IT632 – Software Engineering [Winter 2021-22]

Project Final Evaluation document

on

Question Paper Generating System

Members Details:

Student Name	Student ID
Vasu Pastagia (Team Leader)	202112003
Dhruvi Jariwala	202112009
Anilkumar Vaghari	202112010
Darshan Patil	202112013
Pragati Khurana	202112019
Arjun Solanki	202112038
Aditya Jain	202112075
Dhrumi Shah	202112083
Apoorv Jain	202112112
Sagar Variya	202112114

Guided By:

Professor: Dr. Saurabh Tiwari **Teaching Assistant:** Priyanka Mishra

Index

Introduction

- Question Paper generating system is a web based application that has a huge pool of questions from all domains taught in an institute.
- It facilitates the Admin of an educational institute to generate and distribute question paper for an exam in just few minutes.
- It covers wide portion coverage and no chance of paper leaks.
- There will be no need of transporting papers through police/ security vans to all colleges/ departments
- Our system efficiently excludes the human efforts and saves time and resources.

Overall Description

Scope

The main goal of Question paper generating system is to give flexible user interface to an educational institute for generating efficient question paper and distribute among colleges/ departments.

(1) Admin:

- As an admin, the scope of this project is to generate question paper either from 2 options:
    a. Automatic generation with difficulty level specified.
    b. Customized generation with difficulty level and number of questions for each weightage specified.
- Admin will email the generated question paper to concerned colleges/ departments on exam day.

(2) Faculty:

- As a faculty, the scope of this project is to first get verified by the admin after registration.
- Faculty can then login in the system using the credentials.
- (S)he can add a pool of questions with different difficulty levels and weightage with their respective correct answers.
- Questions can be of subjective or objective type.
- (S)he can edit, view and delete the questions.
- On a day of exam, (s)he can download the question paper generated by admin.

Users

- Admin – The one who generates papers and email them to colleges/ departments
- Institutes - The one who acquire the system to generate papers.
    - Faculties – The one who adds variety of questions and their answers for specific domain.

Stakeholders

- Admin – He is one of the user of the system
- Project Mentor – He guides and manages the project
- Project Leader – He takes care of the development of system
- Project Team – Developers team
- The Institutes and Faculties which uses the system to generate papers – They are the users of the system.

Possible Features

- Faculty is allowed to look at all questions generated in his/her domain by all the faculties.
- Faculty can like or save the question for requesting admin to generate question paper with the "Must haves".
- Faculty can surf on previous year question papers for giving sample to their students.
- Admin is given choice for selecting from various options of topics in generating question paper of a particular subject.
- Admin is allowed to generate multiple question papers for the same exam and choice is given to the institute for finalizing it.
- We can create a question paper with both objective and subjective questions.

Requirement elicitation technique

Name: Brainstorming is used to generate new ideas and find a solution for a specific issue.

What is expectation of System?

System is used to simplify the paper generating method

- According to Subjects
- According to Marks
- According to Difficulty Level
- According Faculty Requirement (Automatic Generation / Customize Generation)

Risk Factors in Systems are:

- Questions can be redundant for different papers.
- The requirement of number of questions can be too high to generate the question paper according to faculty requirements.
- Different Faculty can add Questions for same subject so there may be question redundancy.

Rules to Follow:

- Faculty must enter legitimate information.
- Admin only allows authentic faculties.
- Faculty must be verified by admin in order to access the application.
- Faculty must add unique Questions in their Question part.

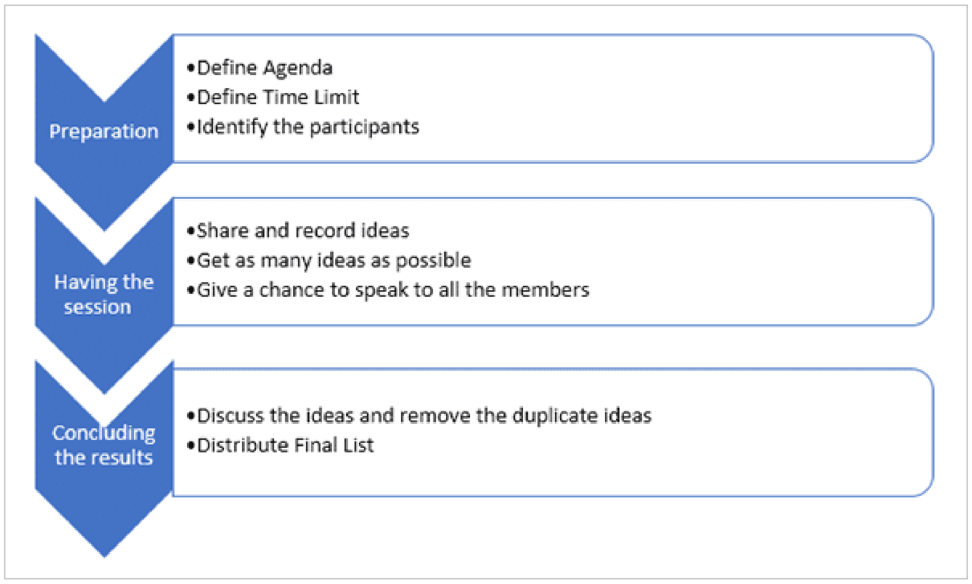
- Faculty must provide correct answers to the questions in subjective and objective manner.

Options Available to Resolve the Current Issues:

- Currently, faculties are getting trouble in generating the paper according to difficulty wise and according to marks.
- So our system provides two types of functionalities:

  (1) Automatic Generation: In this functionality, admin will give the difficulty level for particular subject and according to that System will take the questions from question bank and generate papers.

  (2) Customize Generation: In this functionality, admin will provide the difficulty level and then provide numbers of questions for each marks and after matching that requirements system will generate the question with the provided number of questions and provided difficulty level.


Preparation
•Define Agenda
•Define Time Limit
•Identify the participants
Having the session
•Share and record ideas
•Get as many ideas as possible
•Give a chance to speak to all the members
Concluding the results
•Discuss the ideas and remove the duplicate ideas
•Distribute Final List

Process Model

Name: Incremental Model

Reasons for choosing this process model:

(1) Major requirements of the system were clearly defined, however, some details evolved over time.
(2) A new technology has been used.
(3) There was a need to deliver the system early.
(4) This model best suited our team because we are still in learning phase
(5) If any changes occur at certain phase, we can rebuild as this process model is iterative, we can go back to certain phase

Functional Requirements

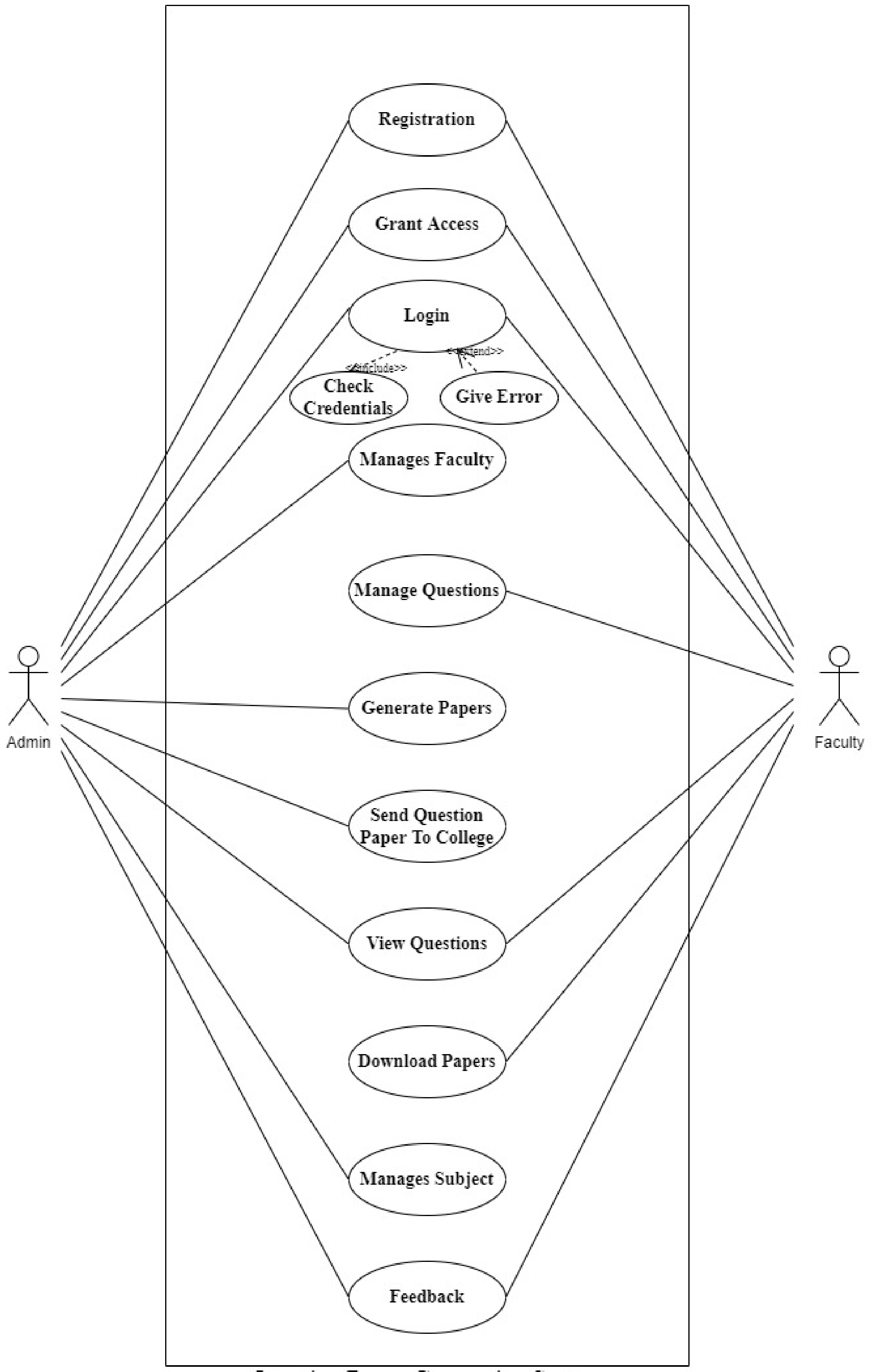
1. Admin:

   - Admin can grant access to faculty.
   - Admin can remove faculty.
   - Admin can view all questions.
   - Generate question papers with two options (Auto-generated and customize with difficulties).
   - Admin can send generated question papers to multiple colleges by adding their emails.

2. Faculty:

   - Faculty can request access to this system by registering him/herself.
   - After approval, faculty can log in.
   - Faculty can add questions and can update that questions and also delete that questions.
   - Download generated question papers (Previous years) with or without answers key.

Use Case Diagram



Question Paper Generating System

Use Case Description

Use Case Name	**Registration**
Actors	Admin, Faculty
Entry Condition	The Admin activate the ‘Registration’ function for Faculties
Flow of Events	▪ System responds by presenting a form with different details to filled-in ▪ Faculty fills the form and submit the form. ▪ The admin review the information provided by Faculty and creates a user in DB by invoking Registration usecase. ▪ The admin selects a response and allocates resources to the User.
Exit Condition	The admin allocate resource and redirect to the login page

Use Case Name	**Grant Access**
Actors	Admin, Faculty
Entry Condition	The Admin activate the ‘Grant Access’ function for Faculties
Flow of Events	System responds by presenting a page with different details and buttons to accept and reject the request of faculties If admin approves then faculty will get the mail of account activated and he/she can login If admin reject then faculty will get the mail of rejection
Exit Condition	The admin send mail of activation and rejection

Use Case Name	**Login**
Actors	Admin, Faculty
Entry Condition	The Admin activate the 'Login' function for Faculties
Flow of Events	System responds by presenting a form with different details to filled-in Faculty completes the form and submit the form. The admin check the credentials provided by Faculty and allow user to login into the system. If given credentials are wrong then faculty can't login
Exit Condition	The admin send allows the Faculty to access functionalities.

Use Case Name	**Manage Faculty**
Actors	Admin
Entry Condition	The Admin activate the 'Manage Faculty' function to accept and reject faculty requests
Flow of Events	System responds by presenting details of faculty with the buttons of activate and deactivate Admin will deactivate the active faculty and vice versa
Exit Condition	The admin will activate and deactivate the faculty

Use case name	**Generate Paper**
Actors	Admin
Entry condition	1. The Admin activates the "**generatePaper**" function as per institutes requirements
Flow of event	2. System responds by presenting two options 'Auto generate' and 'Custom generate' to choose one 3. The Admin choose one of the given options 4. System responds by presenting a form with different details to filled-in 5. The Admin fill the form with required details 6. System generate a question paper as per provided details by the Admin

Exit Condition	7. Generated question papers stored into database

Use case name	**Send Papers To College**
Actors	Admin
Entry condition	1. After generating question paper by **Generate Paper** use case, the Admin activates the **“sendPapersToCollege”** function for send to multiple institutes
Flow of event	2. System responds by presenting different options for sending paper 3. The Admin click on send button to send question paper via email
Exit Condition	4. The Admin receives the acknowledgement mail is sent to the institutes

Use case name	**View Questions**
Actors	Admin, Faculty
Entry condition	1. The Admin & Faculty activates the **“View Questions”** function to view questions and their answers
Flow of event	2. System responds by fetching questions and answers from the database then display on the screen
Exit Condition	3. The Admin & Faculty show questions and answers

Use Case Name	**Manage Questions**
Actors	Admin
Entry Condition	The Admin activate the ‘Manage Questions’ function to insert, update, delete and view the questions
Flow of Events	System responds by presenting details of faculty with the buttons of activate, deactivate, update, and insert. By clicking on activate button, faculty activate the deactivated questions and vice versa.

	By clicking on insert, the form will be displayed to enter the details. By clicking on update, the form will be displayed with the details of questions and then admin will change the details of question.
Exit Condition	The admin will activate, deactivate, insert, view the faculty

Use case name	**Download Papers**
Actors	Faculty
Entry condition	1. The Faculty activates the **“Download Papers”** function to download paper in PDF format and save into his/her device
Flow of event	2. System responds by presenting different option to download and print a PDF 3. The Faculty click on Print button to print PDF of question paper
Exit Condition	3. The Faculty receives the acknowledgement about PDF is downloaded

Use case name	**Manage Subject**
Actors	Admin
Entry condition	1. The Admin activates the **“Manage Subject”** function to add, update, delete and view subjects
Flow of event	2. System responds by presenting options ‘add’, ‘update’, ‘delete’ and view 3. The Admin click on any one of the option 4. System responds by presenting a form with different details to filled-in 5. The Admin fill the form and submit to perform add and update operation
Exit Condition	3. The Admin receives the acknowledgement about subject is added, updated or deleted from the database

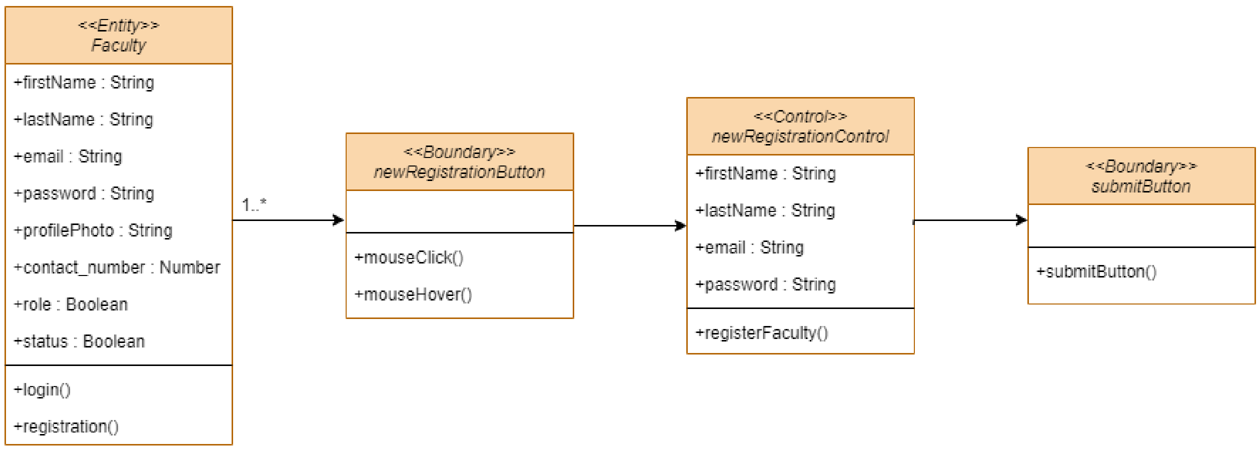
Use case name	**Feedback**
Actors	Admin, Faculty
Entry condition	1. The Faculty activates the **“Feedback”** function to give feedback about working system
Flow of event	2. System responds by presenting a form with different details to filled-in 3. The Faculty fill the form and submit it. All the filled data stored into database 4. The Admin view feedback given by Faculty
Exit Condition	5. The Admin gives the acknowledgement about feedback

Non Functional Requirements

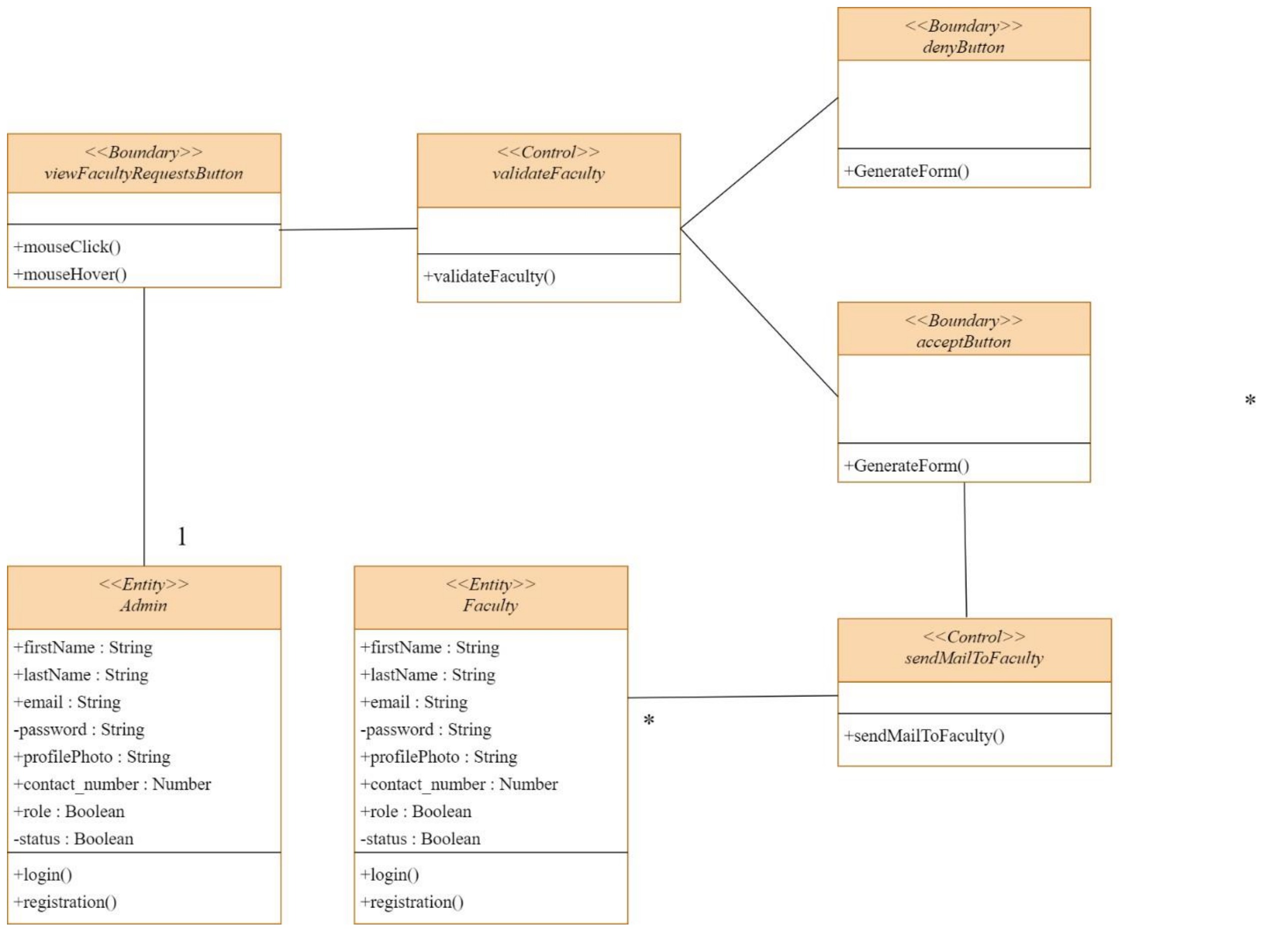
- The system should remain **accessible** 24/7.
- System should be **easy to use** for admin and faculty.
- System should be **accurate** and **reliable**.
- The system should be able to work on all **available** browsers and all **versions** of browsers.
- Good interface is needed.
- The system should able to handle **large data** because there are so many questions of different subjects.
- The system should be **secure** so the question paper is generated by admin only and can be viewed by faculty after exam date and time.

Analysis Class Diagrams

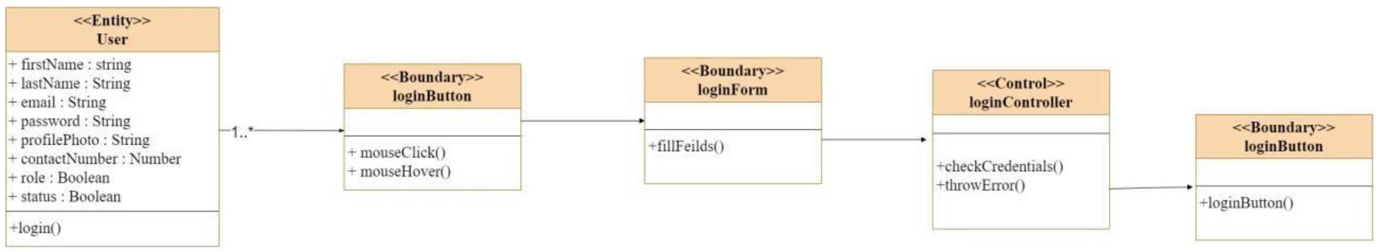
1. Registration



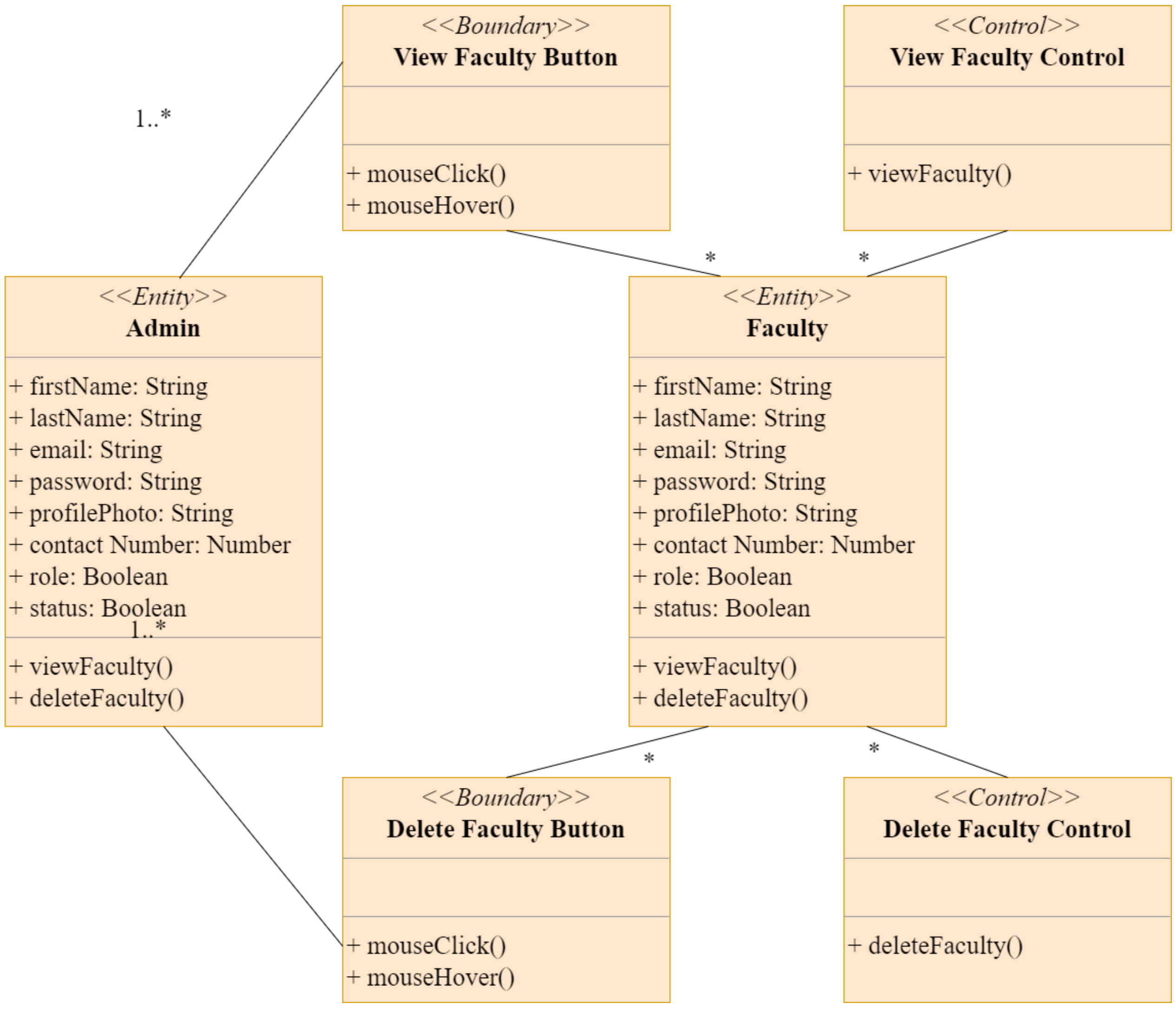
2. Grant access



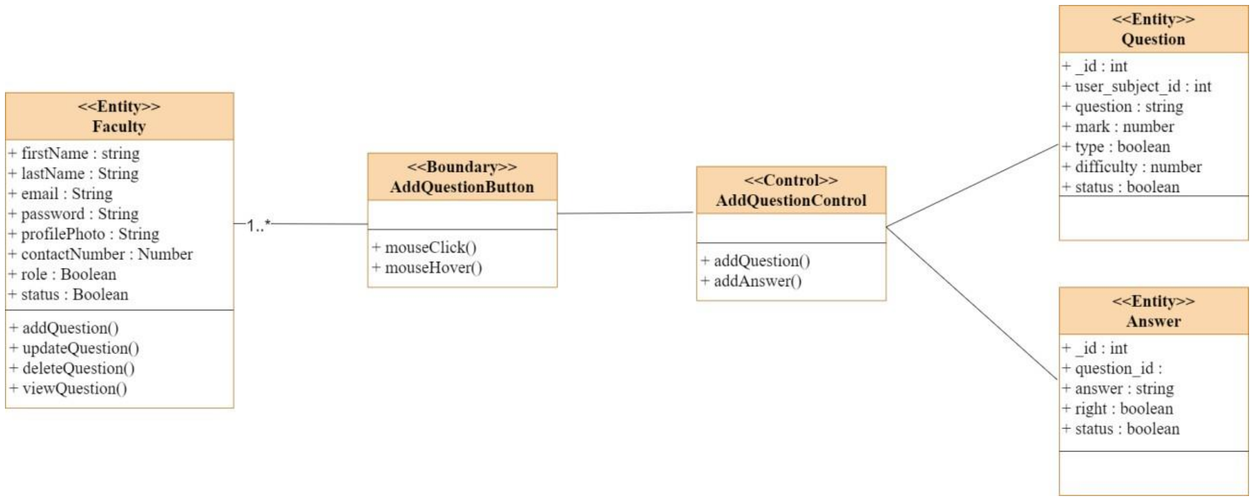
3. Login



4. Manage Faculty

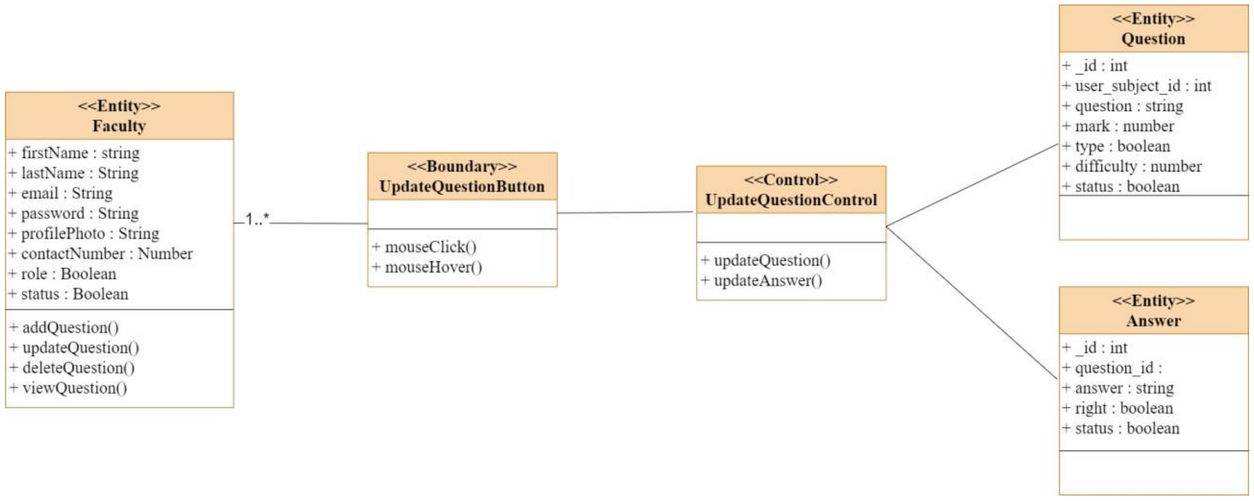


5. Manage Questions
    - Add



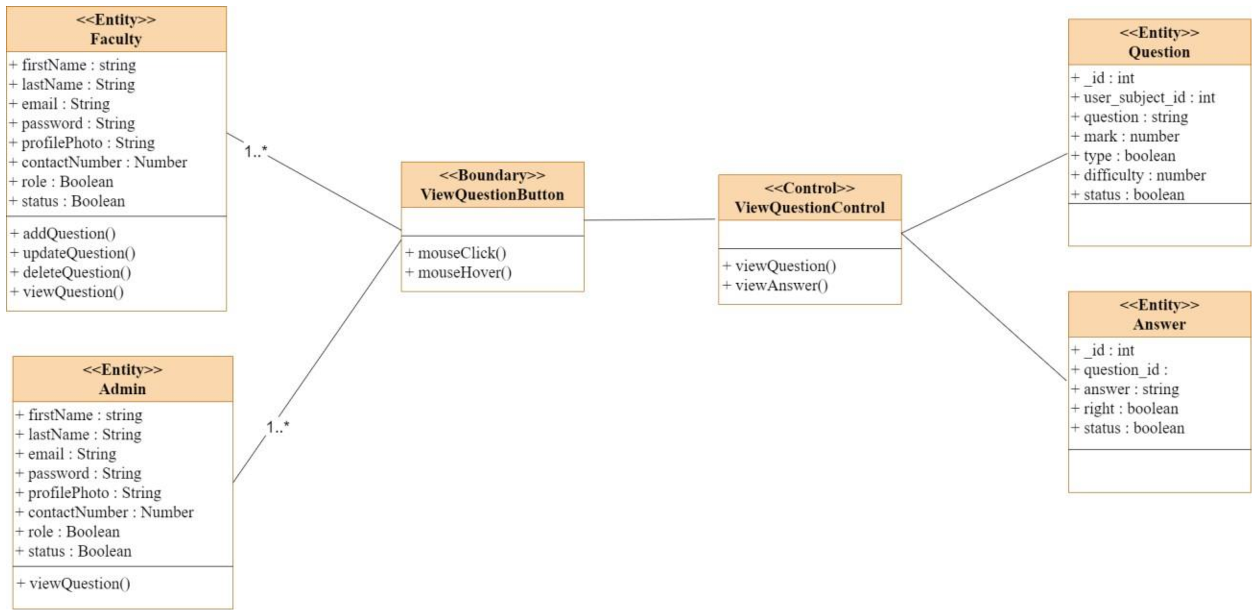
<<Entity>>
Faculty
+ firstName : string
+ lastName : String
+ email : String
+ password : String
+ profilePhoto : String
+ contactNumber : Number
+ role : Boolean
+ status : Boolean
+ addQuestion()
+ updateQuestion()
+ deleteQuestion()
+ viewQuestion()
1..*
<<Boundary>>
AddQuestionButton
+ mouseClick()
+ mouseHover()
<<Control>>
AddQuestionControl
+ addQuestion()
+ addAnswer()
<<Entity>>
Question
+ _id : int
+ user_subject_id : int
+ question : string
+ mark : number
+ type : boolean
+ difficulty : number
+ status : boolean
<<Entity>>
Answer
+ _id : int
+ question_id :
+ answer : string
+ right : boolean
+ status : boolean

- Update

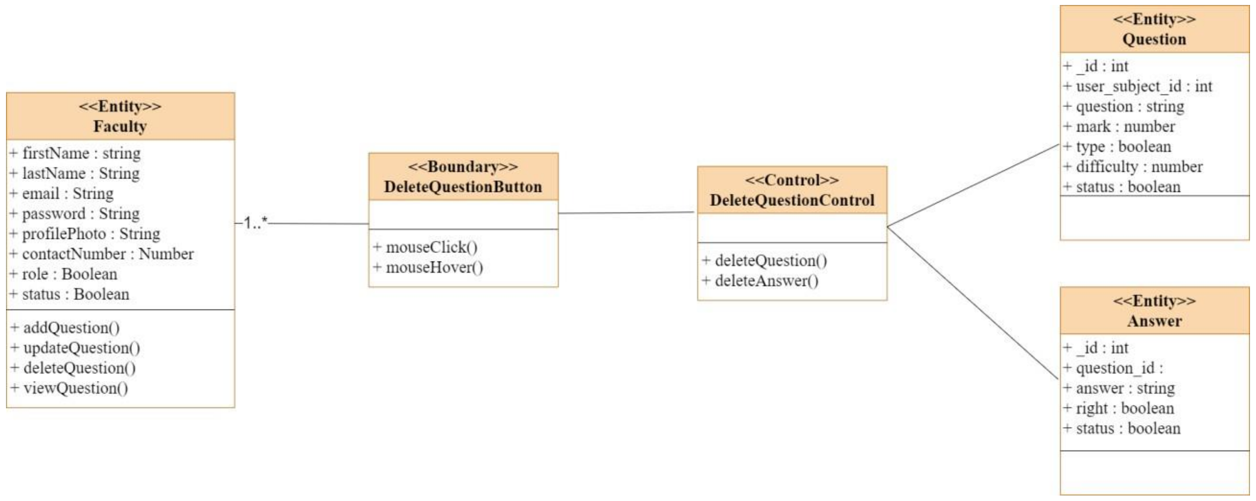


<<Entity>>
Faculty
+ firstName : string
+ lastName : String
+ email : String
+ password : String
+ profilePhoto : String
+ contactNumber : Number
+ role : Boolean
+ status : Boolean
+ addQuestion()
+ updateQuestion()
+ deleteQuestion()
+ viewQuestion()
1..*
<<Boundary>>
UpdateQuestionButton
+ mouseClick()
+ mouseHover()
<<Control>>
UpdateQuestionControl
+ updateQuestion()
+ updateAnswer()
<<Entity>>
Question
+ _id : int
+ user_subject_id : int
+ question : string
+ mark : number
+ type : boolean
+ difficulty : number
+ status : boolean
<<Entity>>
Answer
+ _id : int
+ question_id :
+ answer : string
+ right : boolean
+ status : boolean

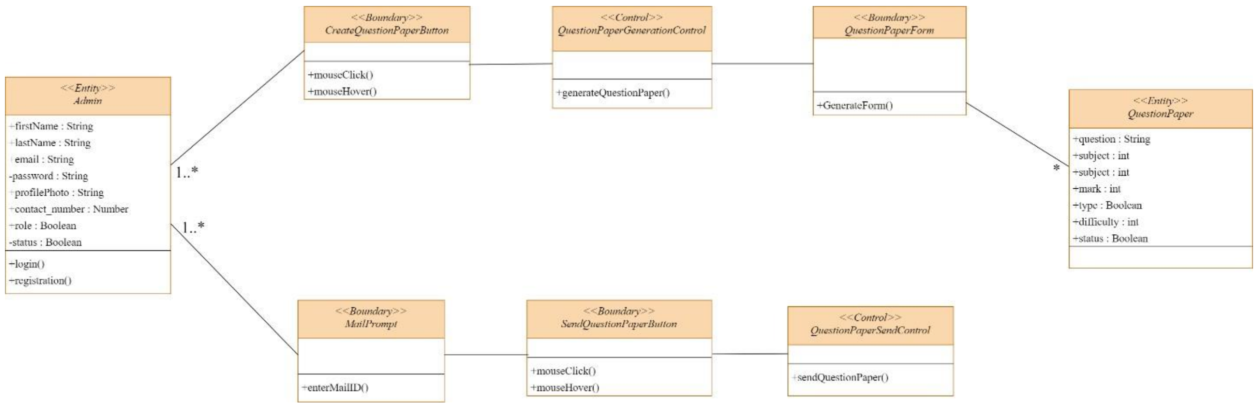
- View



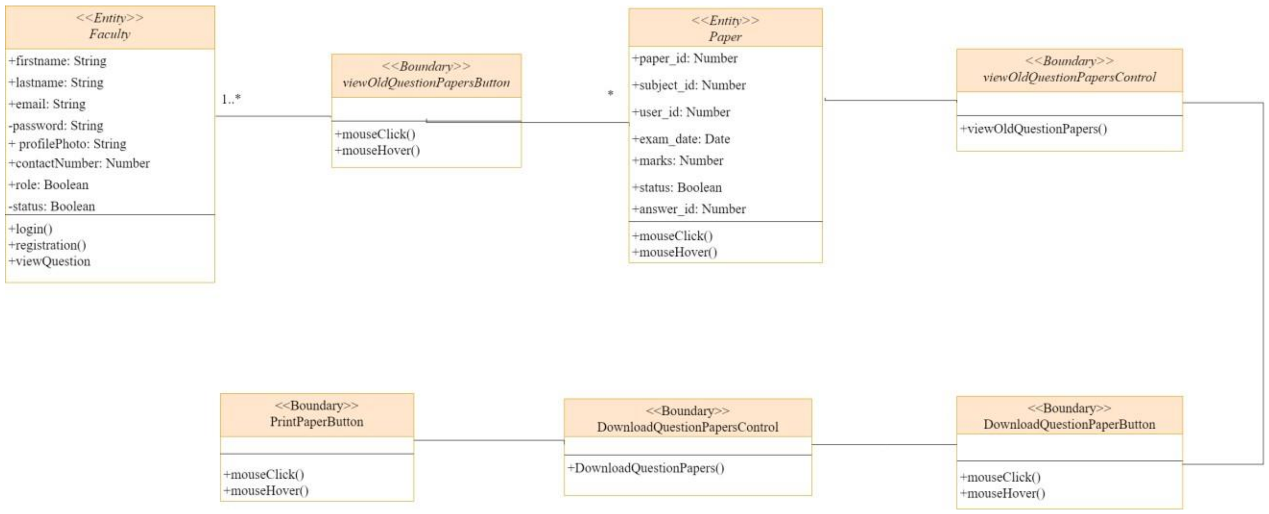
- Delete



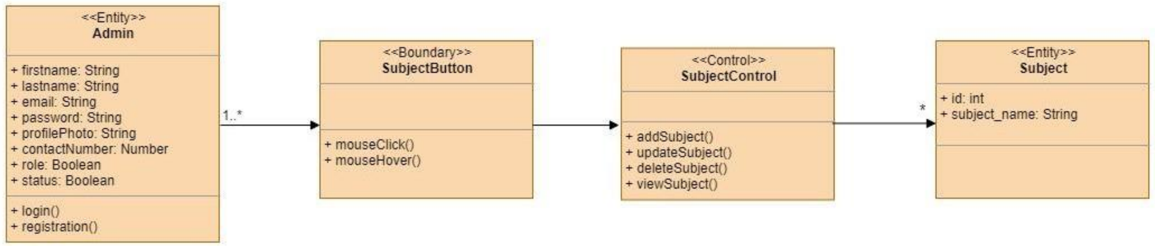
6. Generate paper & send papers to college



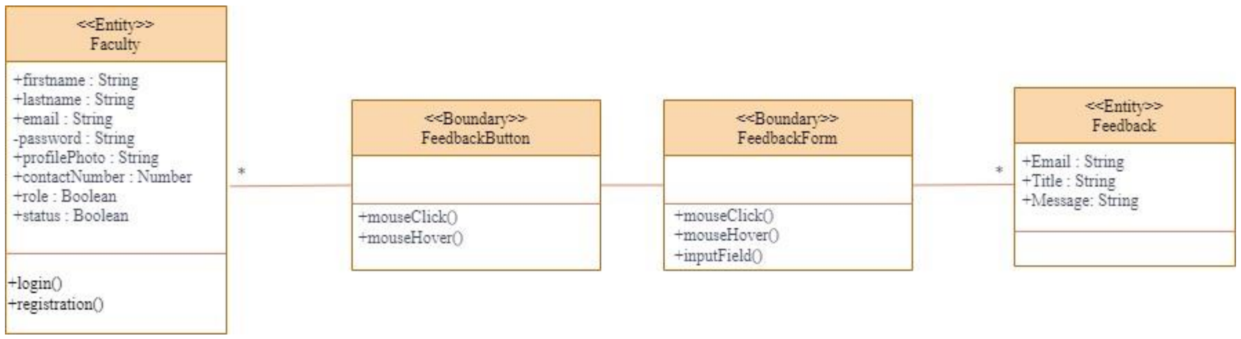
7. Download papers



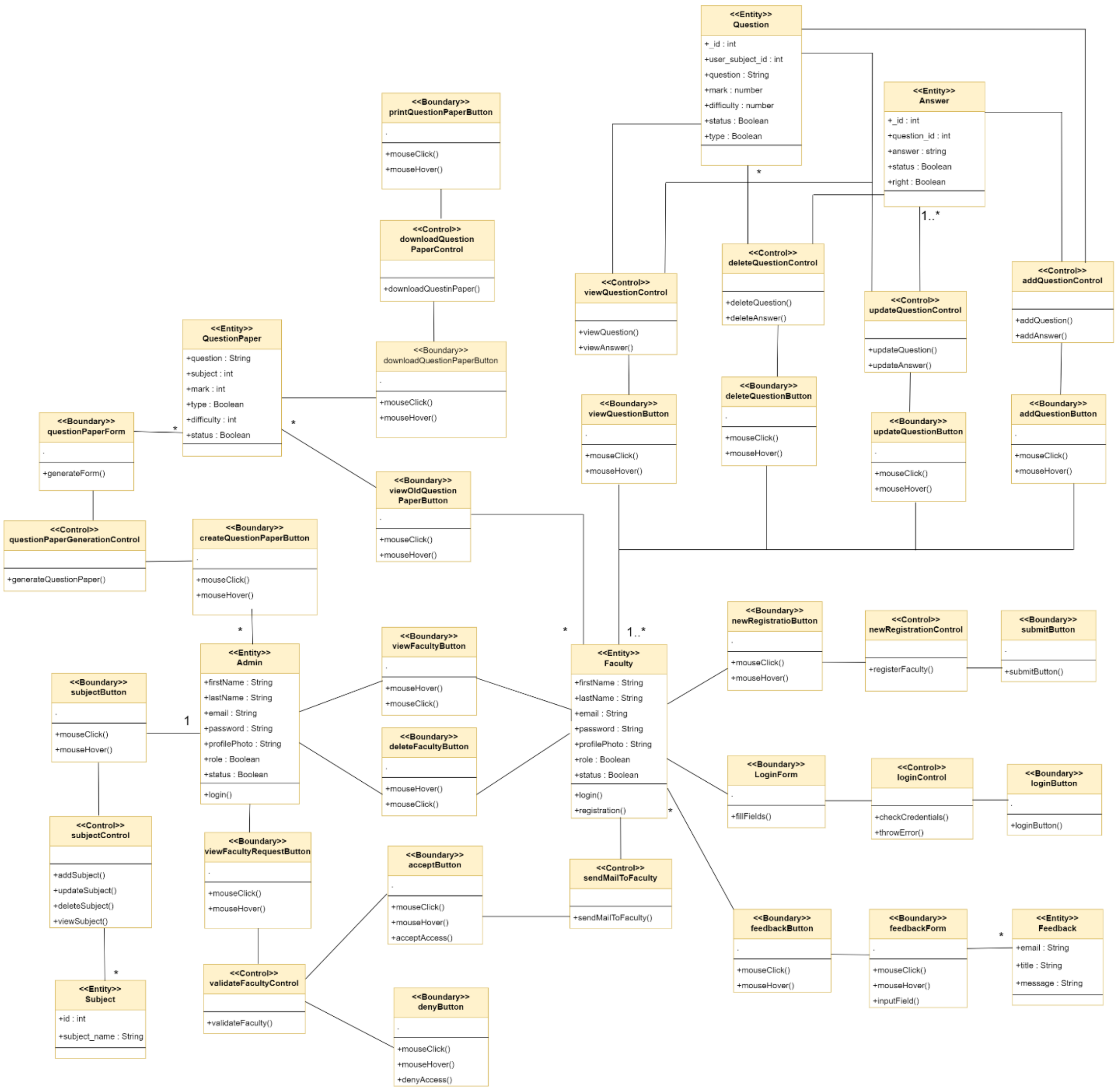
8. Manage Subject



9. Feedback

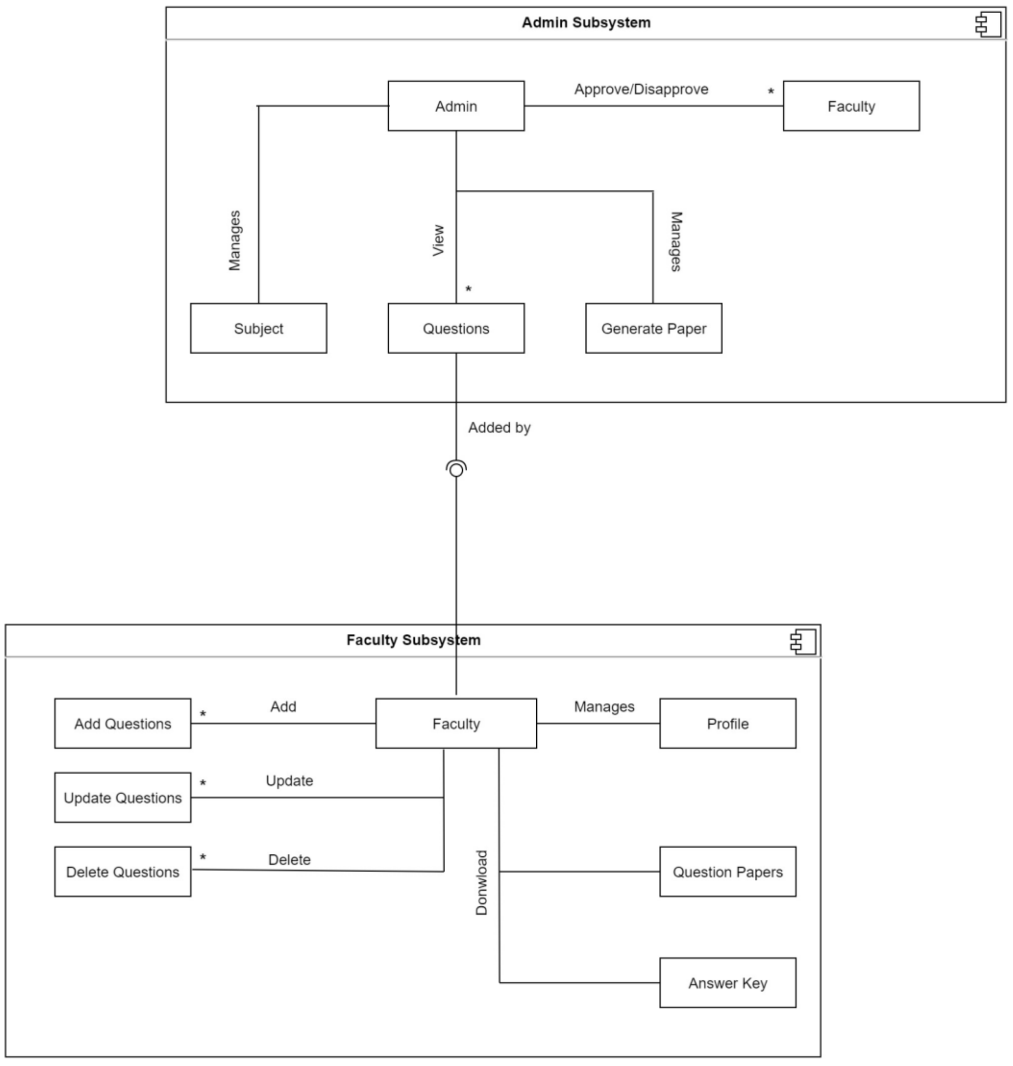


Complete Analysis Class Diagrams

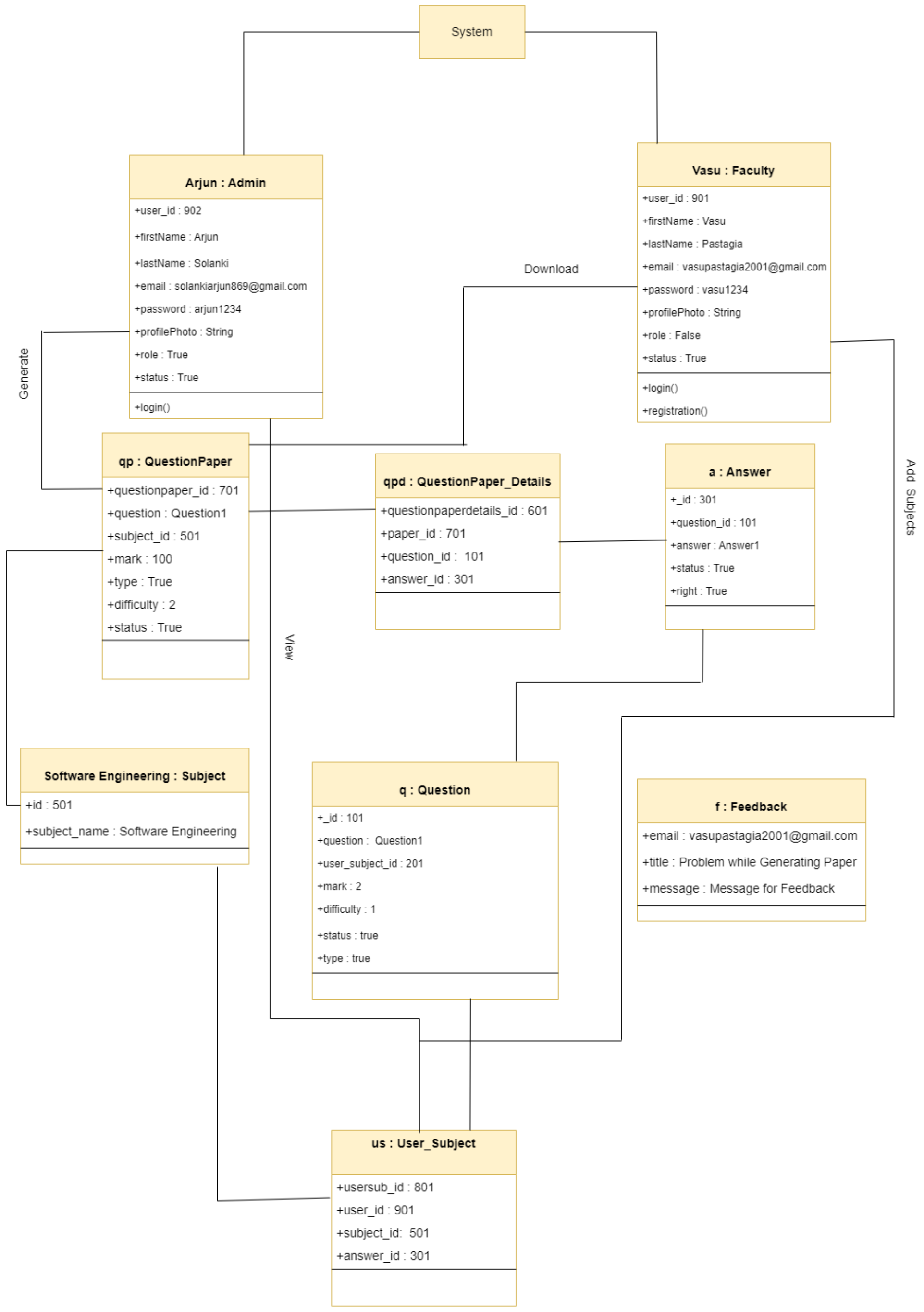

<<Entity>> Question
+_id : int
+user_subject_id : int
+question : String
+mark : number
+difficulty : number
+status : Boolean
+type : Boolean
<<Entity>> Answer
+_id : int
+question_id : int
+answer : string
+status : Boolean
+right : Boolean
1..*
*
<<Boundary>> printQuestionPaperButton
+mouseClick()
+mouseHover()
<<Control>> downloadQuestionPaperControl
+downloadQuestinPaper()
<<Boundary>> downloadQuestionPaperButton
+mouseClick()
+mouseHover()
<<Control>> viewQuestionControl
+viewQuestion()
+viewAnswer()
<<Control>> deleteQuestionControl
+deleteQuestion()
+deleteAnswer()
<<Control>> updateQuestionControl
+updateQuestion()
+updateAnswer()
<<Control>> addQuestionControl
+addQuestion()
+addAnswer()
<<Boundary>> viewQuestionButton
+mouseClick()
+mouseHover()
<<Boundary>> deleteQuestionButton
+mouseClick()
+mouseHover()
<<Boundary>> updateQuestionButton
+mouseClick()
+mouseHover()
<<Boundary>> addQuestionButton
+mouseClick()
+mouseHover()
<<Entity>> QuestionPaper
+question : String
+subject : int
+mark : int
+type : Boolean
+difficulty : int
+status : Boolean
<<Boundary>> questionPaperForm
+generateForm()
<<Boundary>> viewOldQuestionPaperButton
+mouseClick()
+mouseHover()
<<Control>> questionPaperGenerationControl
+generateQuestionPaper()
<<Boundary>> createQuestionPaperButton
+mouseClick()
+mouseHover()
<<Entity>> Admin
+firstName : String
+lastName : String
+email : String
+password : String
+profilePhoto : String
+role : Boolean
+status : Boolean
+login()
<<Boundary>> viewFacultyButton
+mouseHover()
+mouseClick()
<<Boundary>> deleteFacultyButton
+mouseHover()
+mouseClick()
<<Entity>> Faculty
+firstName : String
+lastName : String
+email : String
+password : String
+profilePhoto : String
+role : Boolean
+status : Boolean
+login()
+registration()
1..*
<<Boundary>> newRegistratioButton
+mouseClick()
+mouseHover()
<<Control>> newRegistrationControl
+registerFaculty()
<<Boundary>> submitButton
+submitButton()
<<Boundary>> LoginForm
+fillFields()
<<Control>> loginControl
+checkCredentials()
+throwError()
<<Boundary>> loginButton
+loginButton()
<<Boundary>> subjectButton
+mouseClick()
+mouseHover()
1
<<Control>> subjectControl
+addSubject()
+updateSubject()
+deleteSubject()
+viewSubject()
<<Entity>> Subject
+id : int
+subject_name : String
<<Boundary>> viewFacultyRequestButton
+mouseClick()
+mouseHover()
<<Control>> validateFacultyControl
+validateFaculty()
<<Boundary>> acceptButton
+mouseClick()
+mouseHover()
+acceptAccess()
<<Boundary>> denyButton
+mouseClick()
+mouseHover()
+denyAccess()
<<Control>> sendMailToFaculty
+sendMailToFaculty()
<<Boundary>> feedbackButton
+mouseClick()
+mouseHover()
<<Boundary>> feedbackForm
+mouseClick()
+mouseHover()
+inputField()
<<Entity>> Feedback
+email : String
+title : String
+message : String

System Design

Sub System Design



Object Design



Testing Plan and Strategies

Objective

Objective of our testing of this question paper generating web application is to succeed in all the possible test cases and provide a complete working system.

In our testing plan we are going to test features of our web application which are given below.

- Register
- Login
- Manage Subject
- Manage Questions
- Generate paper
- Feedback

In our testing plan we are not going to test features of our web application which are given below.

- View Questions
- Grant Access
- Download papers
- Logout

Testing Approach

Our testing approach is UI testing, unit Testing, integration Testing, System Testing.

UI Testing

UI is testing method that is used for testing the visual elements to verify that they are functioning according to requirements

Unit Testing (Manual)

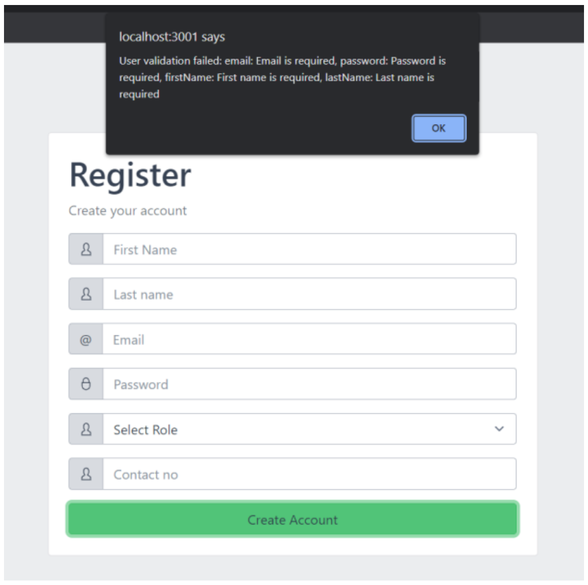
Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for process operation.

Based on the test cases we tested module by module testing.

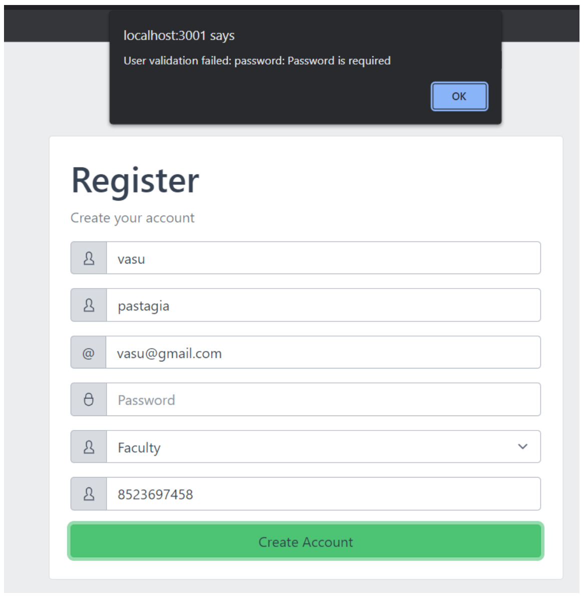
- Register

Sr. no	**Feature**	**Details**	**Output came**	**Output expected**	**expected**
01	Register	All fields will be empty	invalid	invalid	Empty fields are not allowed
02	Register	Email must have @ and .com	invalid	invalid	Email cant be without @ and suffix .com
03	Register	Phone number less than or more than 10 digits	invalid	invalid	Every number has only 10 digits and starts with 9,8,6

04	Register	All fields filled	valid	valid	If everything is perfect register user.


localhost:3001 says

User validation failed: email: Email is required, password: Password is required, firstName: First name is required, lastName: Last name is required


localhost:3001 says

User validation failed: password: Password is required

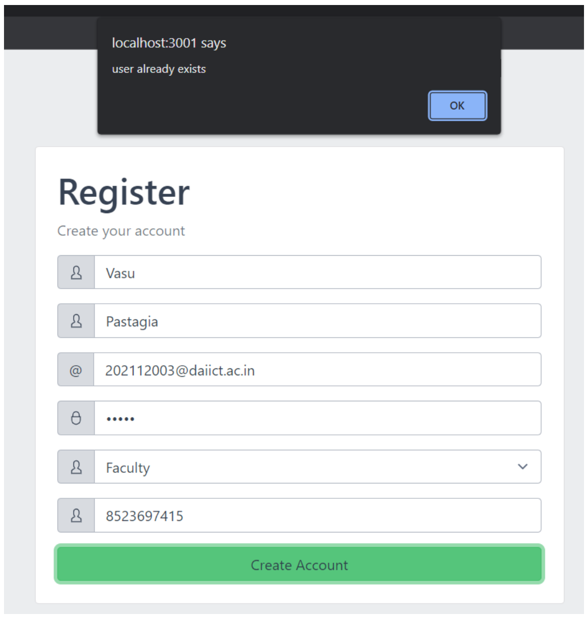
vasu

pastagia

vasu@gmail.com

Faculty

8523697458


localhost:3001 says

user already exists

Vasu

Pastagia

202112003@daiict.ac.in

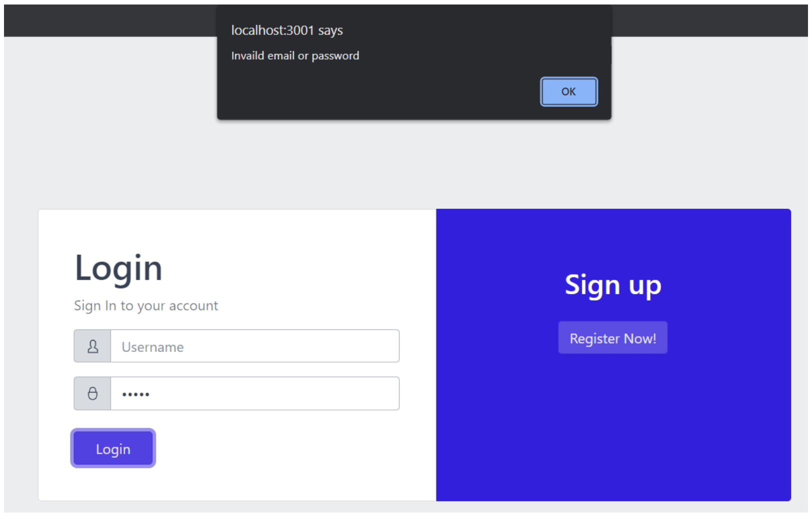
•••••

Faculty

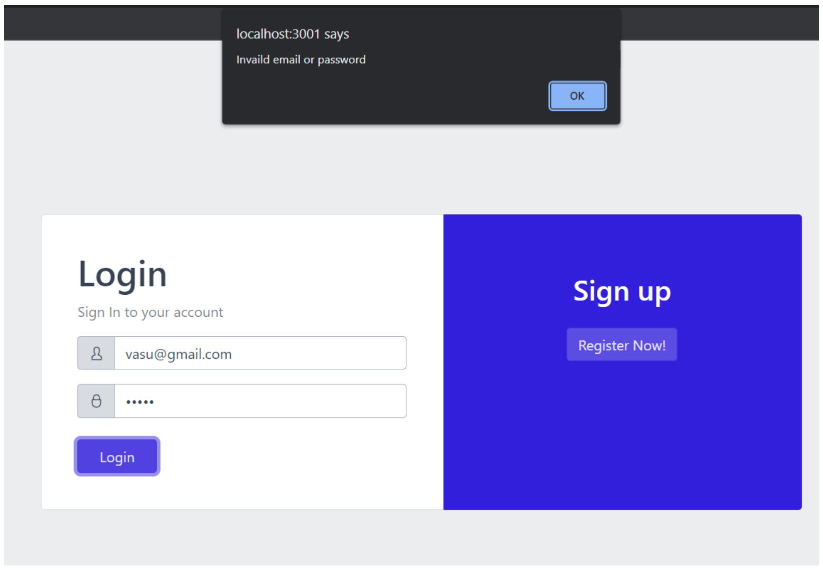
8523697415

- Login

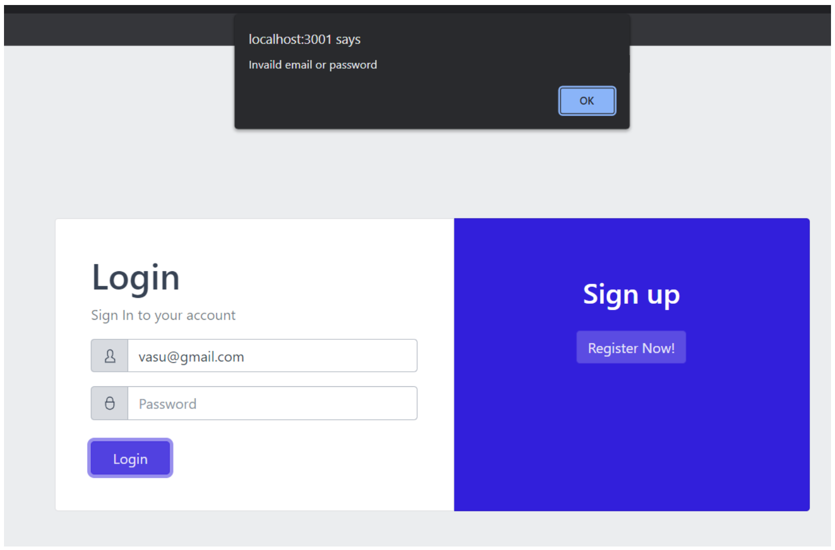
Sr. No	Feature	Details	Output came	Output expected	Expected
01	Login	Id passwords field must not be empty	invalid	invalid	Empty fields are not allowed
02	Login	Id passwords must match with database	invalid	invalid	Data stored must be similar to data entered
03	Login	User id must be present in database as user	Valid	Valid	Admin will give access to user then he/she will be considered as author


localhost:3001 says

Invaild email or password


localhost:3001 says

Invaild email or password

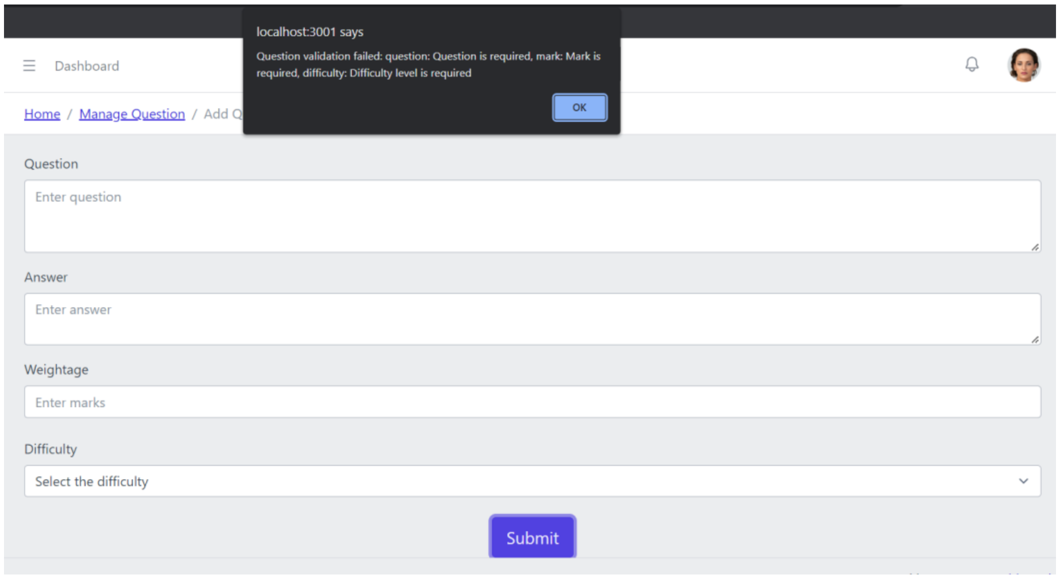

localhost:3001 says

Invaild email or password

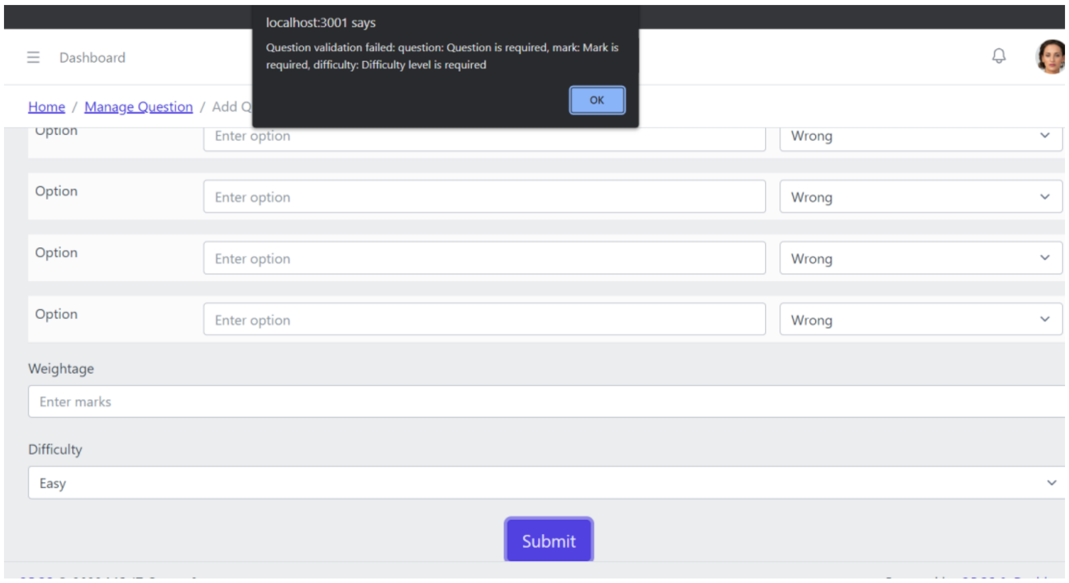
- Add question

Sr. No	**Feature**	**Details**	**Output came**	**Output expected**	**Expected**
01	Add Subjective Question	If any of the field is empty	invalid	Invalid	Empty fields are not allowed

02	Add Objective Question	If any of the field is empty	invalid	Invalid	Empty fields are not allowed


localhost:3001 says

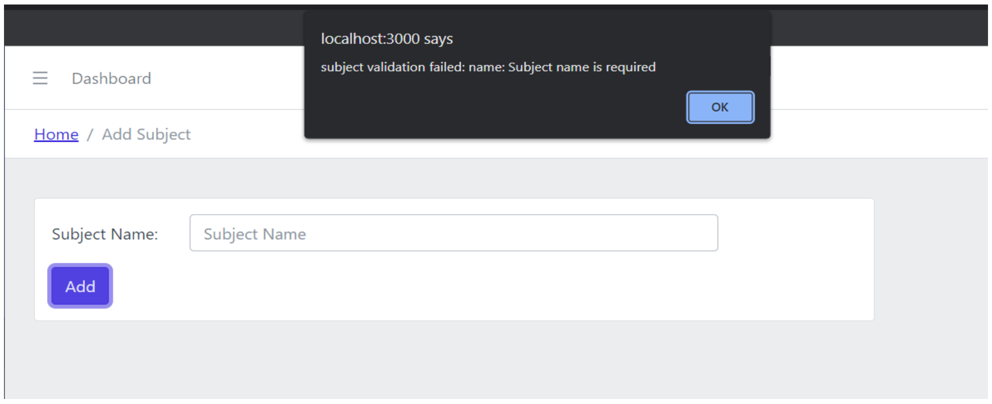
Question validation failed: question: Question is required, mark: Mark is required, difficulty: Difficulty level is required


localhost:3001 says

Question validation failed: question: Question is required, mark: Mark is required, difficulty: Difficulty level is required

- Add subject

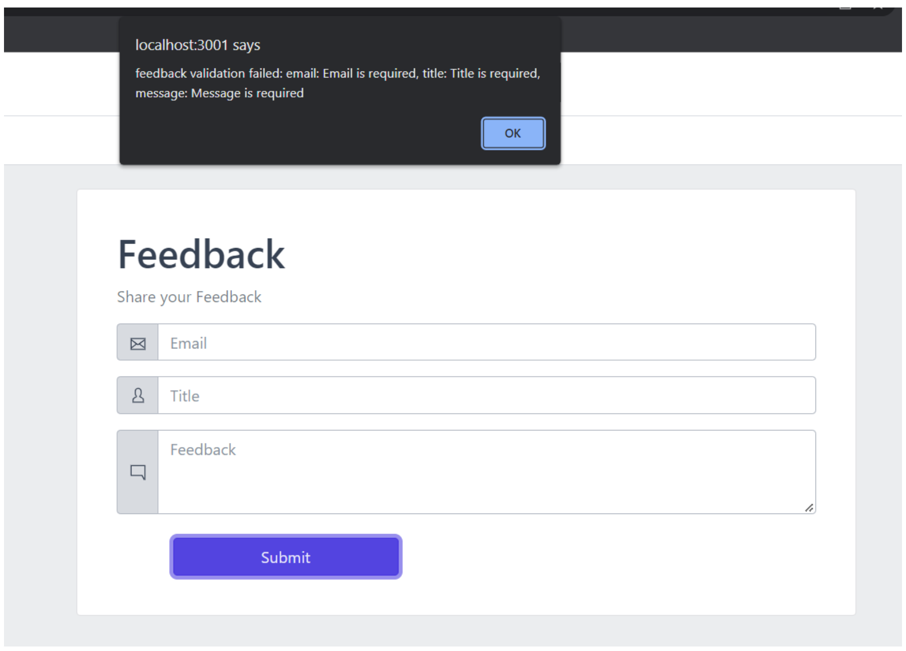
01	Subject required Field	Field will be left empty	invalid	Invalid	Empty field is not allowed
02	Subject required Field	Field filled with subject name	Valid	Valid	Subject will be added.


localhost:3000 says

subject validation failed: name: Subject name is required

- Feedback(faculty)

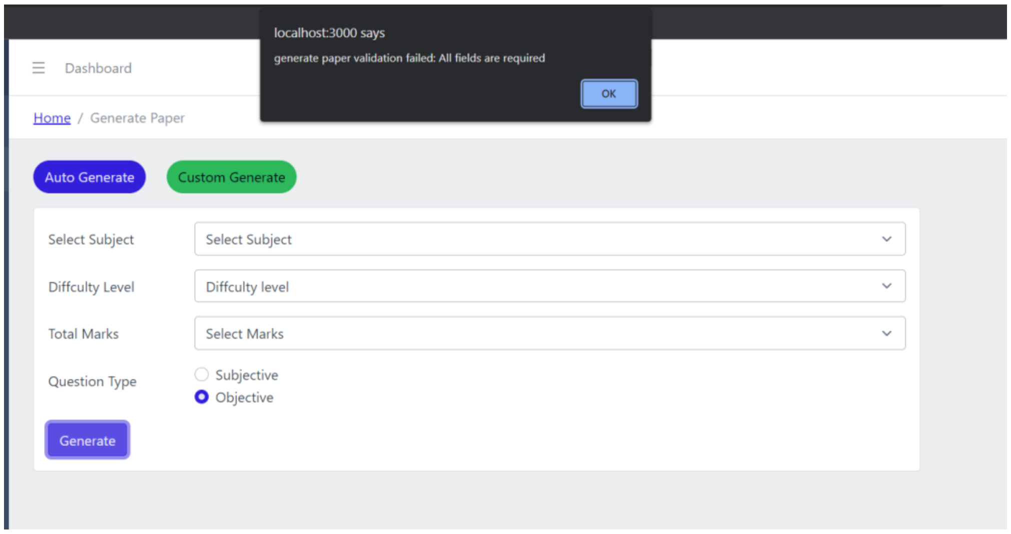
Sr no	**Feature**	**Details**	**Output Came**	**Output Expected**	**Expected**
01	Faculty Feedback	All Field will be left empty.	invalid	Invalid	Empty field is not allowed
04	Faculty Feedback	Test with registered Email id	Valid	valid	Feedback will be submitted


localhost:3001 says

feedback validation failed: email: Email is required, title: Title is required, message: Message is required

- Automatic Paper generation

Sr. No	Feature	Details	Output came	Output expected	Expected
01	Generate automatic Paper	If any field is empty	invalid	invalid	Empty fields are not allowed

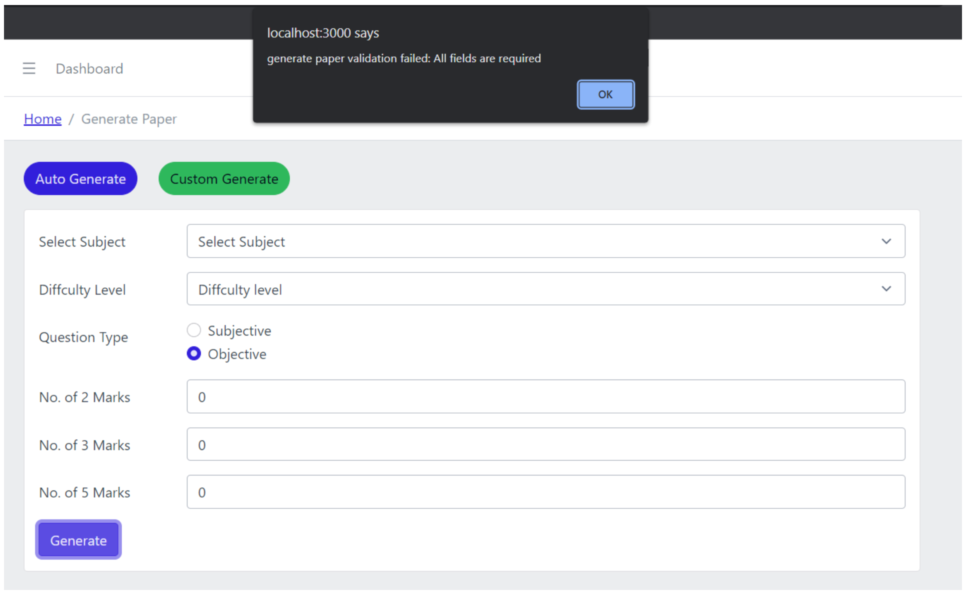

localhost:3000 says

generate paper validation failed: All fields are required

Select Subject: Select Subject
Diffculty Level: Diffculty level
Total Marks: Select Marks
Question Type: Subjective [✓ Objective]

- Customize Paper generation

Sr. No	**Feature**	**Details**	**Output came**	**Output expected**	**Expected**
01	Generate customize Paper	If any field is empty	invalid	invalid	Empty fields are not allowed
02	Generate customize Paper	No. of questions more than 10	Invalid	Invalid	It should be less than or equal to 10



localhost:3000 says

generate paper validation failed: All fields are required

Select Subject: Select Subject

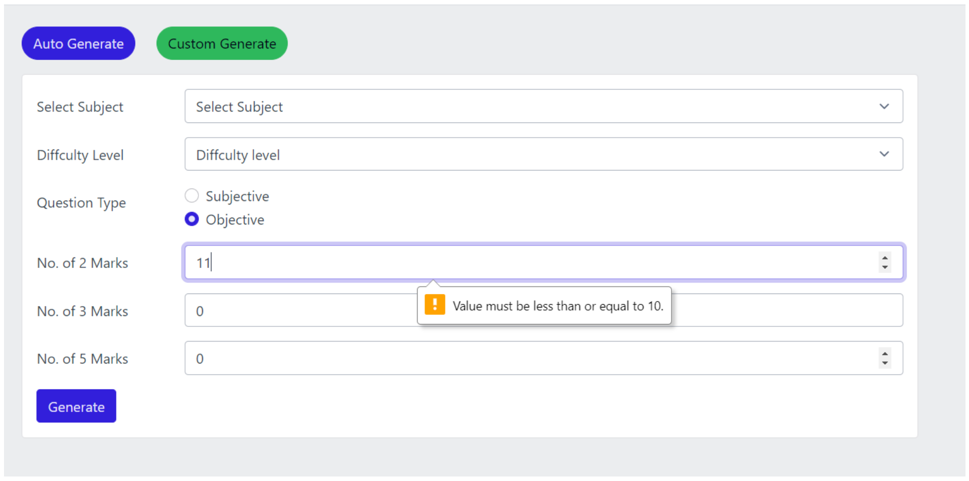
Diffculty Level: Diffculty level

Question Type: Subjective [✓ Objective]

No. of 2 Marks: 0

No. of 3 Marks: 0

No. of 5 Marks: 0



Select Subject: Select Subject

Diffculty Level: Diffculty level

Question Type: Subjective [✓ Objective]

No. of 2 Marks: 11

Value must be less than or equal to 10.

No. of 3 Marks: 0

No. of 5 Marks: 0

Integration testing:

Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

We have combined some modules for testing.

- Verified and login
- Add questions and Edit questions
- Add subject and delete subject

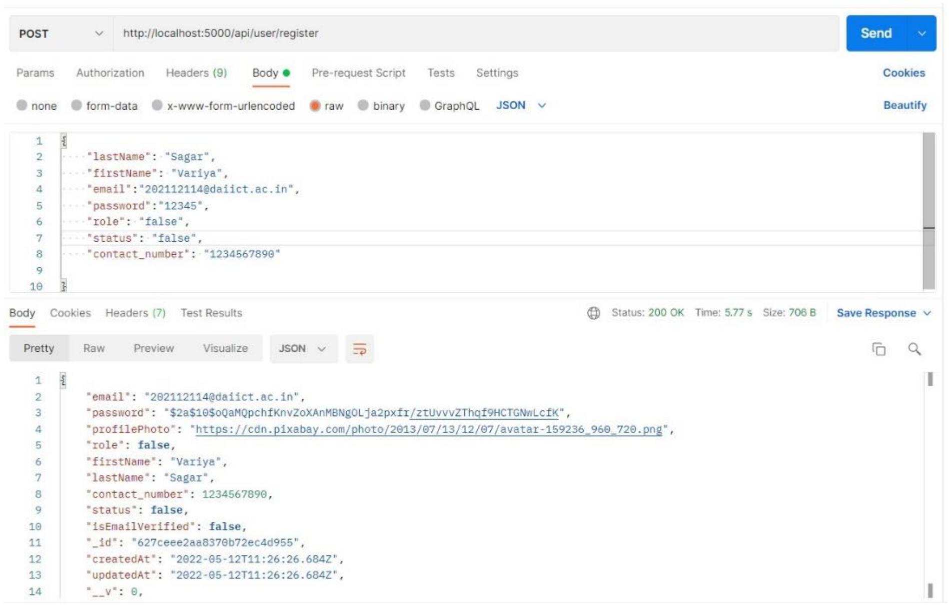
System Testing:

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

- Admin
- Faculty

White box testing

Register Faculty (Success)



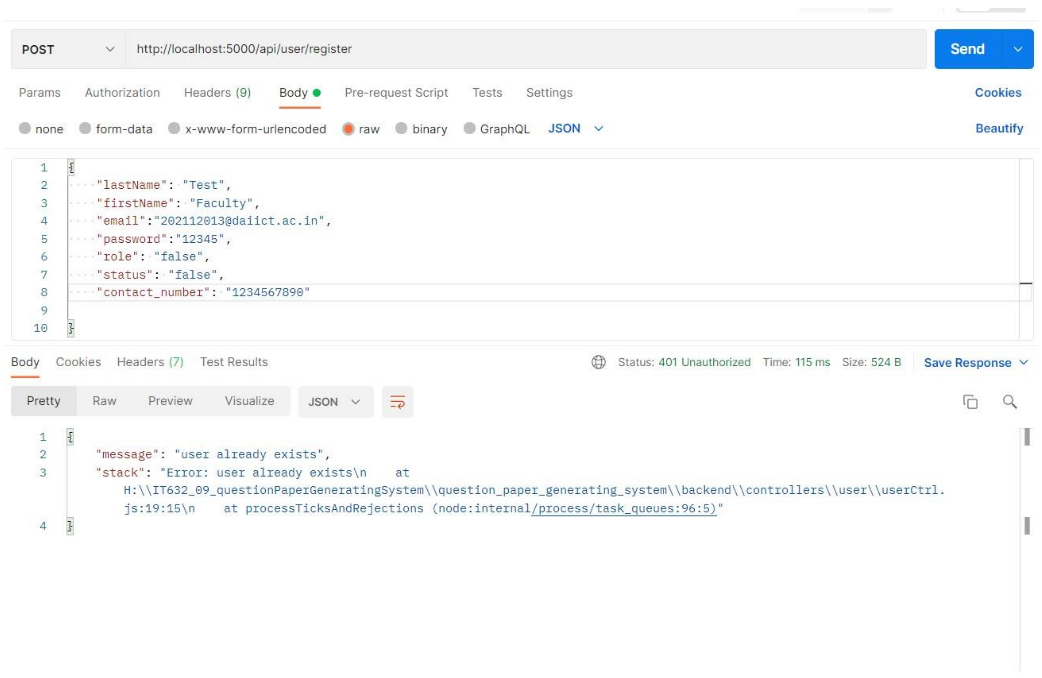
http://localhost:5000/api/user/register

```
{
    "lastName": "Sagar",
    "firstName": "Variya",
    "email":"202112114@daiict.ac.in",
    "password":"12345",
    "role": "false",
    "status": "false",
    "contact_number": "1234567890"

}
```

Status: 200 OK Time: 5.77 s Size: 706 B

```
{
    "email": "202112114@daiict.ac.in",
    "password": "$2a$10$oQaMQpchfKnvZoXAnMBNgOLja2pxfr/ztUvvvZThqf9HCTGNwLcfK",
    "profilePhoto": "https://cdn.pixabay.com/photo/2013/07/13/12/07/avatar-159236_960_720.png",
    "role": false,
    "firstName": "Variya",
    "lastName": "Sagar",
    "contact_number": 1234567890,
    "status": false,
    "isEmailVerified": false,
    "_id": "627ceee2aa8370b72ec4d955",
    "createdAt": "2022-05-12T11:26:26.684Z",
    "updatedAt": "2022-05-12T11:26:26.684Z",
    "__v": 0,
```

Register Faculty (Fail)



POST http://localhost:5000/api/user/register

Params · Authorization · Headers (9) · Body · Pre-request Script · Tests · Settings · Cookies

none · form-data · x-www-form-urlencoded · raw · binary · GraphQL · JSON · Beautify

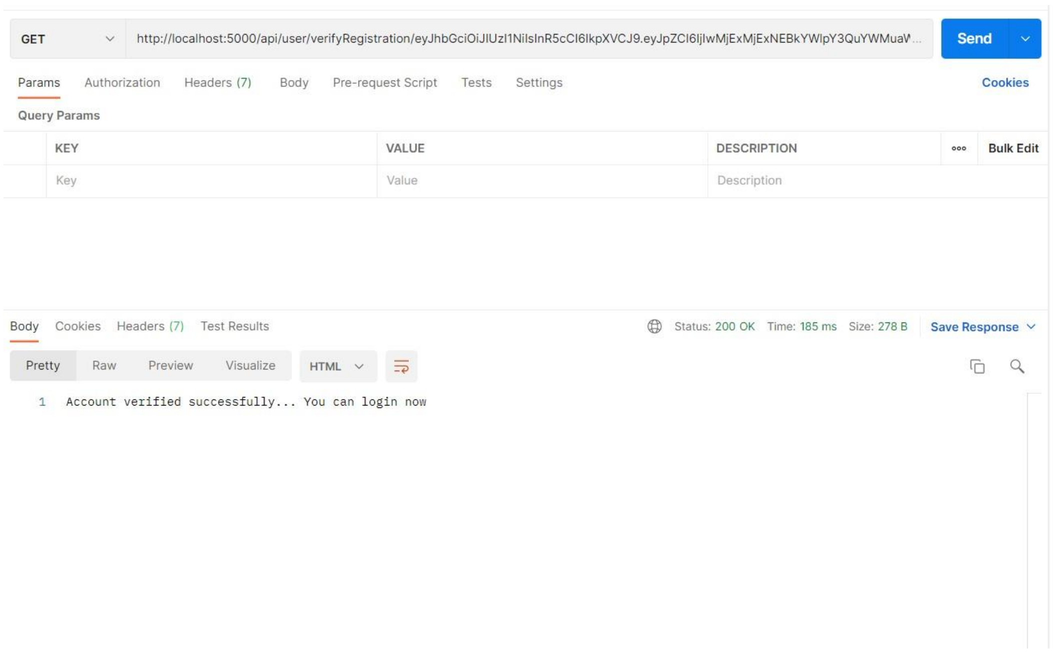
```
{
    "lastName": "Test",
    "firstName": "Faculty",
    "email":"202112013@daiict.ac.in",
    "password":"12345",
    "role": "false",
    "status": "false",
    "contact_number": "1234567890"

}
```

Body · Cookies · Headers (7) · Test Results — Status: 401 Unauthorized Time: 115 ms Size: 524 B Save Response

Pretty · Raw · Preview · Visualize · JSON

```
{
    "message": "user already exists",
    "stack": "Error: user already exists\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\controllers\\user\\userCtrl.
        js:19:15\n    at processTicksAndRejections (node:internal/process/task_queues:96:5)"
}
```

Register Email verification



GET http://localhost:5000/api/user/verifyRegistration/eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjIwMjExMjExNEBkYWlpY3QuYWMuaW...

Params · Authorization · Headers (7) · Body · Pre-request Script · Tests · Settings · Cookies

Query Params

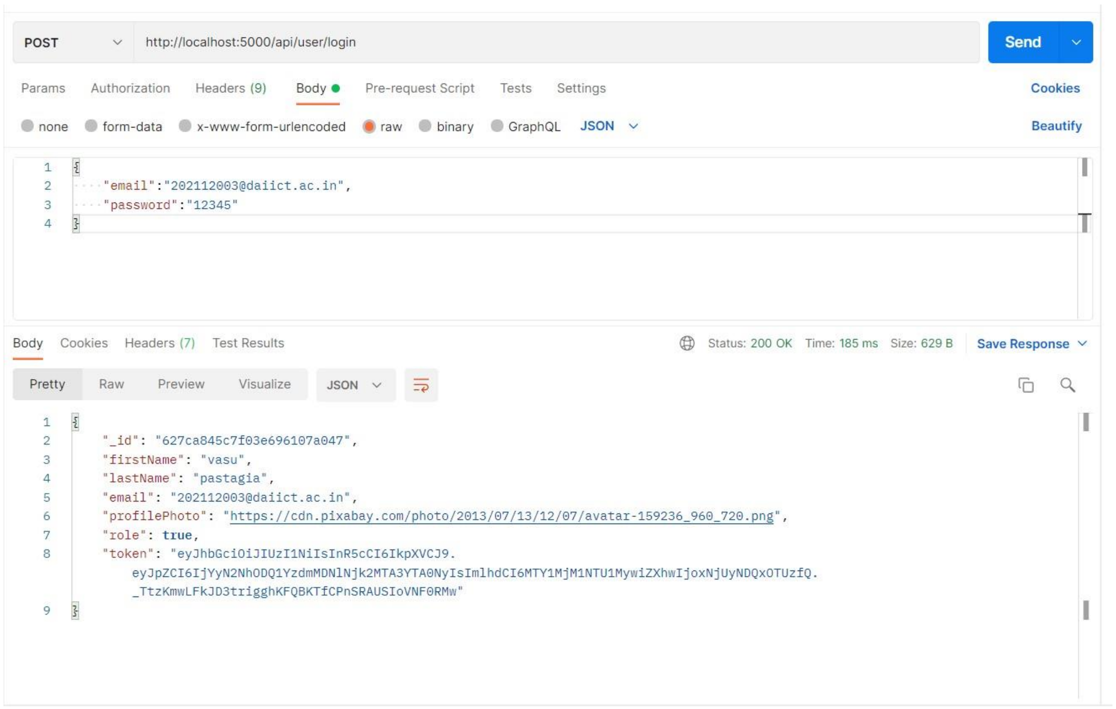
KEY	VALUE	DESCRIPTION
Key	Value	Description

Body · Cookies · Headers (7) · Test Results — Status: 200 OK Time: 185 ms Size: 278 B Save Response

Pretty · Raw · Preview · Visualize · HTML

```
Account verified successfully... You can login now
```

Login (Success)



POST http://localhost:5000/api/user/login

Body: ( ) none ( ) form-data ( ) x-www-form-urlencoded (●) raw ( ) binary ( ) GraphQL JSON

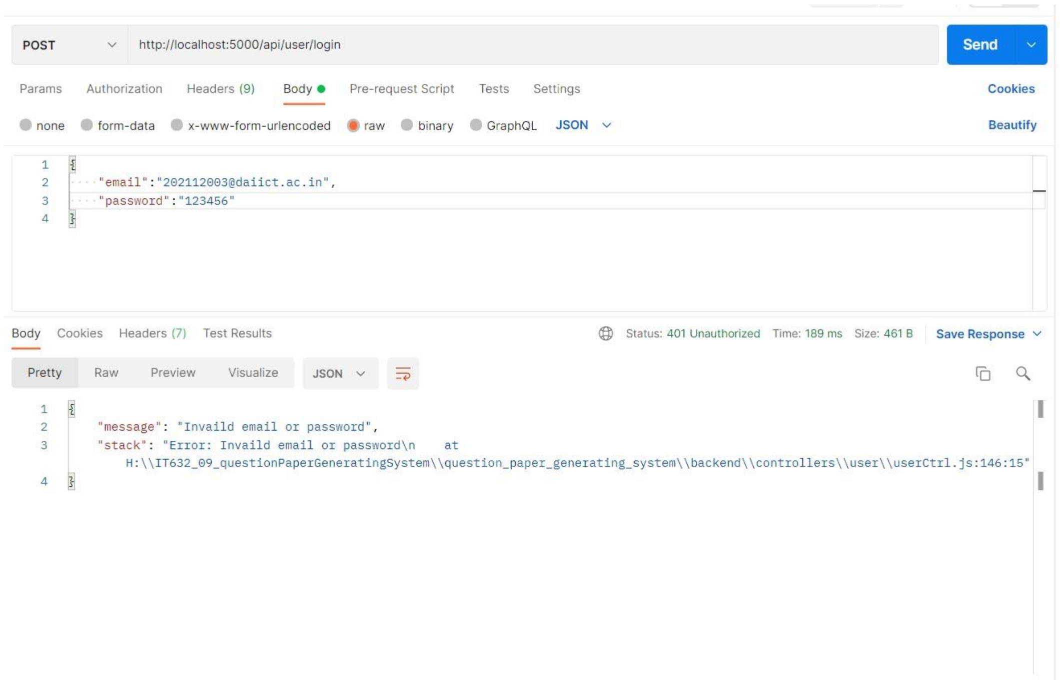
```
{
    "email":"202112003@daiict.ac.in",
    "password":"12345"
}
```

Status: 200 OK Time: 185 ms Size: 629 B

JSON

```
{
    "_id": "627ca845c7f03e696107a047",
    "firstName": "vasu",
    "lastName": "pastagia",
    "email": "202112003@daiict.ac.in",
    "profilePhoto": "https://cdn.pixabay.com/photo/2013/07/13/12/07/avatar-159236_960_720.png",
    "role": true,
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJpZCI6IjYyN2NhODQ1YzdmMDNlNjk2MTA3YTA0NyIsImlhdCI6MTY1MjM1NTU1MywiZXhwIjoxNjUyNDQxOTUzfQ.
        _TtzKmwLFkJD3trigghKFQBKTfCPnSRAUSIoVNF0RMw"
}
```

Login (Fail)



POST http://localhost:5000/api/user/login

Body: ( ) none ( ) form-data ( ) x-www-form-urlencoded (●) raw ( ) binary ( ) GraphQL JSON

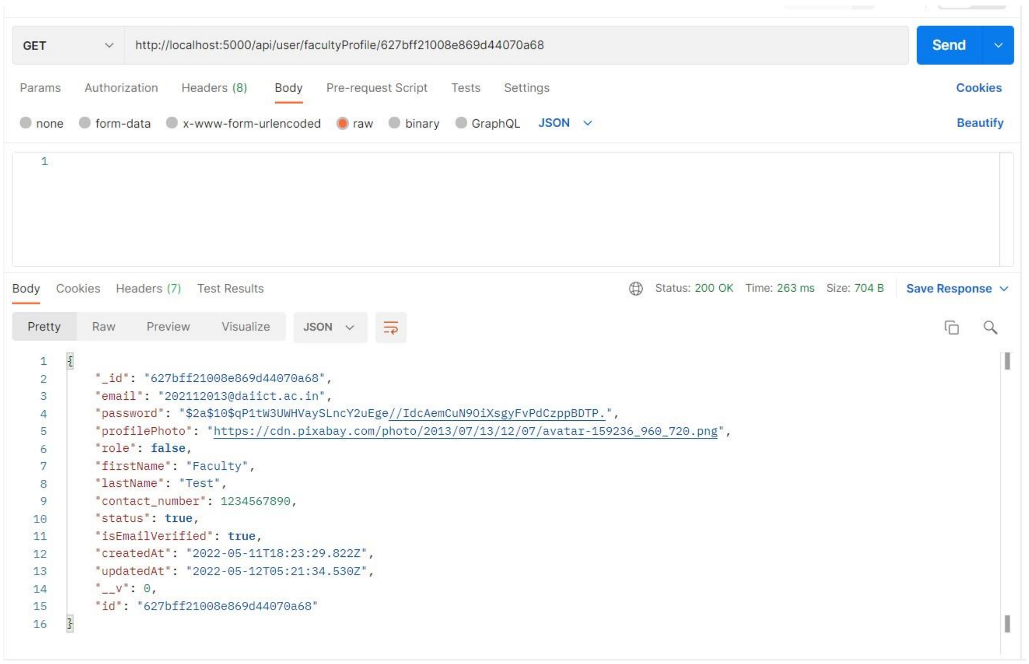
```
{
    "email":"202112003@daiict.ac.in",
    "password":"123456"
}
```

Status: 401 Unauthorized Time: 189 ms Size: 461 B

JSON

```
{
    "message": "Invaild email or password",
    "stack": "Error: Invaild email or password\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\controllers\\user\\userCtrl.js:146:15"
}
```

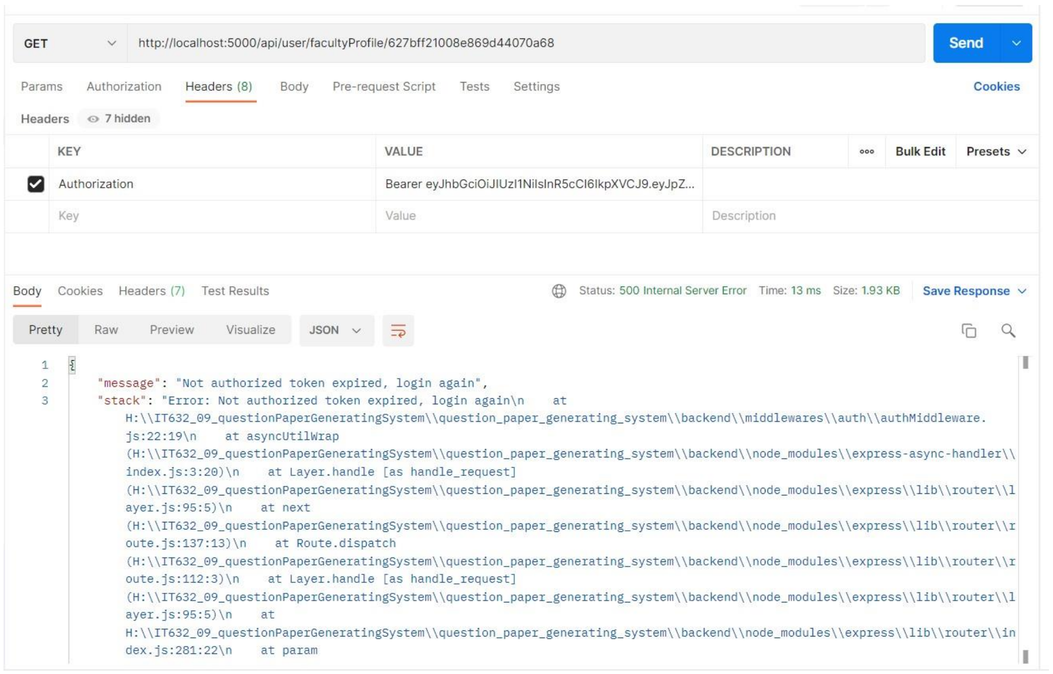
Fetch one Faculty (Success)


GET http://localhost:5000/api/user/facultyProfile/627bff21008e869d44070a68

Status: 200 OK Time: 263 ms Size: 704 B

```
{
    "_id": "627bff21008e869d44070a68",
    "email": "202112013@daiict.ac.in",
    "password": "$2a$10$qP1tW3UWHVaySLncY2uEge//IdcAemCuN9OiXsgyFvPdCzppBDTP.",
    "profilePhoto": "https://cdn.pixabay.com/photo/2013/07/13/12/07/avatar-159236_960_720.png",
    "role": false,
    "firstName": "Faculty",
    "lastName": "Test",
    "contact_number": 1234567890,
    "status": true,
    "isEmailVerified": true,
    "createdAt": "2022-05-11T18:23:29.822Z",
    "updatedAt": "2022-05-12T05:21:34.530Z",
    "__v": 0,
    "id": "627bff21008e869d44070a68"
}
```

Fetch one Faculty (Fail)


GET http://localhost:5000/api/user/facultyProfile/627bff21008e869d44070a68

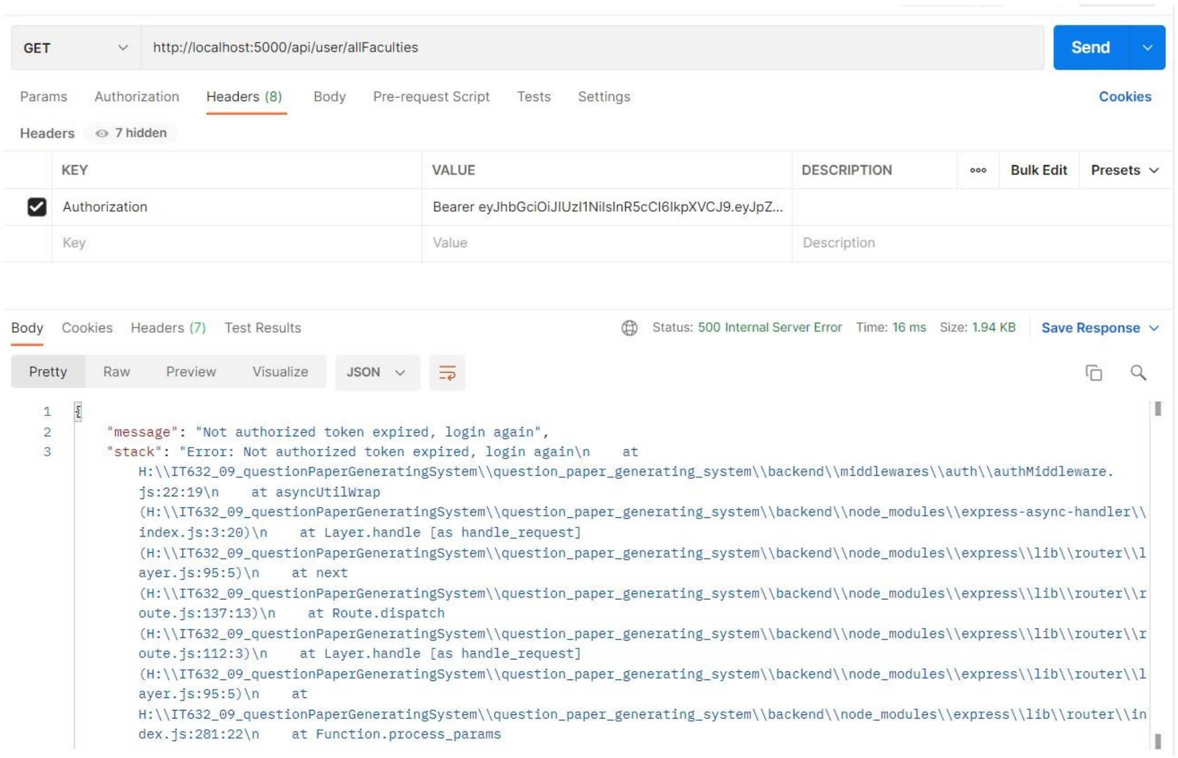
Headers (8)

KEY	VALUE	DESCRIPTION
✓ Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZ...	

Status: 500 Internal Server Error Time: 13 ms Size: 1.93 KB

```
{
    "message": "Not authorized token expired, login again",
    "stack": "Error: Not authorized token expired, login again\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\middlewares\\auth\\authMiddleware.
        js:22:19\n    at asyncUtilWrap
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express-async-handler\\
        index.js:3:20)\n    at Layer.handle [as handle_request]
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\l
        ayer.js:95:5)\n    at next
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\r
        oute.js:137:13)\n    at Route.dispatch
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\r
        oute.js:112:3)\n    at Layer.handle [as handle_request]
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\l
        ayer.js:95:5)\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\in
        dex.js:281:22\n    at param
```

Fetch All Faculties (Fail)



GET http://localhost:5000/api/user/allFaculties

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

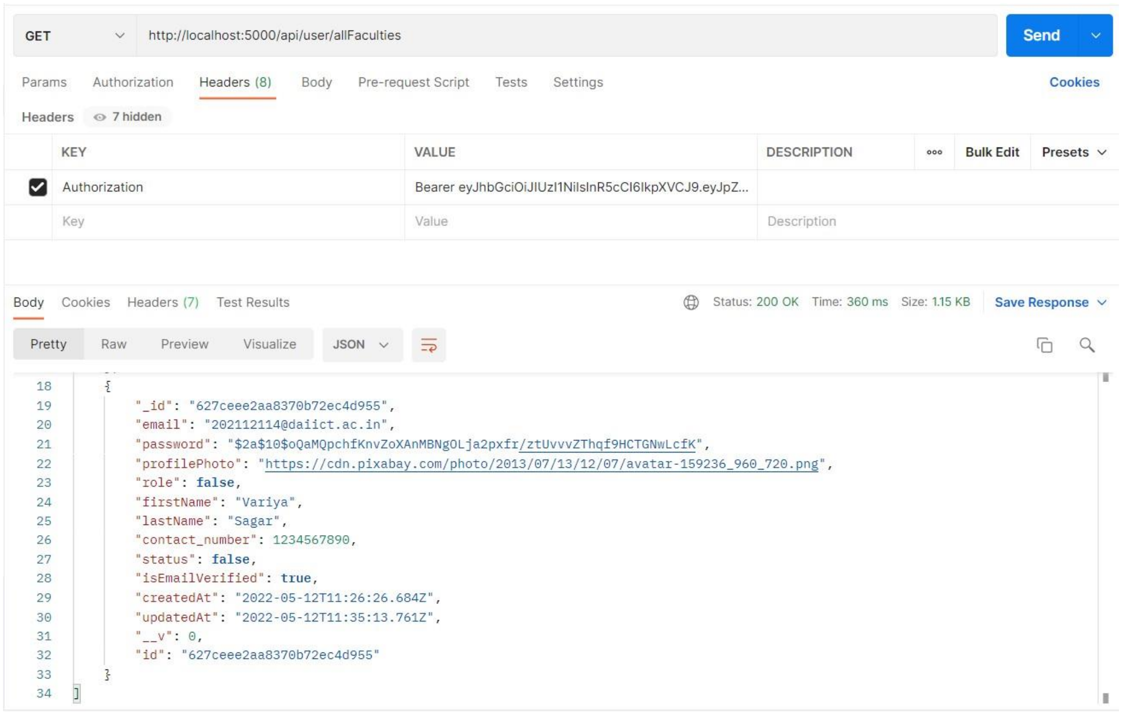
Headers 7 hidden

KEY	VALUE	DESCRIPTION
✓ Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZ...	

Body Cookies Headers (7) Test Results — Status: 500 Internal Server Error Time: 16 ms Size: 1.94 KB

```
{
    "message": "Not authorized token expired, login again",
    "stack": "Error: Not authorized token expired, login again\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\middlewares\\auth\\authMiddleware.
        js:22:19\n    at asyncUtilWrap
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express-async-handler\\
        index.js:3:20)\n    at Layer.handle [as handle_request]
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\l
        ayer.js:95:5)\n    at next
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\r
        oute.js:137:13)\n    at Route.dispatch
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\r
        oute.js:112:3)\n    at Layer.handle [as handle_request]
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\l
        ayer.js:95:5)\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\express\\lib\\router\\in
        dex.js:281:22\n    at Function.process_params
```

Fetch all faculties (Success)



GET http://localhost:5000/api/user/allFaculties

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

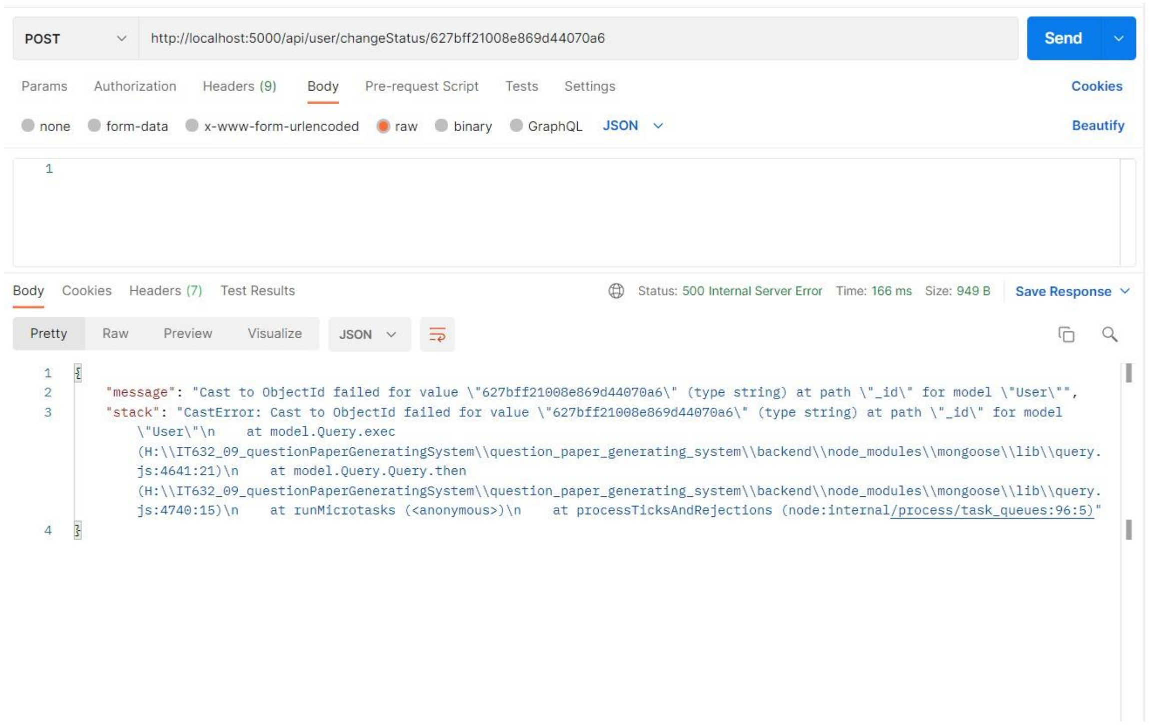
Headers 7 hidden

KEY	VALUE	DESCRIPTION
✓ Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZ...	

Body Cookies Headers (7) Test Results — Status: 200 OK Time: 360 ms Size: 1.15 KB

```
    {
        "_id": "627ceee2aa8370b72ec4d955",
        "email": "202112114@daiict.ac.in",
        "password": "$2a$10$oQaMQpchfKnvZoXAnMBNgOLja2pxfr/ztUvvvZThqf9HCTGNwLcfK",
        "profilePhoto": "https://cdn.pixabay.com/photo/2013/07/13/12/07/avatar-159236_960_720.png",
        "role": false,
        "firstName": "Variya",
        "lastName": "Sagar",
        "contact_number": 1234567890,
        "status": false,
        "isEmailVerified": true,
        "createdAt": "2022-05-12T11:26:26.684Z",
        "updatedAt": "2022-05-12T11:35:13.761Z",
        "__v": 0,
        "id": "627ceee2aa8370b72ec4d955"
    }
]
```

Faculty status change (Fail)



POST http://localhost:5000/api/user/changeStatus/627bff21008e869d44070a6

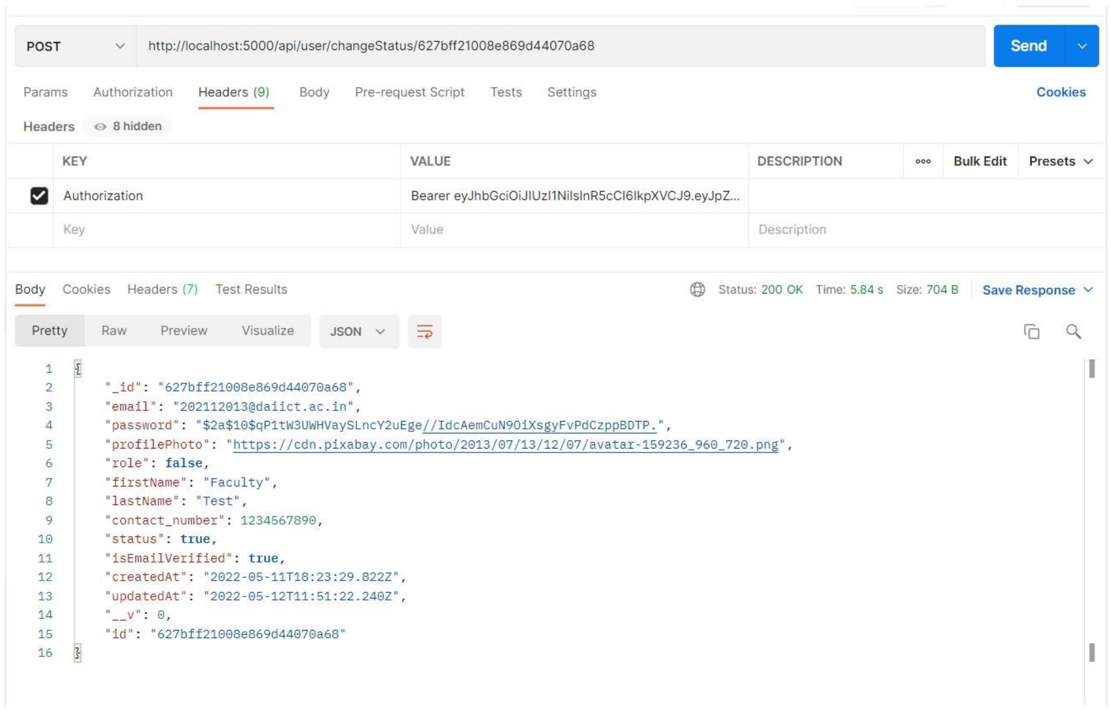
Params · Authorization · Headers (9) · **Body** · Pre-request Script · Tests · Settings · Cookies

none · form-data · x-www-form-urlencoded · [✓ raw] · binary · GraphQL · JSON · Beautify

Body · Cookies · Headers (7) · Test Results — Status: 500 Internal Server Error Time: 166 ms Size: 949 B

```
{
    "message": "Cast to ObjectId failed for value \"627bff21008e869d44070a6\" (type string) at path \"_id\" for model \"User\"",
    "stack": "CastError: Cast to ObjectId failed for value \"627bff21008e869d44070a6\" (type string) at path \"_id\" for model
        \"User\"\n    at model.Query.exec
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\mongoose\\lib\\query.
        js:4641:21)\n    at model.Query.Query.then
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\mongoose\\lib\\query.
        js:4740:15)\n    at runMicrotasks (<anonymous>)\n    at processTicksAndRejections (node:internal/process/task_queues:96:5)"
}
```

Faculty status change (Success)



POST http://localhost:5000/api/user/changeStatus/627bff21008e869d44070a68

Params · Authorization · **Headers (9)** · Body · Pre-request Script · Tests · Settings · Cookies

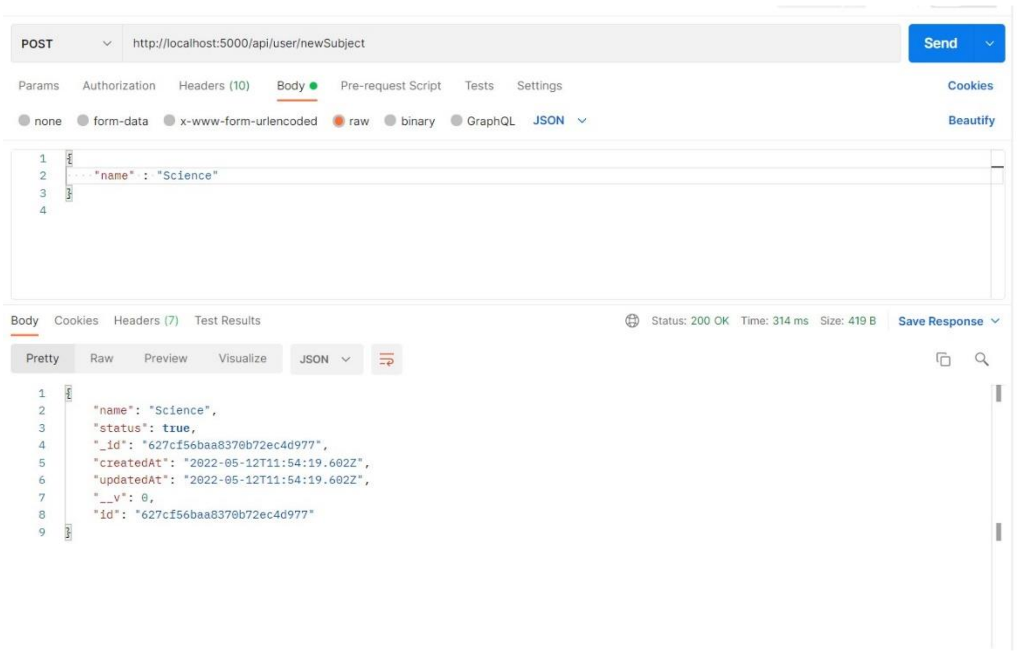
Headers · 8 hidden

	KEY	VALUE	DESCRIPTION
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZ...	
	Key	Value	Description

Body · Cookies · Headers (7) · Test Results — Status: 200 OK Time: 5.84 s Size: 704 B

```
{
    "_id": "627bff21008e869d44070a68",
    "email": "202112013@daiict.ac.in",
    "password": "$2a$10$qP1tW3UWHVaySLncY2uEge//IdcAemCuN9OiXsgyFvPdCzppBDTP.",
    "profilePhoto": "https://cdn.pixabay.com/photo/2013/07/13/12/07/avatar-159236_960_720.png",
    "role": false,
    "firstName": "Faculty",
    "lastName": "Test",
    "contact_number": 1234567890,
    "status": true,
    "isEmailVerified": true,
    "createdAt": "2022-05-11T18:23:29.822Z",
    "updatedAt": "2022-05-12T11:51:22.240Z",
    "__v": 0,
    "id": "627bff21008e869d44070a68"
}
```

Add new subject by admin (Fail)



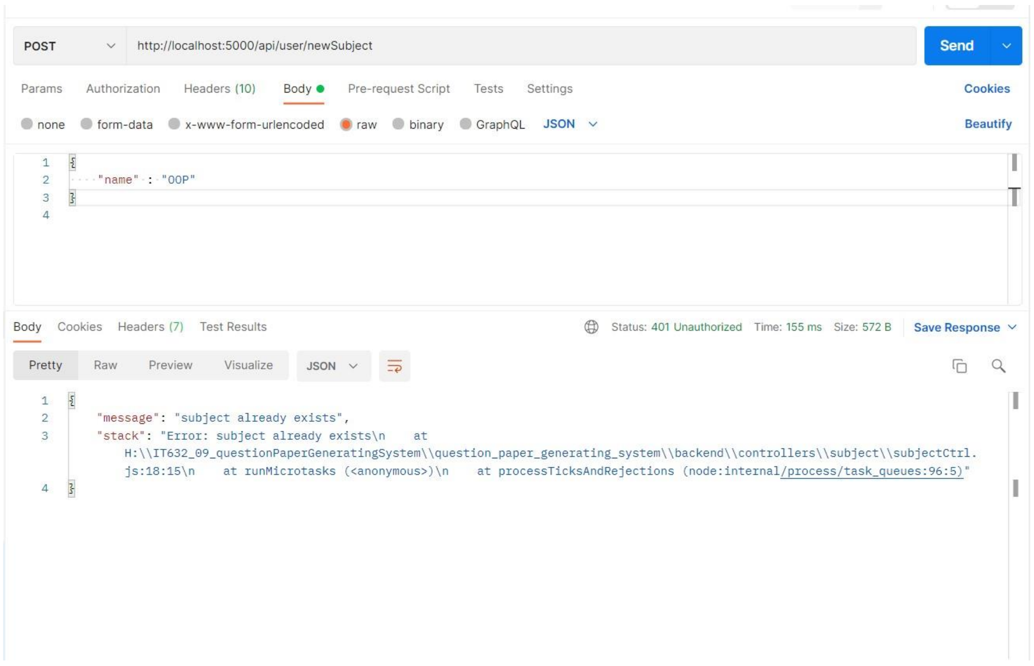
http://localhost:5000/api/user/newSubject

```
{
    "name" : "Science"
}
```

Status: 200 OK Time: 314 ms Size: 419 B

```
{
    "name": "Science",
    "status": true,
    "_id": "627cf56baa8370b72ec4d977",
    "createdAt": "2022-05-12T11:54:19.602Z",
    "updatedAt": "2022-05-12T11:54:19.602Z",
    "__v": 0,
    "id": "627cf56baa8370b72ec4d977"
}
```

Add new subject by admin (Success)



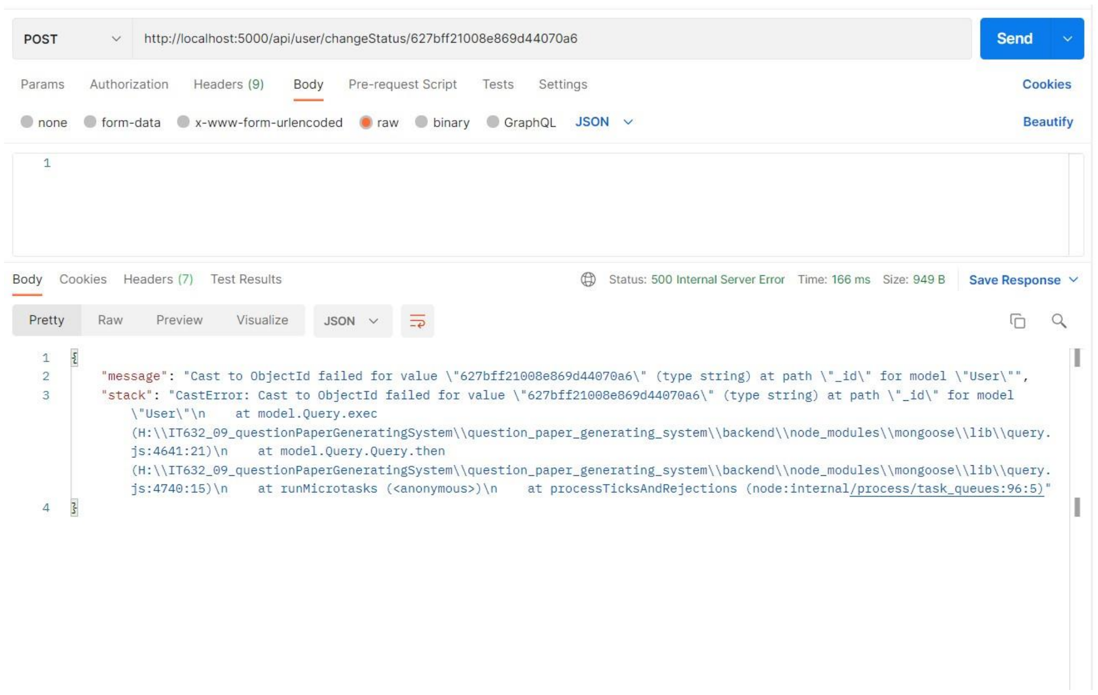
http://localhost:5000/api/user/newSubject

```
{
    "name" : "OOP"
}
```

Status: 401 Unauthorized Time: 155 ms Size: 572 B

```
{
    "message": "subject already exists",
    "stack": "Error: subject already exists\n    at
       H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\controllers\\subject\\subjectCtrl.
       js:18:15\n    at runMicrotasks (<anonymous>)\n    at processTicksAndRejections (node:internal/process/task_queues:96:5)"
}
```

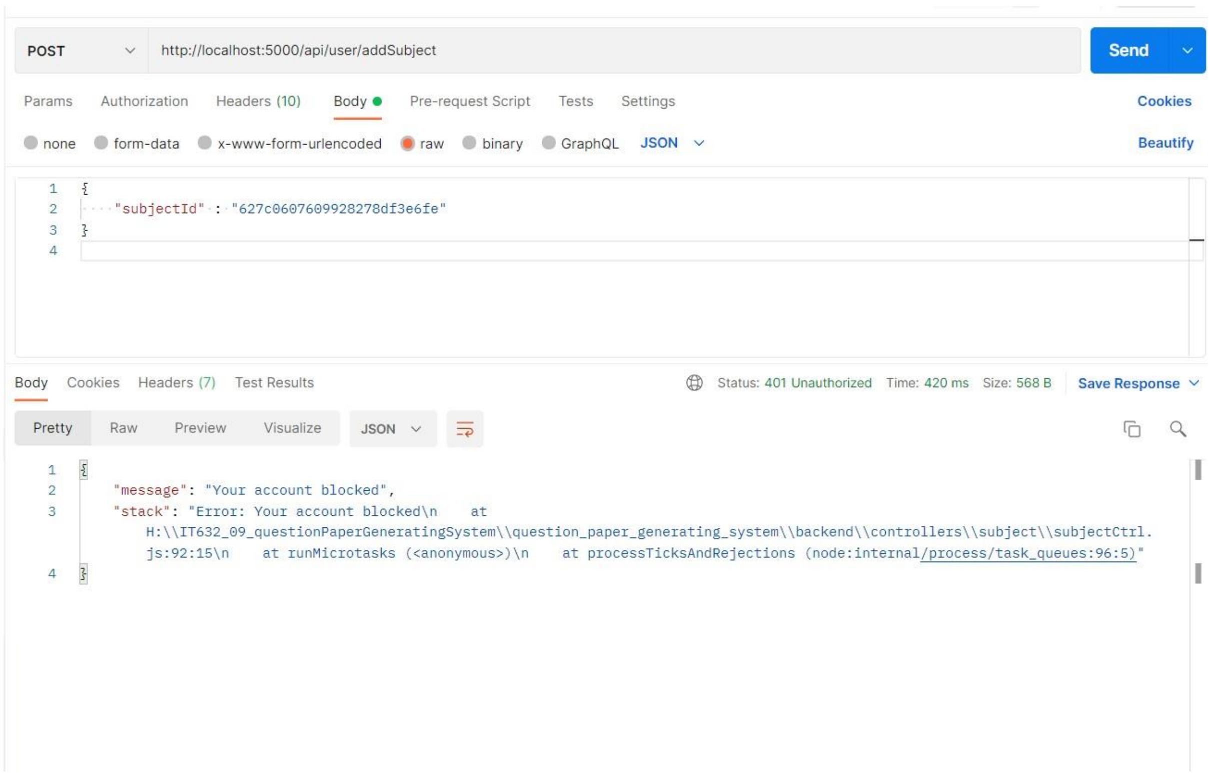
Add new teaching subject by faculty



POST
http://localhost:5000/api/user/changeStatus/627bff21008e869d44070a6
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
Body Cookies Headers (7) Test Results
Status: 500 Internal Server Error Time: 166 ms Size: 949 B Save Response
Pretty Raw Preview Visualize JSON

```
{
    "message": "Cast to ObjectId failed for value \"627bff21008e869d44070a6\" (type string) at path \"_id\" for model \"User\"",
    "stack": "CastError: Cast to ObjectId failed for value \"627bff21008e869d44070a6\" (type string) at path \"_id\" for model
        \"User\"\n    at model.Query.exec
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\mongoose\\lib\\query.
        js:4641:21)\n    at model.Query.Query.then
        (H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\node_modules\\mongoose\\lib\\query.
        js:4740:15)\n    at runMicrotasks (<anonymous>)\n    at processTicksAndRejections (node:internal/process/task_queues:96:5)"
}
```

Add new Teaching subject (Fail)

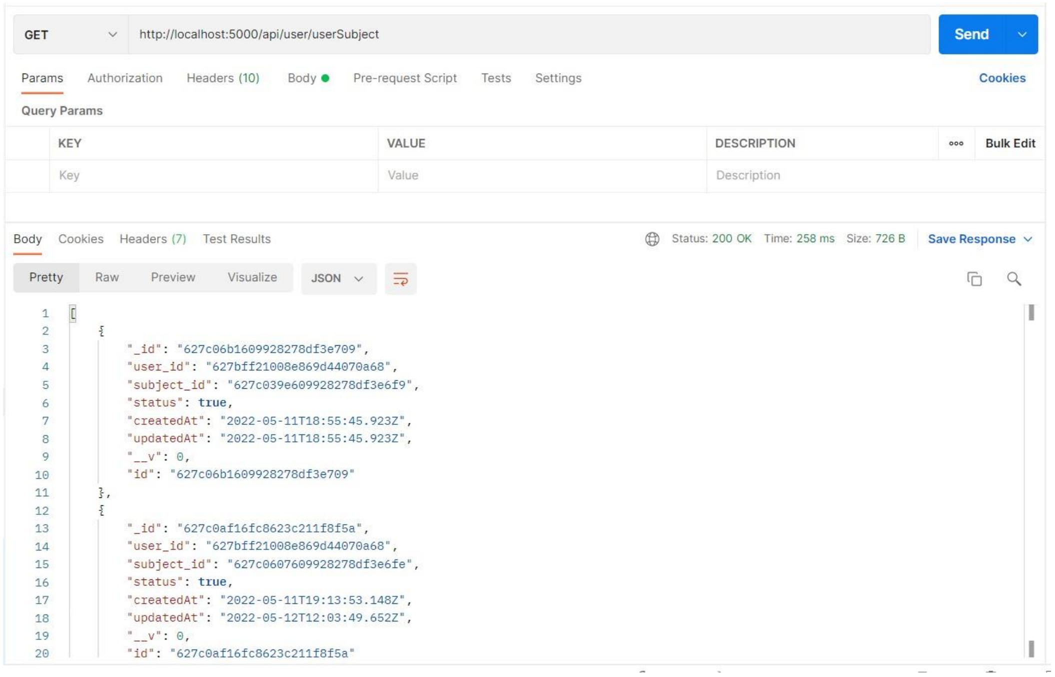


POST
http://localhost:5000/api/user/addSubject
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify
Body Cookies Headers (7) Test Results
Status: 401 Unauthorized Time: 420 ms Size: 568 B Save Response
Pretty Raw Preview Visualize JSON

```
{
    "subjectId" : "627c0607609928278df3e6fe"
}
```

```
{
    "message": "Your account blocked",
    "stack": "Error: Your account blocked\n    at
        H:\\IT632_09_questionPaperGeneratingSystem\\question_paper_generating_system\\backend\\controllers\\subject\\subjectCtrl.
        js:92:15\n    at runMicrotasks (<anonymous>)\n    at processTicksAndRejections (node:internal/process/task_queues:96:5)"
}
```

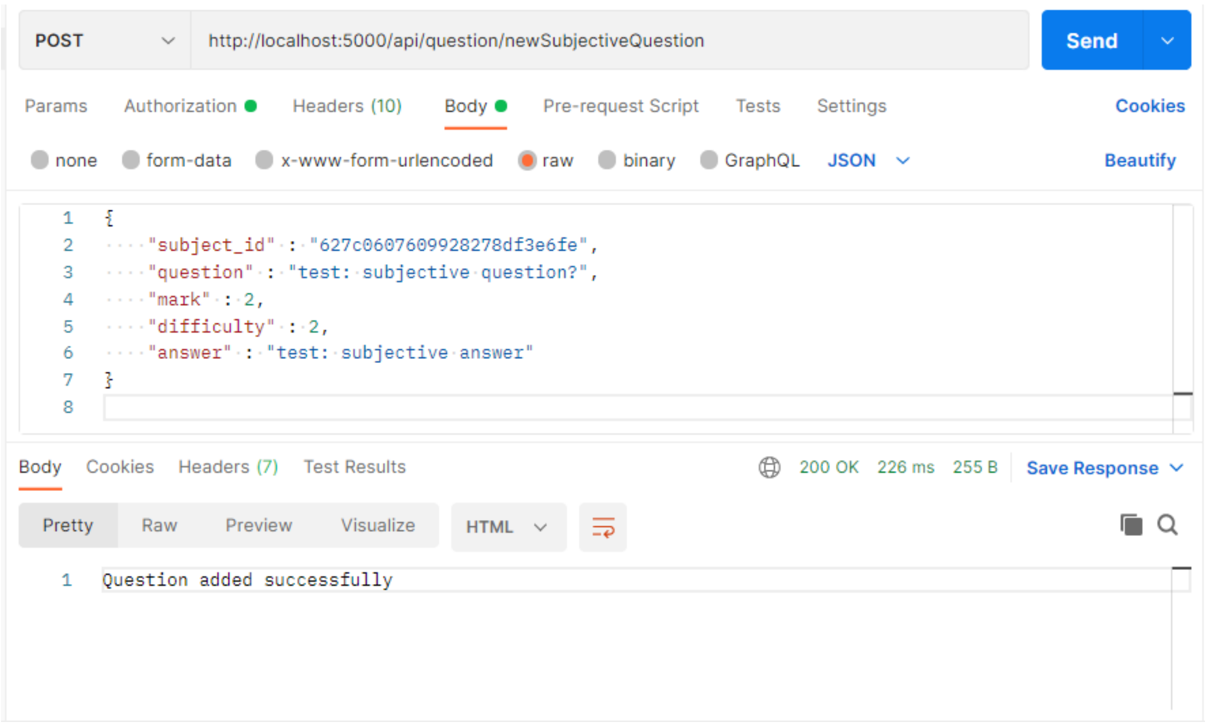
Fetch Teaching Subject for faculty (success)


GET http://localhost:5000/api/user/userSubject

Status: 200 OK Time: 258 ms Size: 726 B

```
[
    {
        "_id": "627c06b1609928278df3e709",
        "user_id": "627bff21008e869d44070a68",
        "subject_id": "627c039e609928278df3e6f9",
        "status": true,
        "createdAt": "2022-05-11T18:55:45.923Z",
        "updatedAt": "2022-05-11T18:55:45.923Z",
        "__v": 0,
        "id": "627c06b1609928278df3e709"
    },
    {
        "_id": "627c0af16fc8623c211f8f5a",
        "user_id": "627bff21008e869d44070a68",
        "subject_id": "627c0607609928278df3e6fe",
        "status": true,
        "createdAt": "2022-05-11T19:13:53.148Z",
        "updatedAt": "2022-05-12T12:03:49.652Z",
        "__v": 0,
        "id": "627c0af16fc8623c211f8f5a"
```

Faculty add subjective:

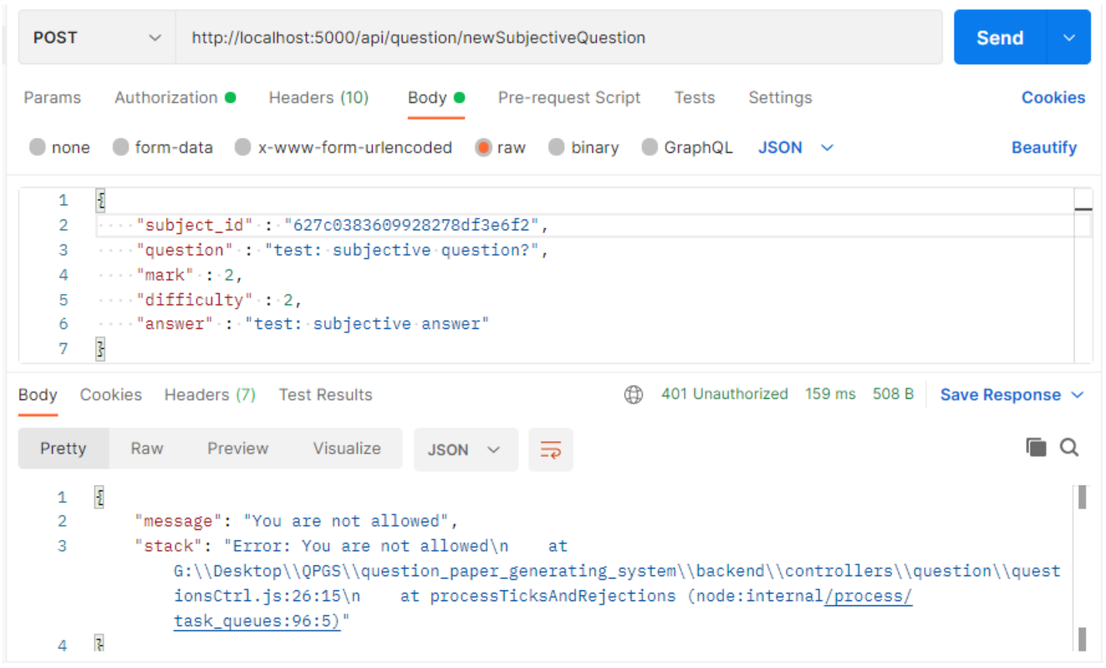

POST http://localhost:5000/api/question/newSubjectiveQuestion

```
{
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: subjective question?",
    "mark" : 2,
    "difficulty" : 2,
    "answer" : "test: subjective answer"
}
```

200 OK 226 ms 255 B

```
Question added successfully
```

Faculty add subjective (wrong subject):

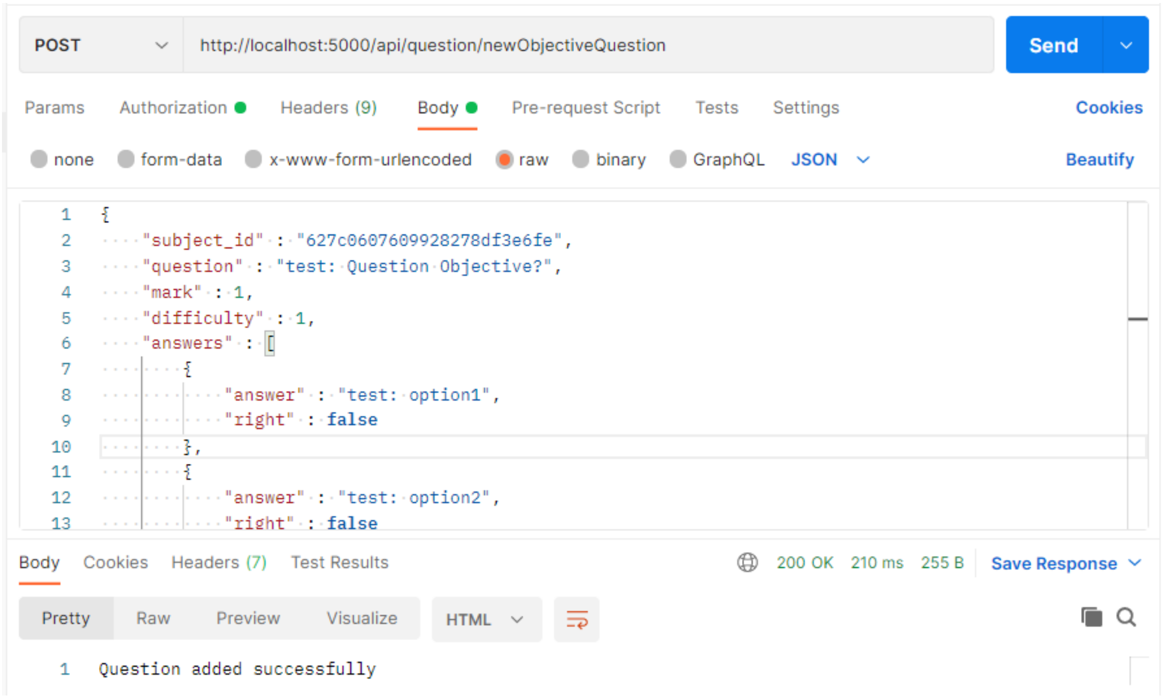


```
http://localhost:5000/api/question/newSubjectiveQuestion
```

```
{
    "subject_id" : "627c0383609928278df3e6f2",
    "question" : "test: subjective question?",
    "mark" : 2,
    "difficulty" : 2,
    "answer" : "test: subjective answer"
}
```

```
{
    "message": "You are not allowed",
    "stack": "Error: You are not allowed\n    at
        G:\\Desktop\\QPGS\\question_paper_generating_system\\backend\\controllers\\question\\quest
        ionsCtrl.js:26:15\n    at processTicksAndRejections (node:internal/process/
        task_queues:96:5)"
}
```

Faculty add objective:

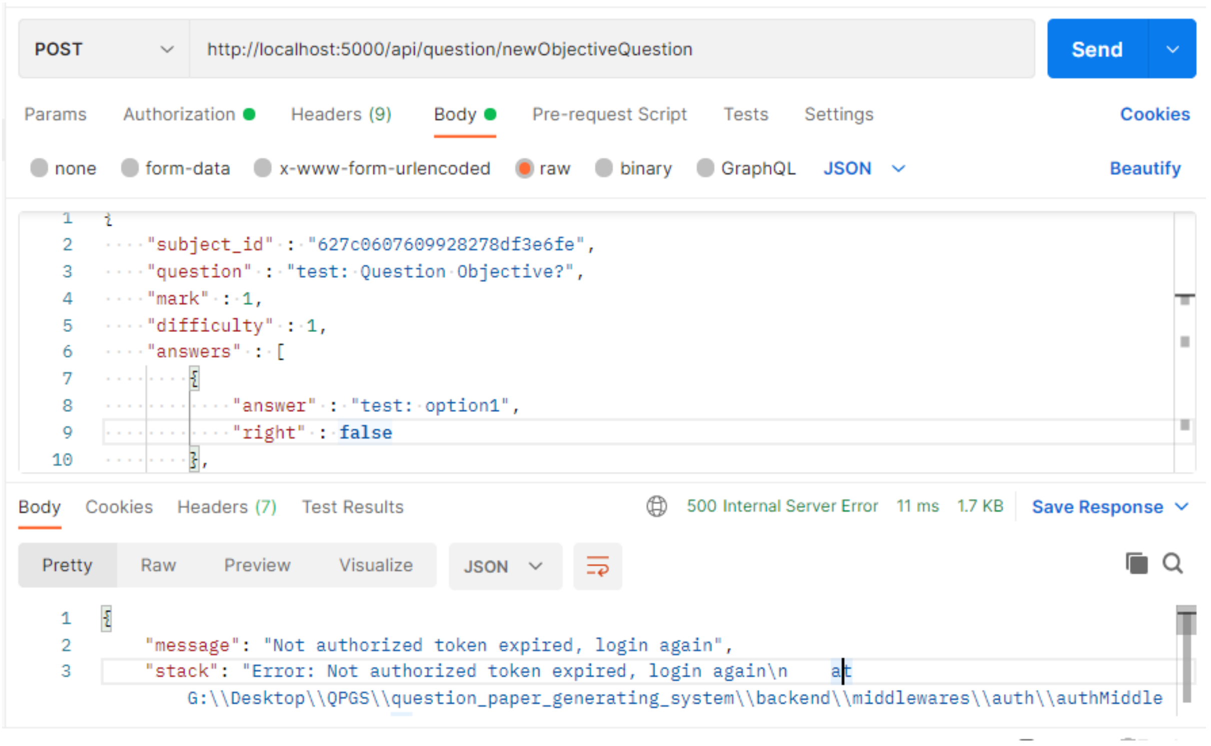


```
http://localhost:5000/api/question/newObjectiveQuestion
```

```
{
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: Question Objective?",
    "mark" : 1,
    "difficulty" : 1,
    "answers" : [
        {
            "answer" : "test: option1",
            "right" : false
        },
        {
            "answer" : "test: option2",
            "right" : false
```

```
Question added successfully
```

Faculty add objective (wrong):



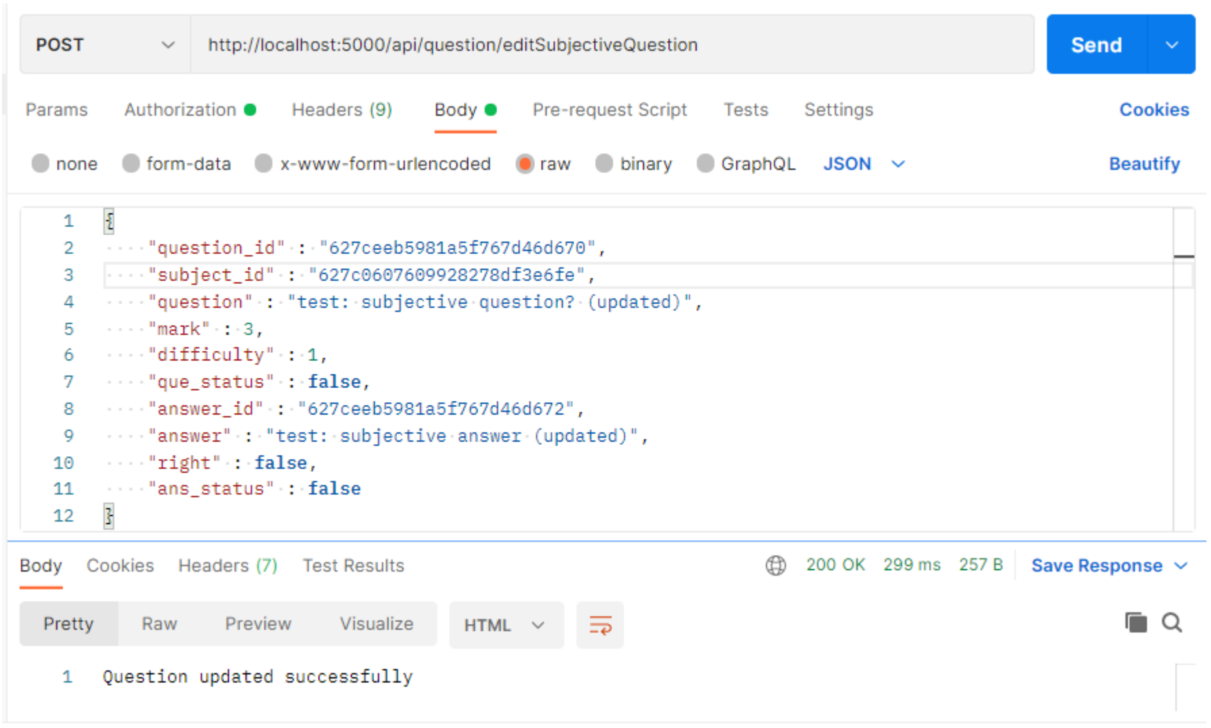
http://localhost:5000/api/question/newObjectiveQuestion

```
{
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: Question Objective?",
    "mark" : 1,
    "difficulty" : 1,
    "answers" : [
        {
            "answer" : "test: option1",
            "right" : false
        },
```

500 Internal Server Error 11 ms 1.7 KB

```
{
    "message": "Not authorized token expired, login again",
    "stack": "Error: Not authorized token expired, login again\n    at
        G:\\Desktop\\QPGS\\question_paper_generating_system\\backend\\middlewares\\auth\\authMiddle
```

Faculty edit subjective:



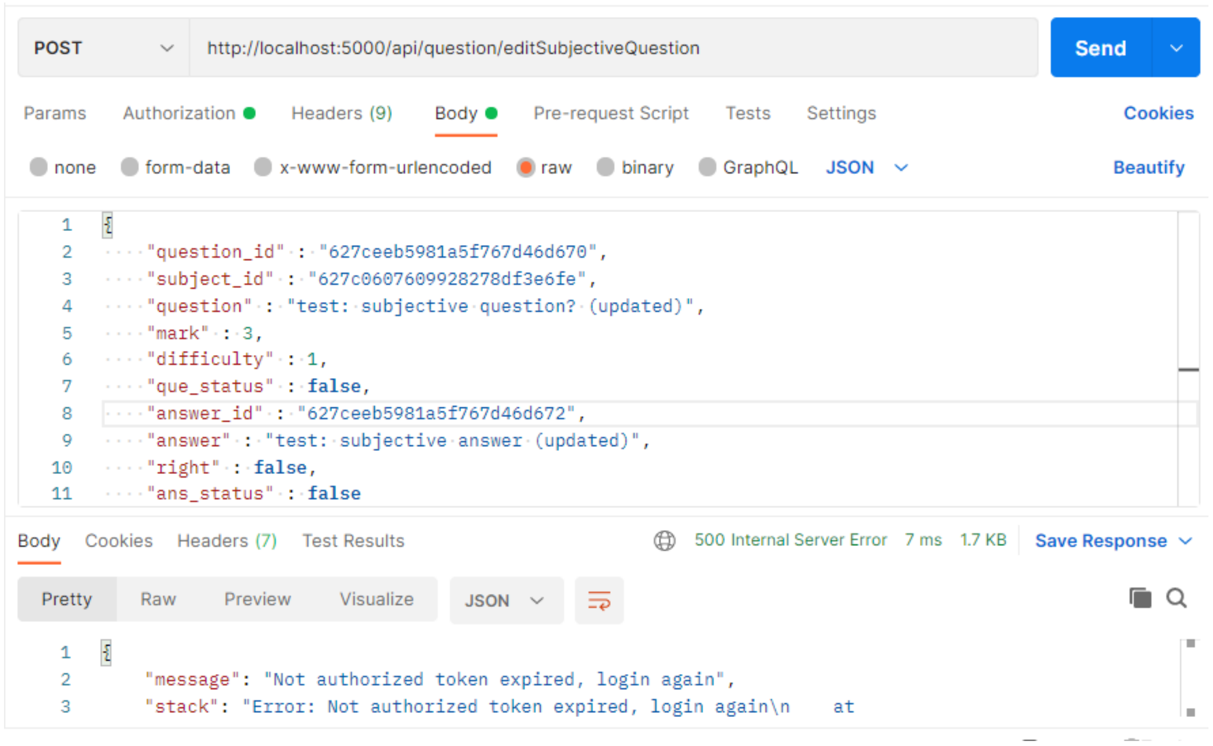
http://localhost:5000/api/question/editSubjectiveQuestion

```
{
    "question_id" : "627ceeb5981a5f767d46d670",
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: subjective question? (updated)",
    "mark" : 3,
    "difficulty" : 1,
    "que_status" : false,
    "answer_id" : "627ceeb5981a5f767d46d672",
    "answer" : "test: subjective answer (updated)",
    "right" : false,
    "ans_status" : false
}
```

200 OK 299 ms 257 B

```
Question updated successfully
```

Faculty edit subjective (wrong):



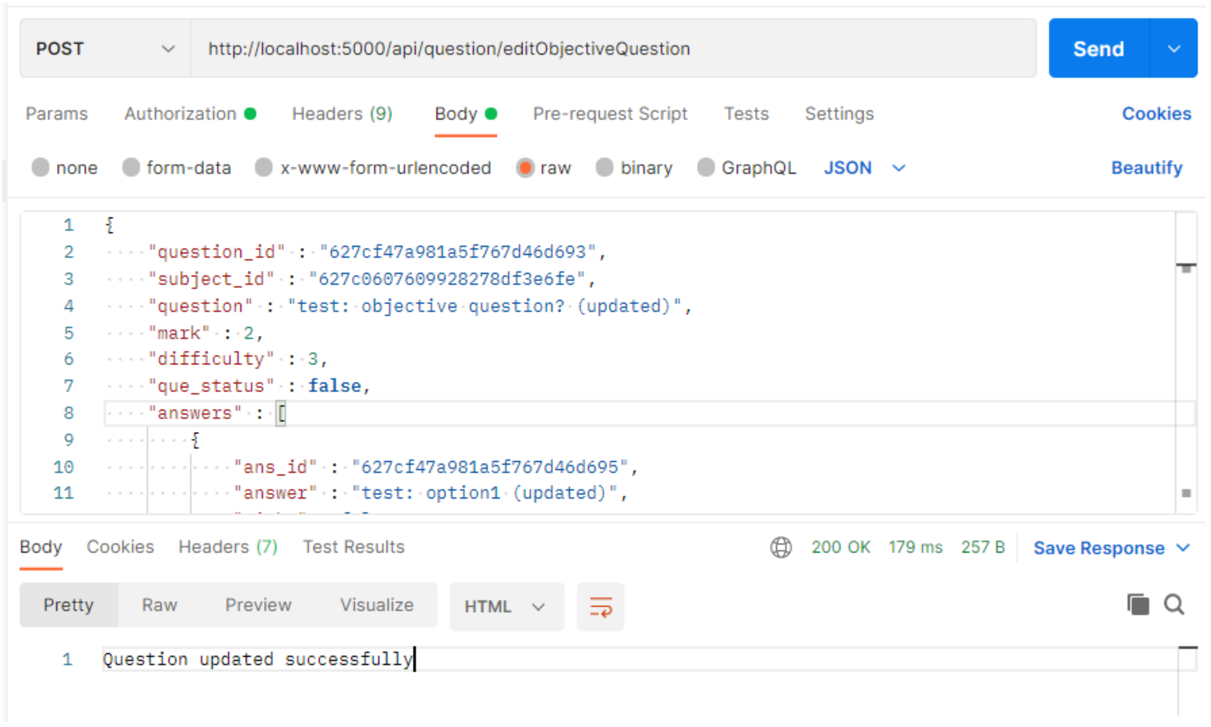
http://localhost:5000/api/question/editSubjectiveQuestion

```
{
    "question_id" : "627ceeb5981a5f767d46d670",
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: subjective question? (updated)",
    "mark" : 3,
    "difficulty" : 1,
    "que_status" : false,
    "answer_id" : "627ceeb5981a5f767d46d672",
    "answer" : "test: subjective answer (updated)",
    "right" : false,
    "ans_status" : false
```

500 Internal Server Error 7 ms 1.7 KB

```
{
    "message": "Not authorized token expired, login again",
    "stack": "Error: Not authorized token expired, login again\n    at
```

Faculty edit objective:



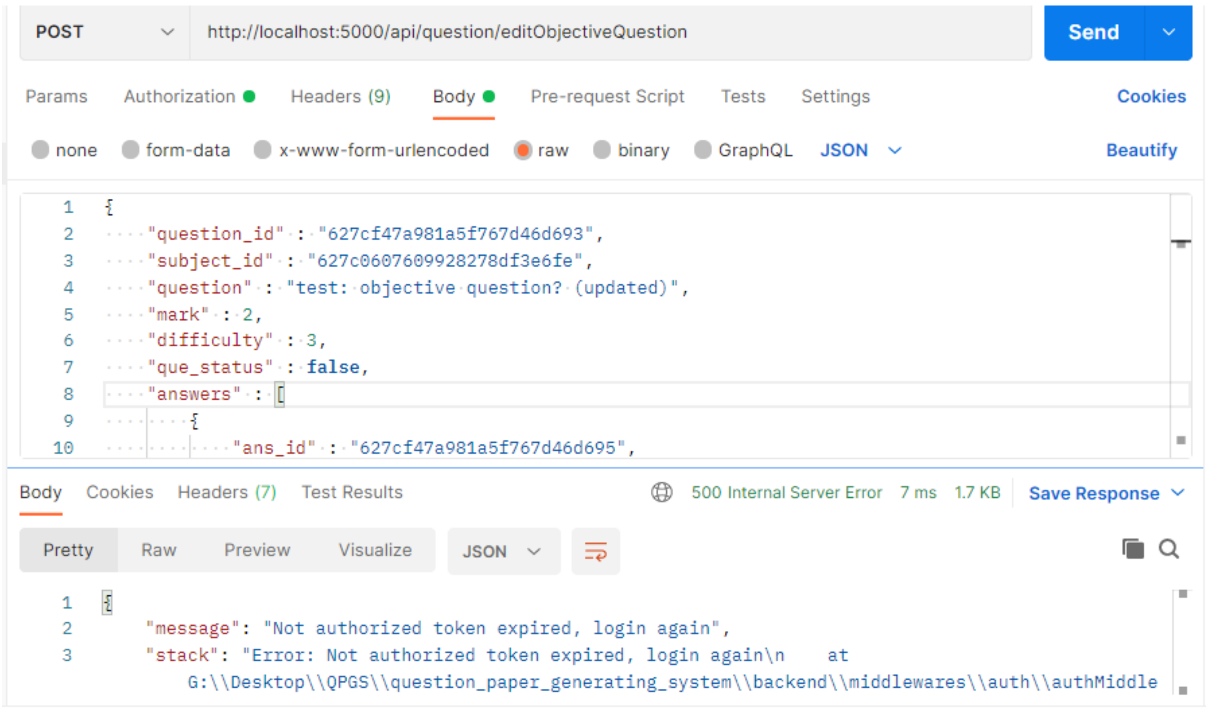
http://localhost:5000/api/question/editObjectiveQuestion

```
{
    "question_id" : "627cf47a981a5f767d46d693",
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: objective question? (updated)",
    "mark" : 2,
    "difficulty" : 3,
    "que_status" : false,
    "answers" : [
        {
            "ans_id" : "627cf47a981a5f767d46d695",
            "answer" : "test: option1 (updated)",
```

200 OK 179 ms 257 B

```
Question updated successfully
```

Faculty edit objective (wrong):



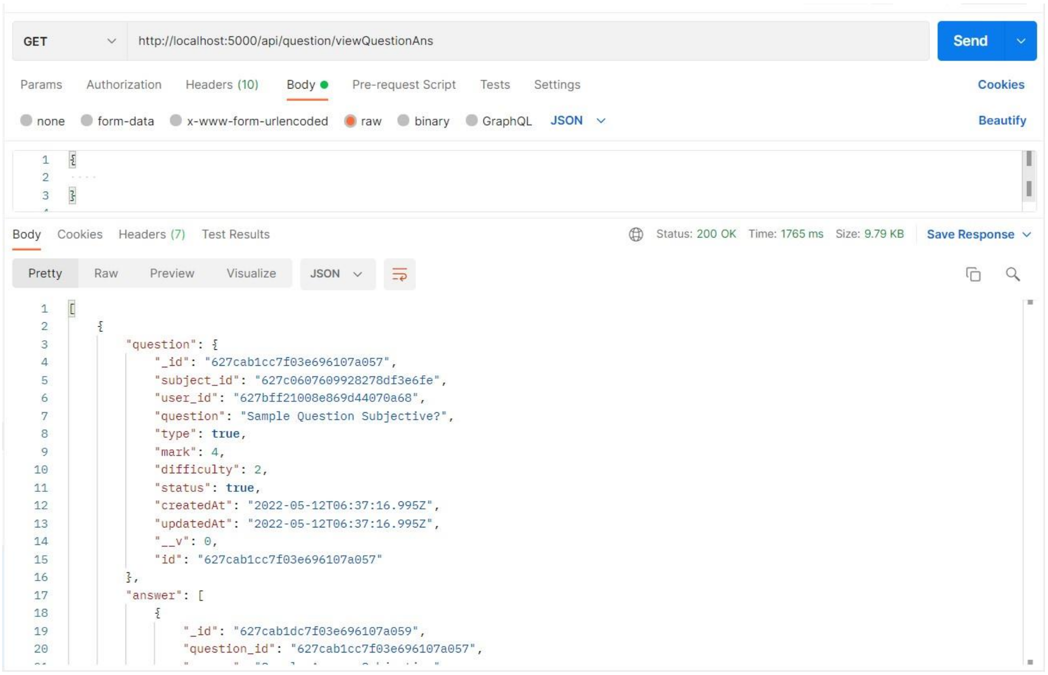
http://localhost:5000/api/question/editObjectiveQuestion

```
{
    "question_id" : "627cf47a981a5f767d46d693",
    "subject_id" : "627c0607609928278df3e6fe",
    "question" : "test: objective question? (updated)",
    "mark" : 2,
    "difficulty" : 3,
    "que_status" : false,
    "answers" : [
        {
            "ans_id" : "627cf47a981a5f767d46d695",
```

500 Internal Server Error   7 ms   1.7 KB

```
{
    "message": "Not authorized token expired, login again",
    "stack": "Error: Not authorized token expired, login again\n    at
        G:\\Desktop\\QPGS\\question_paper_generating_system\\backend\\middlewares\\auth\\authMiddle
```

View Question Answer Combined (success)



http://localhost:5000/api/question/viewQuestionAns

```
{
    
}
```

Status: 200 OK   Time: 1765 ms   Size: 9.79 KB

```
[
    {
        "question": {
            "_id": "627cab1cc7f03e696107a057",
            "subject_id": "627c0607609928278df3e6fe",
            "user_id": "627bff21008e869d44070a68",
            "question": "Sample Question Subjective?",
            "type": true,
            "mark": 4,
            "difficulty": 2,
            "status": true,
            "createdAt": "2022-05-12T06:37:16.995Z",
            "updatedAt": "2022-05-12T06:37:16.995Z",
            "__v": 0,
            "id": "627cab1cc7f03e696107a057"
        },
        "answer": [
            {
                "_id": "627cab1dc7f03e696107a059",
                "question_id": "627cab1cc7f03e696107a057",
```

Challenges

- Time limitations plays a vital role in software development. When there is no sufficient time for the development some times the product don't meet the quality standards as the developers works under pressure and output decreases.

- It feels like a common problem when one developer works with another developer's code. This situation created a problem for the developer as it takes lot time of the new developer to understand the code.

- Keeping everyone together is crucial for a successful project. However, it became very challenging for the project leader to do it since each individual is from a different background and has distinct skills. So, there was a need to find ways to keep them on the same track. This made everyone share the same goal and were on the same page.

- Sometimes, unrealistic expectations and tight deadlines make the team demotivated. Plus, conflict among the members and the problems faced by them affected the project.

Open Issues

1. Currently, system doesn't allow generating paper according to topic selection.
2. System is not allowing both subjective and objective questions together.
3. Objective questions can only have 4 answer options available currently.

Lesson Learned

- Everything takes longer than you think. Especially in programming. It is hard to estimate how much time a feature will take even if everything goes smoothly.
- Start small, then extend.
    - Whether creating a new system, or adding a feature to an existing system, we always start by making a very simple version with almost none of the required functionality.
    - Then we extended the solution step by step, until it does what it is supposed to.
    - Instead, we learn as we go along, and this newly discovered information gets used in the solution.

- Change one thing at a time.
    - When you develop, and some tests fail, or a feature stops working, it's much easier to find the problem if you only changed one thing.

    - In other words, use short iterations. Do one thing, make sure it works, repeat.

- First understand the existing code.
    - Most coding requires changing existing code in some way. Even if it is a new feature, it needs to fit into the existing program.
    - And before you can fit the new stuff in, you need to understand the current solution.
    - This means that reading code is a skill that is as necessary as writing code. It is also part of the reason why seemingly small changes can still take a long time – you must understand the context in which you make the change.

- Fix the easiest bug first.
    - In many cases, there are quite a few bugs or problems that a developer knows how to solve or where to find a solution. So deal with them first, because most of the bugs are usually connected, and changes with one affect the other.
    - So, instead of dividing them into "less" and "more time-consuming to solve," divide them into "known" and "unknown."

Contribution

Student ID	Student Name	Contribution
202112003	Vasu Pastagia (Leader)	▪ Frontend ▪ Backend Integration ▪ Testing ▪ Documentation (analysis class diagram) ▪ Management
202112009	Dhruvi Jariwala	▪ Backend ▪ Final Documentation(Intro, scope, process model reasons, complete analysis class diagram, open issues) ▪ Final PPT
20212010	Anil Vaghari	▪ Frontend ▪ PPT ▪ Documentation(use case description, analysis class diagram)
20212013	Darshan Patil	▪ Backend ▪ Testing ▪ Documentation (analysis class diagram)
202112019	Pragati Khurana	▪ Frontend ▪ Backend Integration ▪ Documentation (analysis class diagram)
202112038	Arjun Solanki	▪ Backend

		▪ Documentation(requirement elicitation, use case description, object design, analysis class diagram)
202112075	Aditya Jain	▪ Documentation(subsystem design, analysis class diagram)
202112083	Dhrumi Shah	▪ Frontend ▪ PPT, ▪ Documentation(use case diagram, analysis class diagram, challenges and lesson learned)
202112112	Apoorv Jain	▪ Frontend ▪ Testing ▪ Documentation(analysis class diagram)
202112114	Sagar Variya	▪ Backend ▪ Testing ▪ Documentation(analysis class diagram)