student app and clicks on the live classroom notification. The request will hit the Backend server and the router logic will ensure the request is from Student portal and routes the request accordingly. Within the Student Portal backend service, the URL will be identified that this request corresponds to Create Live Classroom and corresponding method in this case getLiveNotifications will be invoked which is found in the views.py and within this method it invokes required Serializer -> Models to fetch details from Database and once the data is fetched it does all the business logic and serializer responds/constructs with the required json response. The process of querying and converting tabular database values into JSON or another format is called **serialization**.

**REQUEST- @POST("https://ams-lb.airlearn.ai/api/joinMeeting")**



**Student Android App – GET CHAPTERS**

1. **Displaying the chapters for the subject**

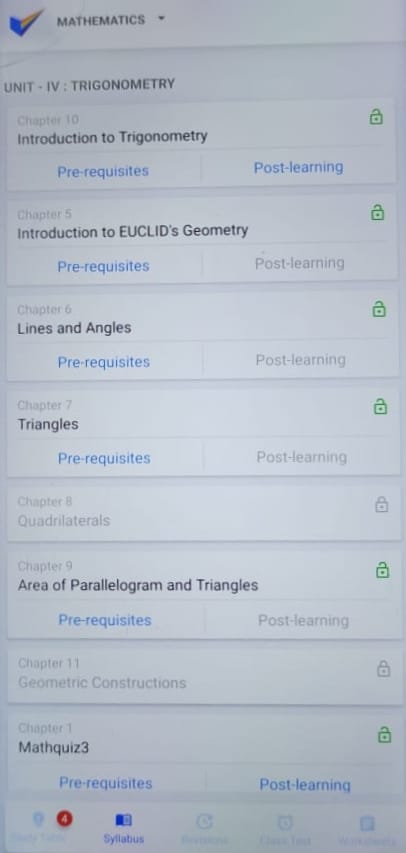
This Api is used to render the chapters of the particular subject, when the student Logs in he is rendered with subject based on the class he studies, here the student selects the particular subject and then the student gets the Number of chapters for that particular subject. Here we pass the subject id to fetch the chapters of that particular chapter and each chapter contains chapter id.

the student has to go to the syllabus pat in the footer to see the number of chapters present for that particular subject and each subject has a pre-requisites and post learning

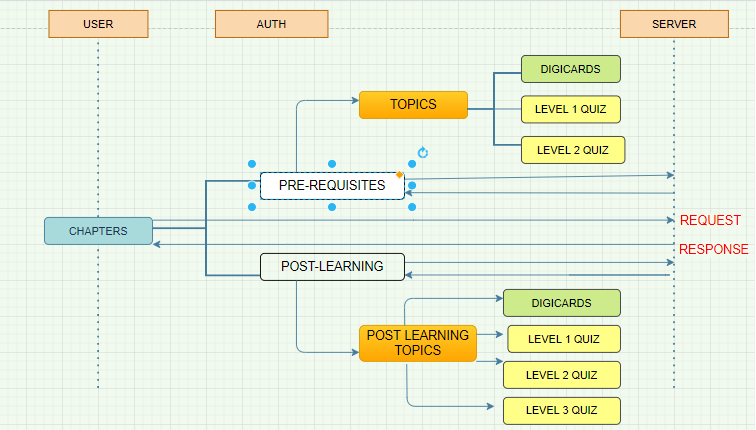
**REQUEST** - GET("/student/subject/{subject\_id}/chapters/")

**RESPONSE** - "id": 20, "index": 4, "title": "TRIGONOMETRY", "chapters": [{"id": 238, "index": 10, "title": "Introduction to Trigonometry", "status": 1, "is\_post\_learning\_enabled": true, "is\_locked": false,

“status”: ”SUCCESS”, “statusCode”:200,”message”:” Details fetched successfully”



**FLOWCHART 3**



**Student Android App – GET TOPICS**

1. **Displaying the Topics of that particular Chapter**

This Api is used to fetch the topics of that particular Chapter, each Chapter will have a number topics, when the student goes to the syllabus and then clicks the particular chapter, each chapter will have a pre-requisite (pre-learning) and post-learning, in this pre-learning the student will have a number of topics.

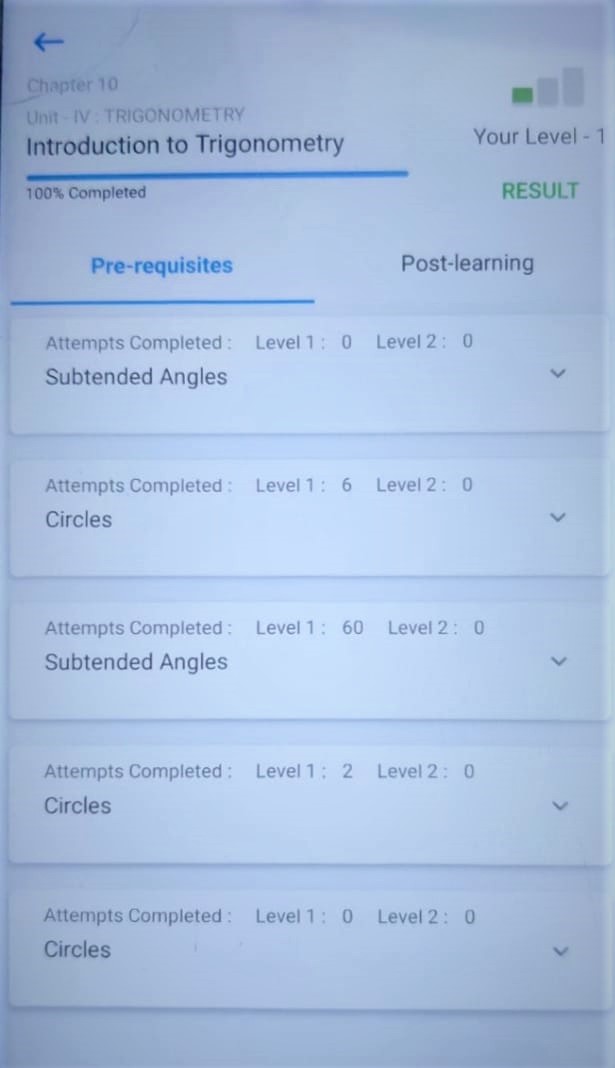
Each topic has a Digicard and quiz , the student has to go through the digicard and then he can take up the quiz for that particular topic there will be 2 level of quiz for each Topic, the student has to take this 2 levels of quiz to complete this Topic

**REQUEST** - GET("/student/subject/{subject\_id}/chapter/{chapter\_id}/topics/")

**RESPONSE -** topics": [{"id": 1065, "title": "Subtended Angles", "index": 1, "status": 0, "digital\_card\_status": 0, "deadline\_date": null, "quiz": [{"level": 1, "is\_locked": true, "attempts": 0, "is\_attempts\_exceeded": false, "status": 0, "attempt\_left": 2, "total\_quiz\_attempts": 0, "failed\_quiz\_attempts": 0, "need\_attention\_count": 0, "need\_attention\_count\_must": 0},

{ "level": 2,"is\_locked": true,"result\_seen": false,"attempts": 0,"status": 0,"need\_attention\_count": 1,"is\_attempts\_exceeded": false,"pinpoint": 2,"attempt\_left": 2,"total\_quiz\_attempts": 0,"failed\_quiz\_attempts": 0, "need\_attention\_count\_must": 0 }]

status”: ”SUCCESS”, “statusCode”:200,”message”:” Details fetched successfully”



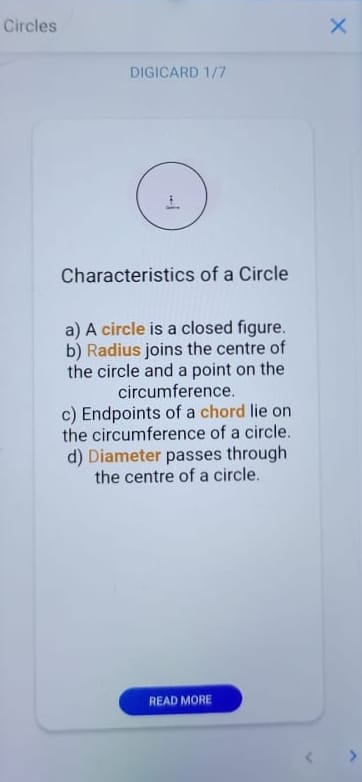
**Student Android App – DIGICARDS**

1. Displaying the Digicards

When the student Clicks on Topics, the student gets the number of topics for that particular Chapter and each topic will have the Related digicards, this digicards contain important points that the student has to go through, to unlock the first level of quiz the student has to go through this digicards , at the end of the digicards he is asked - to take Quiz, the Student cannot take Quiz until he finishes the digicards

**REQUEST** - GET("/student/topic/{topics\_id}/digital-cards/")

**RESPONSE** - "topic": "importtin", "digital\_cards": []}, "statusCode": 200, "message": "Digital Cards Fetched successfully!"



**Student Android App – RELATED DIGICARDS**

1. **Displaying the Related Digicard(Particular digicard)**

This Api is used to get the digicard of particular Api of the particular topic, to fetch the particular/related digicard we will pass the Topic id to the path so that we will get the Digicard of that particular topic

REQUEST - GET("/student/digital-card/{digi\_id}/related/")

RESPONSE - message: "Related Digital Cards Fetched successfully!","statusCode":200}

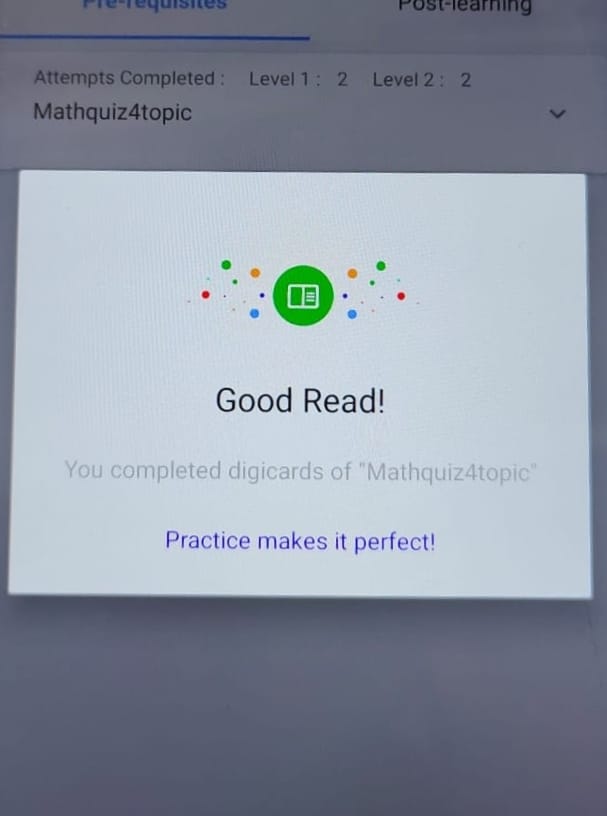
**Student Android App –COMPLETED DIGICARDS**

1. **Displays if digicard reading is completed**

This Api is triggered at the end of the digicard , if the student completes the digicard then the student at the end of the digicard the Student has to click the Button MARK COMPLETE, once he clicks on this button, this Api is triggered, here it displays the student that he has completed the digicard and then he will be able to take up the quiz

**REQUEST** - GET("/student/topic/{topic\_id}/digital-cards-completed/")

**RESPONSE** - {"data": {"next\_quiz\_level": null, "attempts": 0}, "status": "SUCCESS", "statusCode": 200, "message": "Topic's Digital Cards completed"}



**Student Android App –QUIZ**

1. **Displaying the quizzes**

After Completing the Digicards the student have to go for the quizzes, the quiz can be taken only after completing the digicards , from the teacher portal,

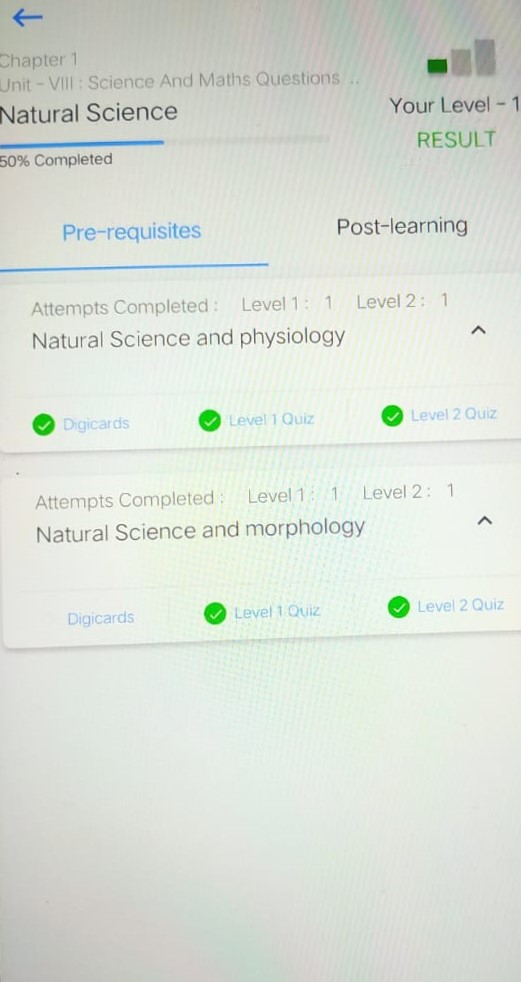
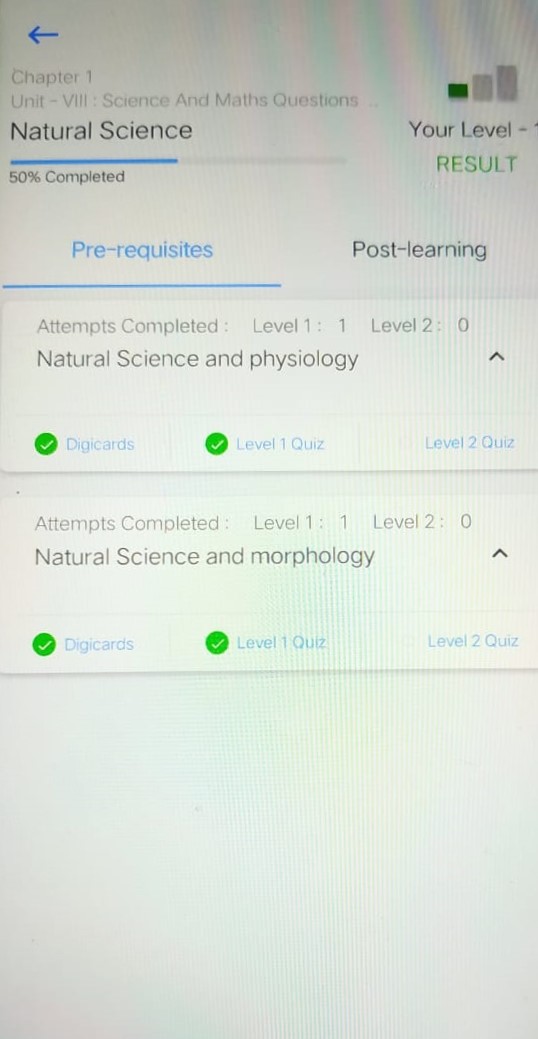
the teacher updates the quizzes for the particular chapter, in the app side the student will get to know when he goes to that particular quiz

**Pre-requisites** - In pre-learning the students will have 2 levels of quiz. these quizzes are of chapter evaluation that is it contains questions of a particular chapter

Here the student has maximum of 2 levels of quiz, where each level has 2 attempts, if the students fail the level 1 quiz in the first attempt, then he gets another chance of taking the quiz. this will be his last attempt of taking quiz, if he clears the quiz then he can go forward for level 2 quiz, if not the students fail the 2nd level of quiz, then he will not be able to take the quiz again.

After the student fails the 2nd level of quiz it goes the **Need attention** in teacher's portal, here the teacher has to take a special attention for that particular student, the teacher has to unlock the quiz from the teacher's portal site, only then the student can take the quiz again,

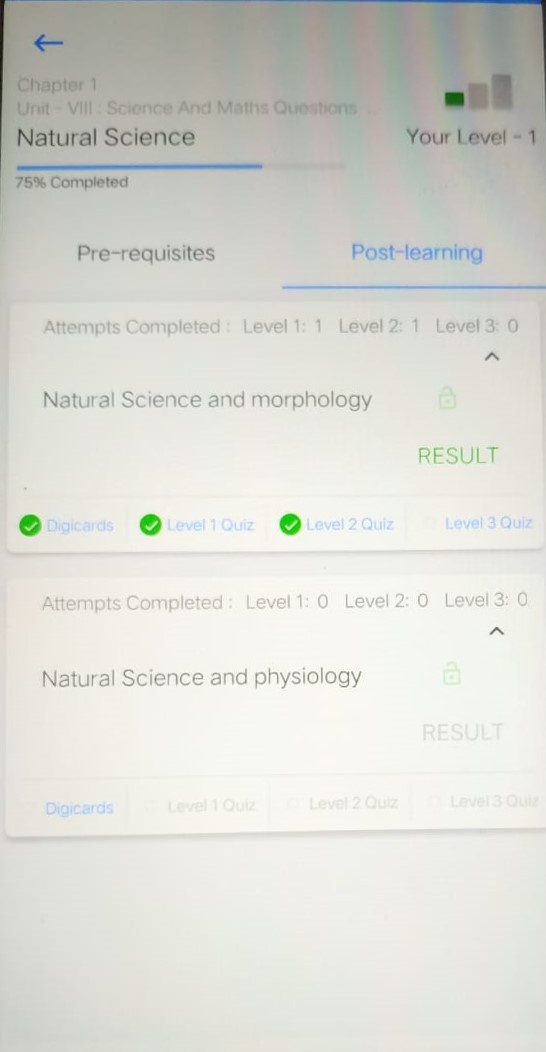
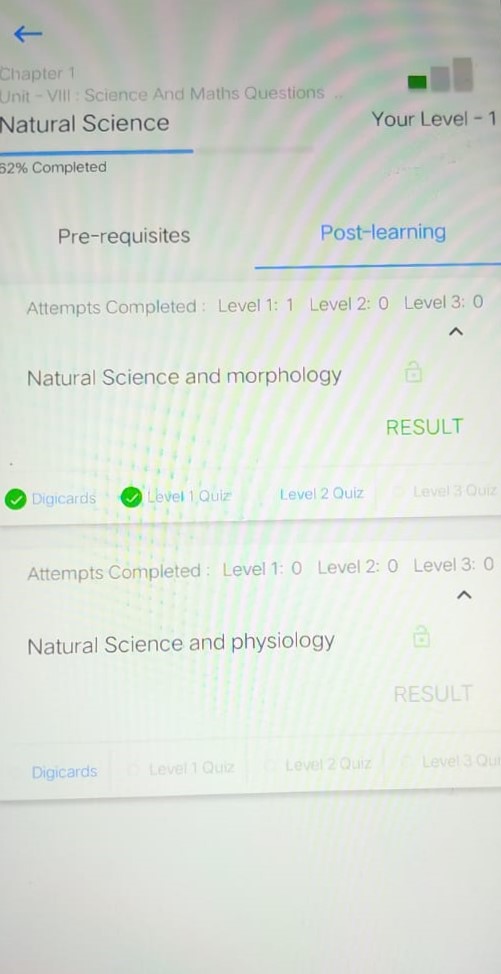
If the student completes the level 1 quiz, then he is considered as a level 1 student, and then the student can go to the level2 quiz, here also the student has 2 attempts to take up quiz, If the student fails in the 2nd attempt, then he will not get another chance to write the quiz, he will be promoted to post learning as a level 1 student.



**Post- learning –**  In post learning the quizzes will be of topic evaluation, post learning contains 3 level of quizzes

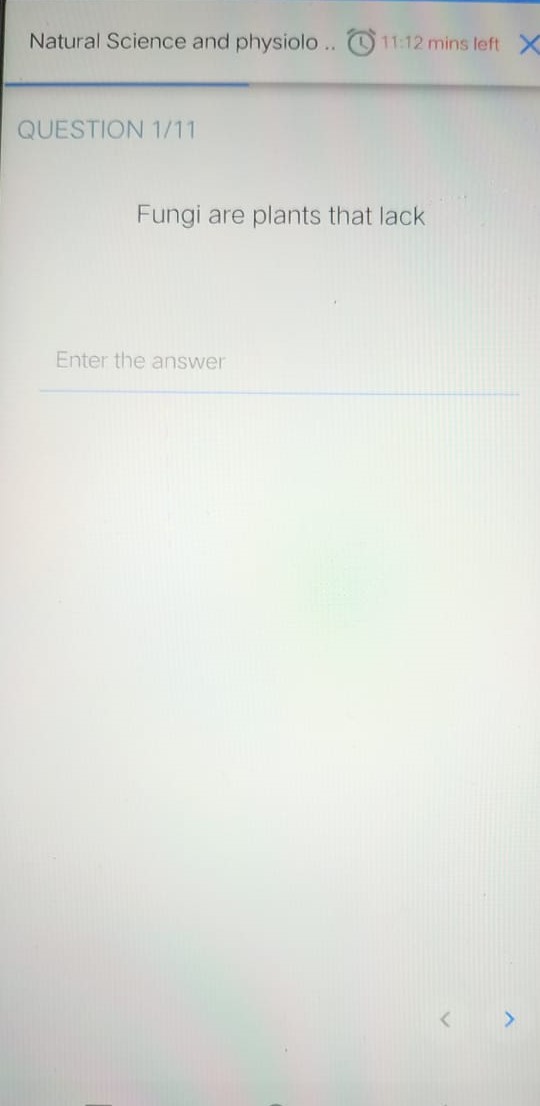
Here the student will be considered as a level 1 student if he has not completed the level 2 quiz of pre-learning, he has to go from level 1 again in post learning, here the student will have 2 attempts and again it goes to need attention in teacher portal side if he fails in the 2nd attempt,

If the student has completed the 2nd level of pre-learning then he is considered as a level 2 student ,and then in post learning he is skipped with the level 1 quiz, he is directly passed on to level 2 quiz, if the student fails the level 2 quiz, then he will be considered as a level 1 student of post learning and then he had to clear the level one quiz to get back as a level 2 student



**REQUEST -** GET("/student/topic/{topics\_id}/level/{level\_id}/ generate-quiz-questions/")

**RESPONSE -** "title": "pushtoposttopic", "duration": 20, "quiz\_id": 6367, "no\_of\_show\_math\_keyboard": 0}, "questions": [{"id": 3190, "question": " <p>1+2=\_\_\_\_\_\_\_\_\_</p>", "description": null, "number\_of\_answers": 1, "group": 3428, "answer\_label": null, "option\_type": 2, "options": [], "attachments": [], "show\_math\_keyboard": false, "answer\_disclaimer": "<p>yes</p>", "show\_disclaimer": true}]}, "status": "SUCCESS", "statusCode": 200, "message": "Details fetched successfully"}



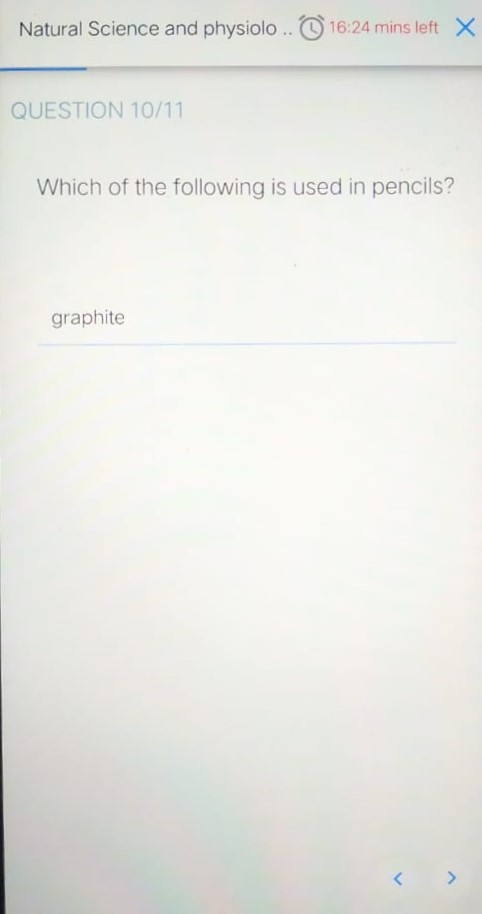
**Student Android App –SUBMITTING QUIZ**

1. **GETTING THE ANSWER FOR THE QUIZ**

After the question received from the teacher portal to the student portal, the student answer the question, this answer are the sent to the back end and then they are pushed to the teacher portal, after the student finish all the quiz the student will click on the **submit button**, onclick of the submit button the student will get result of the particular quiz. this validation of quiz takes place at the backend and he gets the result on the same day. The result can be viewed only once, it will not be available after viewing the result more than once

**REQUEST -** POST("/student/topic/{topic\_id}/level/{quiz\_level}/quiz/{quiz\_id}/submit-quiz-answer/")

**RESPONSE** - {"message": "Answer submitted successfully","status":"SUCCESS","statusCode":200}



**Student Android App –RESULT**

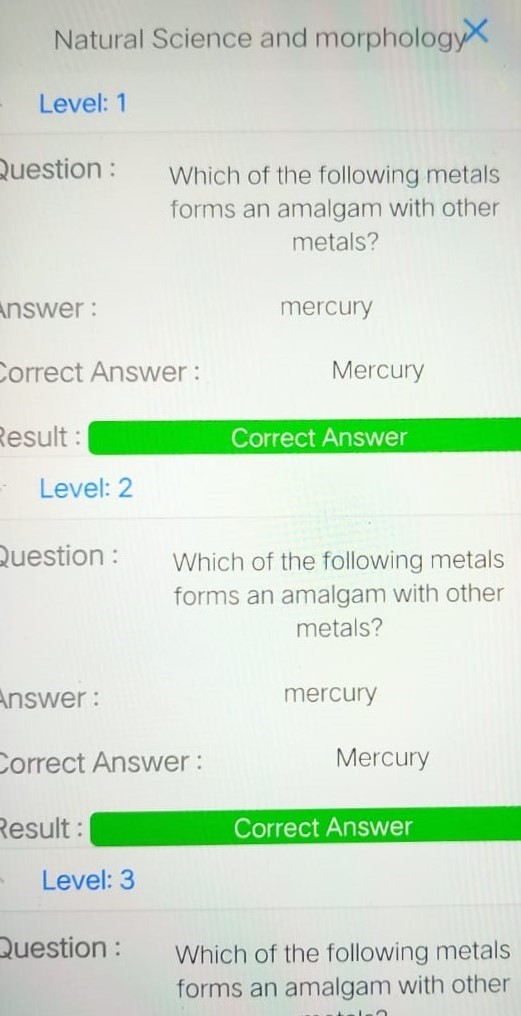
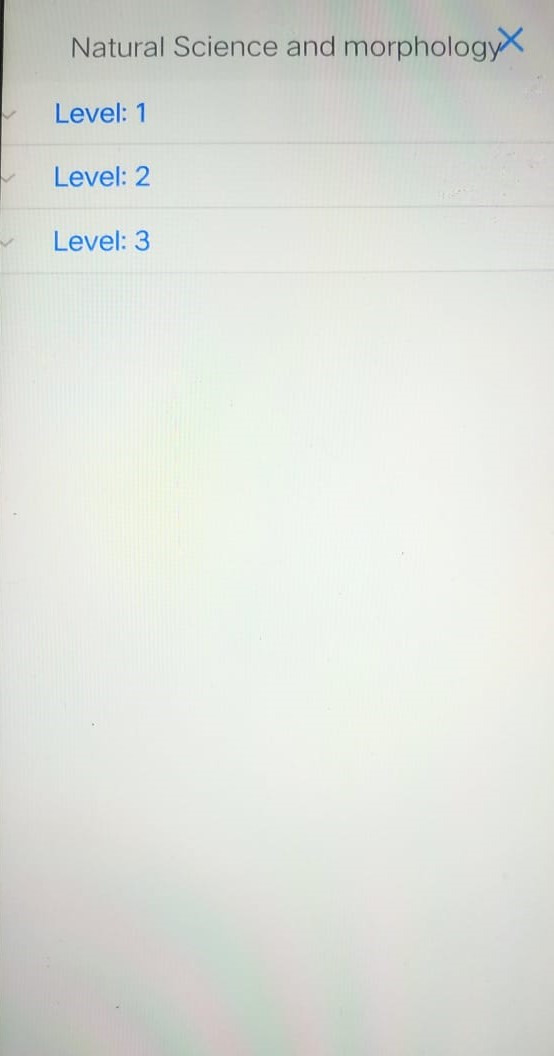
1. **GETTING THE RESULT FOR THE QUIZ**

After answering the question in the last page there will be submit button on click of the submit button the student will receive the result, Here the results are evaluated after the student has complete his quiz, in the back-end the answer written by the student and the answer given by the teacher are checked and then it gets evaluated. if both the answer matches then the student we get a mark. In the result section the answer given by the teacher, and the answer written or selected by the student is displayed

this result will be displayed only once and the student will not be able to view the result again.

**REQUEST-** GET("/student/topic/{topic\_id}/learning\_type/{learning\_type}/quiz\_level/{quiz\_level}/result \_status/")

**RESPONSE -** "message": "Result seen","status":"SUCCESS","statusCode":200



**Student Android App –RESULT FROM TEACHER**

1. **Displaying the results that teacher updated**

This Api is used to see the results that the teacher has updated for that particular student.

This Api requires Student id to know the result of particular student, and topic id for topic

**REQUEST-** POST("/teacher/topic/{topic\_id}/student/{student\_id}/")

**RESPONSE** - "body": {"data": {"id":1132,"index":1,"learning\_type":2,"quiz":[{"answer\_disclaimer":"","answer\_for\_display":"","answer":"5,"id":3191,"option\_type":2,"options" :[] "message": "Details fetched successfully","status":"SUCCESS","statusCode":200

**Student Android App – Average Daily Usage**

1. **Displaying the average usage of the students**

Here the student's average daily usage is taken up here, the daily usage is taken only when the student logs in and takes up the quiz and any other activity, it does not Calculates when he starts the app,

Here the average daily usage is given to particular student for a particular chapter

In average daily usage the we pass the student id for that particular student and subject\_id so that we can get to know that the time spent by the student for that particular subject.

The average daily usage is taken count when the student takes up the quiz and while reading the digicards (Read more)

**REQUEST -** POST("/student/user/{student\_id}/subject/ {subject\_id}/average-daily-usage")

**RESPONSE -**  "status":"SUCCESS","statusCode":200}, "message": "OK"

**Student Android App – REVISIONS**

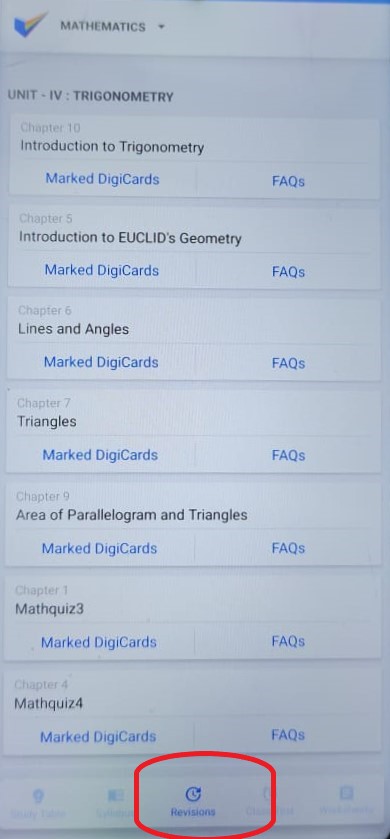
1. **Displaying the chapters for revision**

Here this API is used to get that chapters that has to be revised by the student, This module is used to view and revise the digicards which are marked using mark for revise button in digicards of postlearning topics only.

In the UPSCHOOL Android app the student has a revision page, this module contains the topic that has to be revised by the student where the student clicks on a button in digicard for revise more,once he clicks the button the topic in the post learning comes to the revision module, revision module contains a list of topics that has to be revised for particular topic

**REQUEST -** GET(" /student/subject/{subject\_id}/revision-chapters/")

**RESPONSE -**  "status":"SUCCESS","statusCode":200}, "message": "OK"



**Student Android App – ADD REVISIONS**

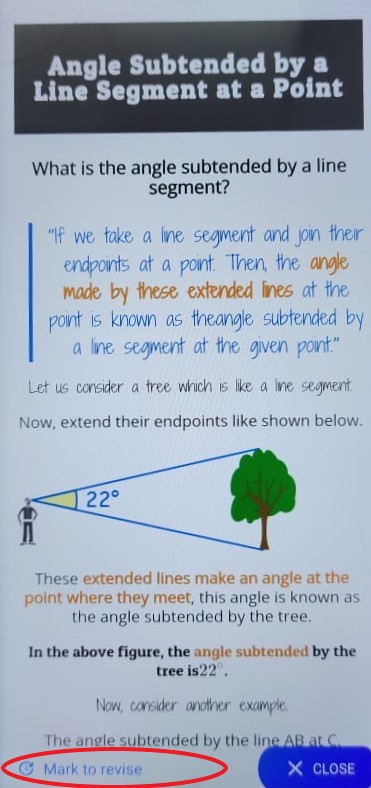
1. **ADDING CHAPTERS TO REVISION FROM DIGICARD**

Here this API is used to add the chapters from digicards to revisions .In the post learning the student will have an option to add digicards to

In the UPSCHOOL Android app the student has a revision page, this module contains the topic that has to be revised by the student where the student clicks on a button in digicard **MARK TO REVISE** ,once he clicks the button the topic in the post learning comes to the revision module

**REQUEST -** GET("/student/digital-card/{digicard\_id}/add-to-revision/")

**RESPONSE -** "message": "Digital Card added to revision successfully!","statusCode":200}



**Student Android App – REMOVE REVISIONS**

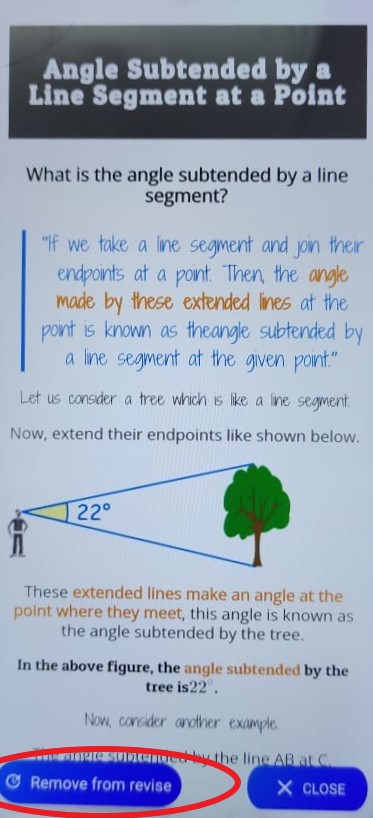
1. **REMOVING CHAPTERS TO REVISION FROM DIGICARD**

Here this API is used to remove the digicard from revision module. In the post learning the student will have an option to remove those digicards that was added before in the digicards module in post learning.

In the UPSCHOOL Android app the student can remove the digicards in the revision module, once the student finishes the digicard that are present in the revision module he can remove the digicard from a button that is placed in the last page of the digicard “**REMOVE FROM REVISE**”. Onclick of it the digicards comes out of the revision module

**REQUEST -** GET("/student/digital-card/{digicard\_id}/remove-revision/")

**RESPONSE -** "message": "Digital Card removed from revision successfully!","statusCode":200}



**Student Android App –FREQUENTLY ASKED QUESTION (FAQ’S)**

1. **DISPLAYING THE FREQUENTLY ASKED QUESTION**

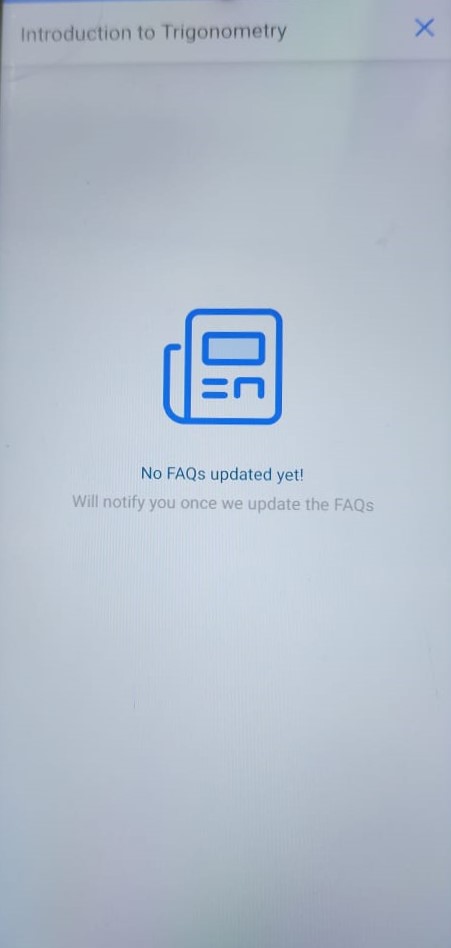
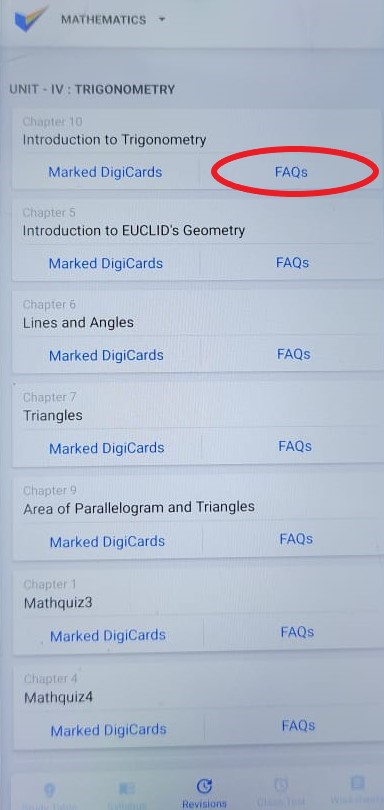
In the revision module we have chapters that has to be revised, after the revising those chapters there is a section below those chapters called FAQ’S. This section contains questions that are frequently asked by the teachers.

All the important questions of that particular chapter are asked in this section.

The students after revising the digicard can go through this questions that are related to digicards of that particular chapter

**REQUEST -** GET("/student/subject/{subject\_id}/faq/")

**RESPONSE -** "statusCode": 200, "status": "SUCCESS", "message": "FAQs are fetched successfully"



**Student Android App – WORKSHEET**

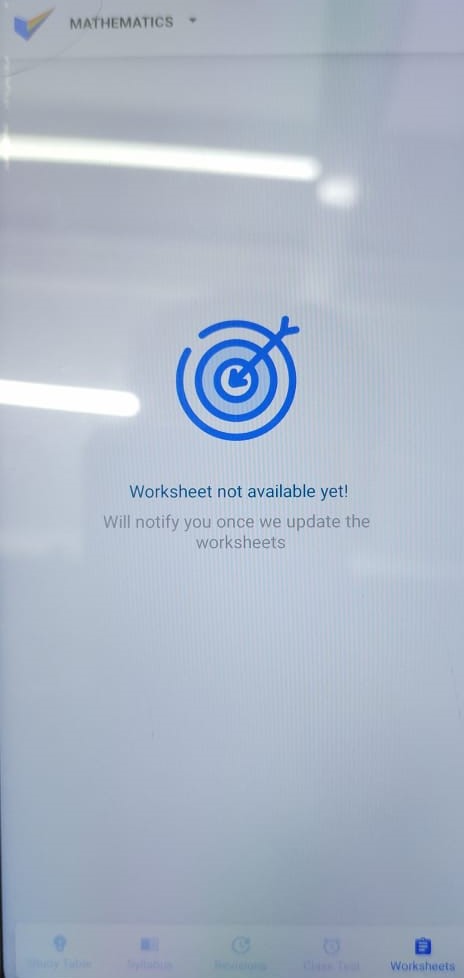
1. **DISPLAYING THE WORKSHEET**

This module is used to submit assignments given by teacher

In the teacher portal the teacher can assign the student with some assignments, these assignments are placed in the worksheet module each particular subject has a separate worksheet. The student can share the worksheet with other and the student can also print the worksheet

**REQUEST -** GET("/student/section/{section\_id}/subject/{subject\_id}/worksheets/")

**RESPONSE -** "statusCode": 200, "status": "SUCCESS", "message": "Work Sheets listed Successfully!"}



**Student Android App – CLASS TEST**

1. **DISPLAYING THE CLASS TEST**

**REQUEST -** @GET("/student/subject/{subject\_id}/class-tests/")

**RESPONSE -**

"completed": [ {"id": 80,"title": "MathTest1","**chapter**": {"id": 238,"title": "Introduction to Trigonometry", "index": 10,"revision\_status": 0}, "duration": 10,"no\_of\_questions": 10,"min\_correct\_ans": 8,"created\_by": {"id": 4,"full\_name": "Teacher anyth Test","profile\_pic": "https://sgp1.digitaloceanspaces.com/upsc-sgp/media-staging/users/4\_Teacher\_anyth\_Test/profile\_picture/1622603932402\_Upschool.jpg","slug": "Dnxboe"},

"**scheduled\_date\_time**": "2021-09-03 01:17:00","**expire\_at**": "2021-09-03T01:27:00Z","status": 3,"time\_taken": 0,"created\_at": "2021-09-03 01:15:51"

},}

**Student Android App – CLASS TEST**

1. **DISPLAYING THE CLASS TEST**

This displays the number of class test that is available in class test module

The class test is given by the teacher of particular subject each of the class test contains a timer within which the student has to complete the test. All the class test will be displayed in the class test section.

In this API request we will pass the subject id that is fetching the Class-test of that particular subject where every subject has a separate class test module.here the class test is displayed in list form

**REQUEST -** @GET("/student/subject/{subject\_id}/class-tests/")

**RESPONSE -**

"completed": [ {"id": 80,"title": "MathTest1","**chapter**": {"id": 238,"title": "Introduction to Trigonometry", "index": 10,"revision\_status": 0}, "duration": 10,"no\_of\_questions": 10,"min\_correct\_ans": 8,"created\_by": {"id": 4,"full\_name": "Teacher anyth Test","profile\_pic": "https://sgp1.digitaloceanspaces.com/upsc-sgp/media-staging/users/4\_Teacher\_anyth\_Test/profile\_picture/1622603932402\_Upschool.jpg","slug": "Dnxboe"},

"**scheduled\_date\_time**": "2021-09-03 01:17:00","**expire\_at**": "2021-09-03T01:27:00Z","status": 3,"time\_taken": 0,"created\_at": "2021-09-03 01:15:51"

},}

**Student Android App – CLASS TEST QUESTIONS**

1. **DISPLAYING THE CLASS TEST QUESTIONS**

This Api is used to display the question of the class test. this Api is a get request where it fetches the question from the backend. In the request we pass the class test id which is unique for every class test. this Api fetches all the questions that comes under that particular class test\_id and each question is displayed in one particular page

**REQUEST -** @ GET("/student/class-test/{class\_test\_id}/generate-questions/")

**RESPONSE -** "questions": [

{"id": 1513,

"**question**": Assertion (A): The gravitational force between the Earth and the Sun is lesser at aphelion than at perihelion <p> **Reason**: Aphelion is the point on the earth's orbit that is farthest from the sun, whereas, perihelion is the point on the earth's orbit when it is closest to the sun. Gravitational force is inversely dependant on the square of the distance. </p>",

"description": null,"number\_of\_answers": 1,"group": 1554,"answer\_label": null,"option\_type": 1,

"options": [ {"id": 987,

"value": <p>Both A and R are true and R is the correct explanation of A.</p>"},

{"id": 988, "value": " <p>Both A and R are true but R is not the correct explanation of A.</p>"}],

"attachments": [],"show\_math\_keyboard": false,"answer\_disclaimer": "","show\_disclaimer": false },

**Student Android App – CLASS TEST ANSWER**

1. **DISPLAYING THE CLASS TEST ANSWER-**

This Api is used to fetch the answer for all questions that is given in the class test, this Api is a get request where it fetches the answer for all the question from the backend. In the request we pass the class test id which is unique for every class test. this Api fetches all the answer for all the questions that comes under that particular class test\_id and each question is displayed in one particular page and the answer are displayed at the end, that is during the result

**REQUEST -** @ GET("/student/class-test/{class\_test\_id}/fetch-answers/")

**RESPONSE -** "answers": [{

"id": 13070,"status": 0,

"": " **question** <p>**Assertion** (A): A black hole is a region of space from which nothing can escape. Not even light can escape from it. So strong is the force of gravity of the black hole. However, we (Earth) dont seem to be pulled by it.

<p>**Reason** (R): The gravitational pull due to black holes is not experienced on earth because the distance between a black and earth is very less. The lesser the distance between the objects, the lesser will be the gravitational force;</p>",

"**description**": null,"option\_type": 1,"student\_answer": null,"**correct\_answer**": "984","answer\_label": null,"number\_of\_answers": 0,

"options": [

{"id": 982, "value": Both A and R are right and R provides a correct explanation for A. },

{ "id": 983, "value":<p>Both A and R are right but R does not provide a correct explanation for A."},

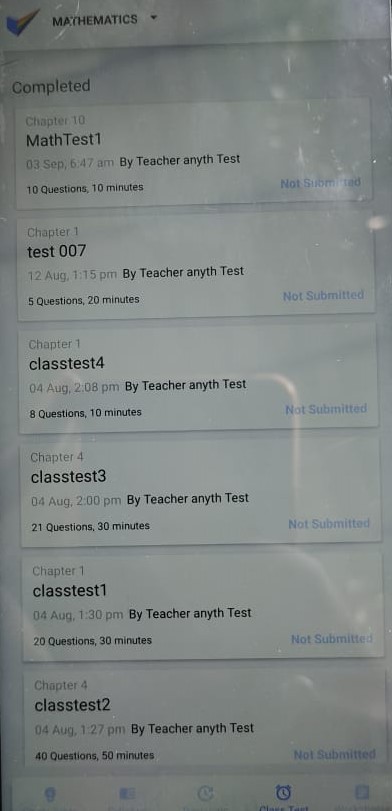
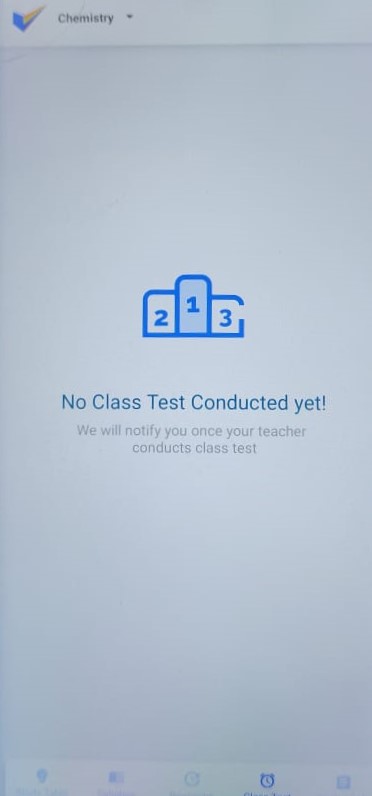
{ "id": 984, "value": A is true but R is false },

{ "id": 985, "value": A is false but R is true. },

{ "id": 986, "value": Both A and R are false.}

]

},



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

