Final Project

Web Engineering

Backend

Ali Hamza 20i1881

Syed Muhammad Mian Abdullah 20i0457

Mahad Rahat 20i1808

15 May, 2023

Work Division

1. Mahad Rahat

Module 1 and 4: Users must sign in to use the application. They will also be able to change their passwords and will be able to recover their accounts. Basically, there are two users such as Student (may act as team lead), Teacher (may act as supervisor) and an Admin as a separate entity.

1.1 Sign Up 1.2 Sign In 1.3 Send Notifications

1.4 Change Theme 1.5 Change Email 1.6 Change Password

1.7 FYP Status 1.8 Keep Signed In 1.9 SignOut

4 : Add, delete and edit requirements

2. Ali Hamza

Teacher will create a project and then add/edit project details. Students can search for a supervisor.

2.1 View Projects 2.2 Create a Project 2.3 Assign Project Lead

2.4 View Deliverables 2.5 Add , delete, update, get Supervisor Evaluation

2.6 Find Supervisor 2.7 Delete Projects , Update project, get project

3. Mian Abdullah

Module 3 and 4: Team Selector Team Lead will add members to the project, view evaluations and send emails and requests to teachers.

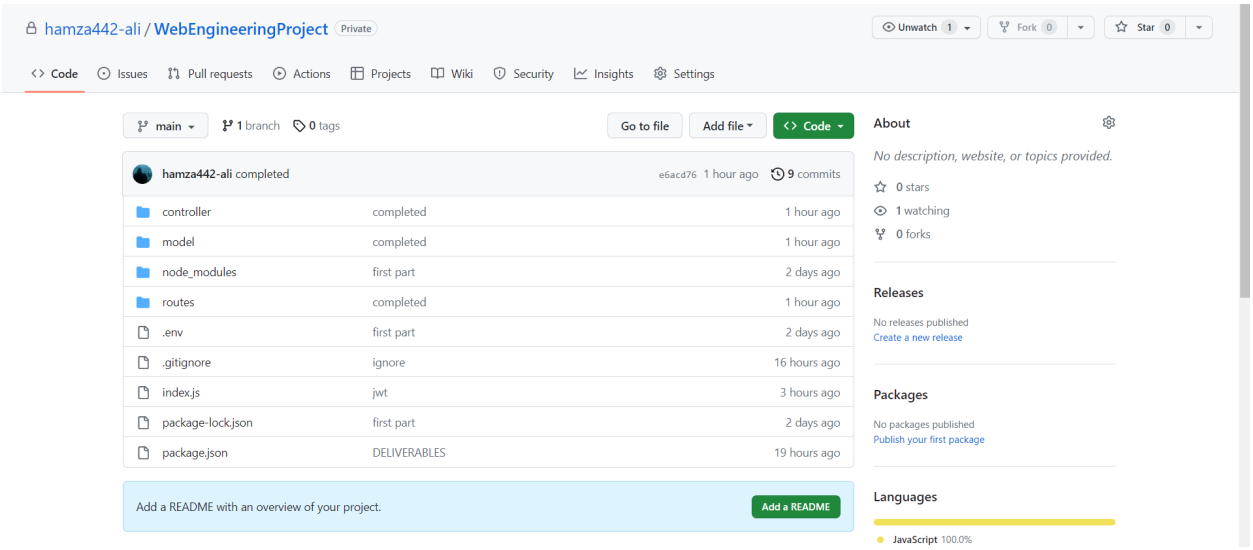
3.1 Add team member 3.2 Assign Role 3.3 Send Email

3.4 Remove member 3.5 View Team Status 3.6 Send Request

3.7 View Supervisor Evaluation 3.8 View All Projects

4: view history, set deadline, create decisions , attach files, set priorities

Ali Hamza:

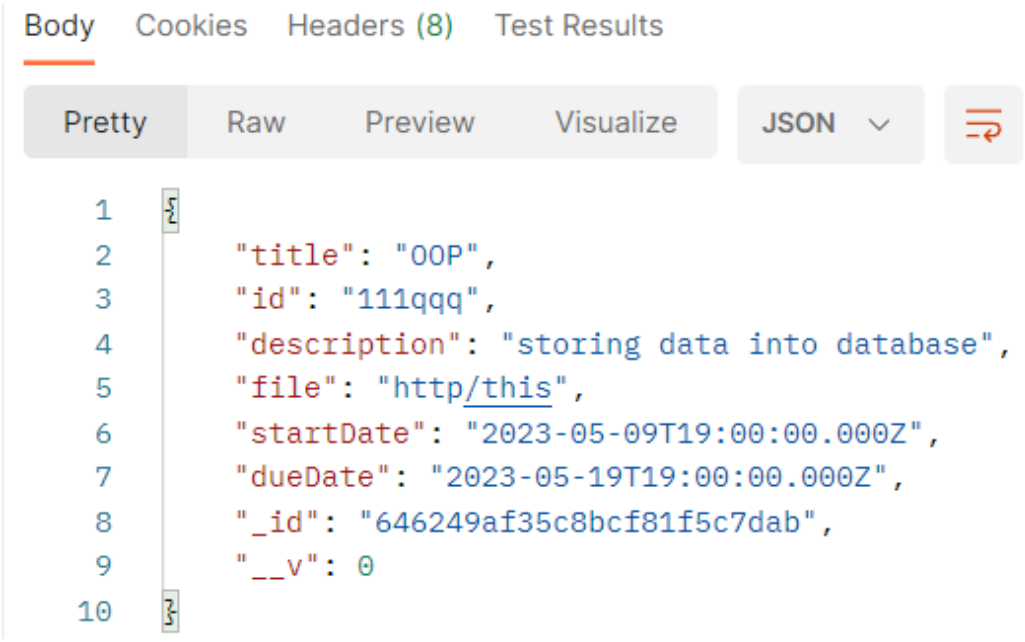


Post man documentation:

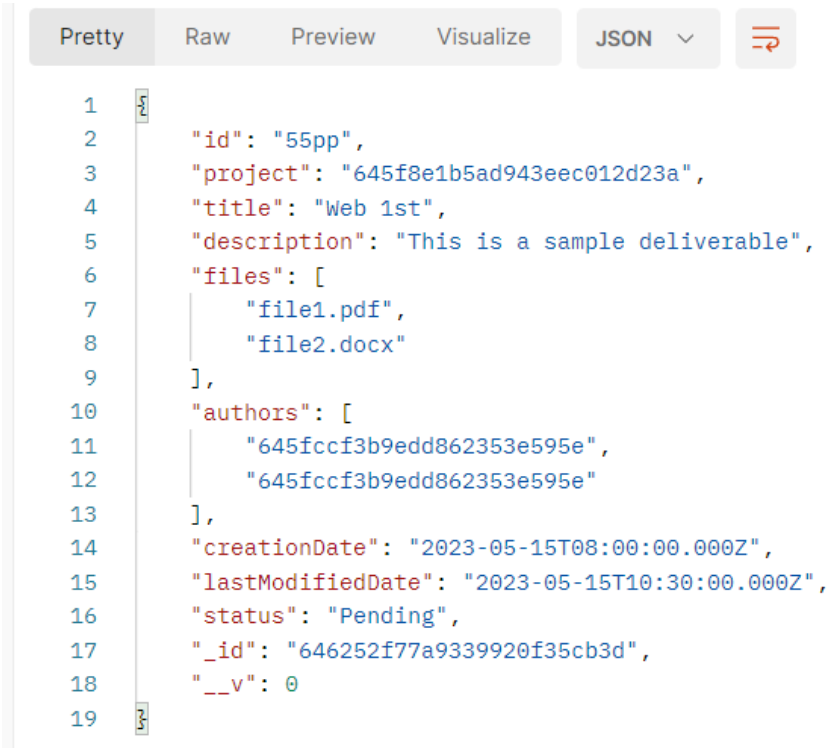
https://documenter.getpostman.com/view/23997045/2s93eeRpbq

- **Sample request and response of the APIs (which you have developed)**

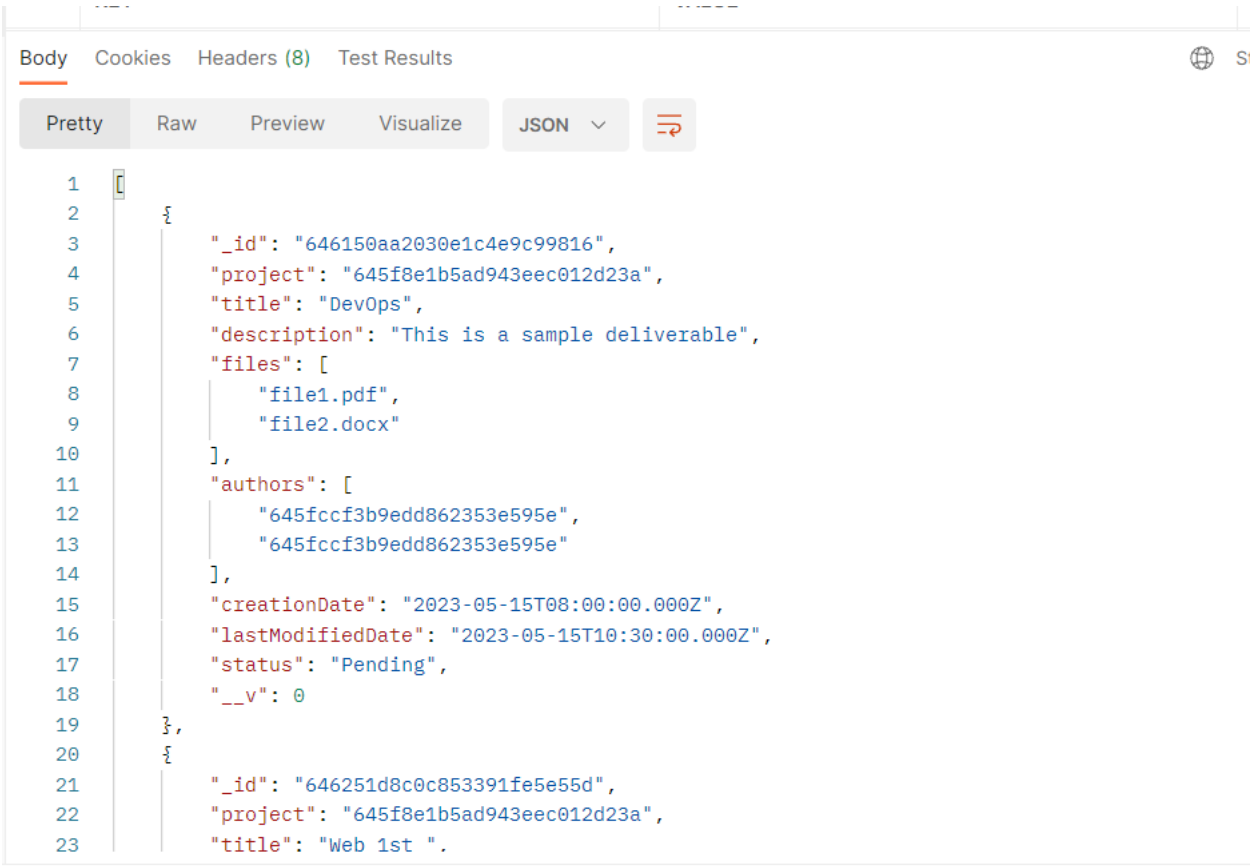
1. http://localhost:3001/projects/createProject



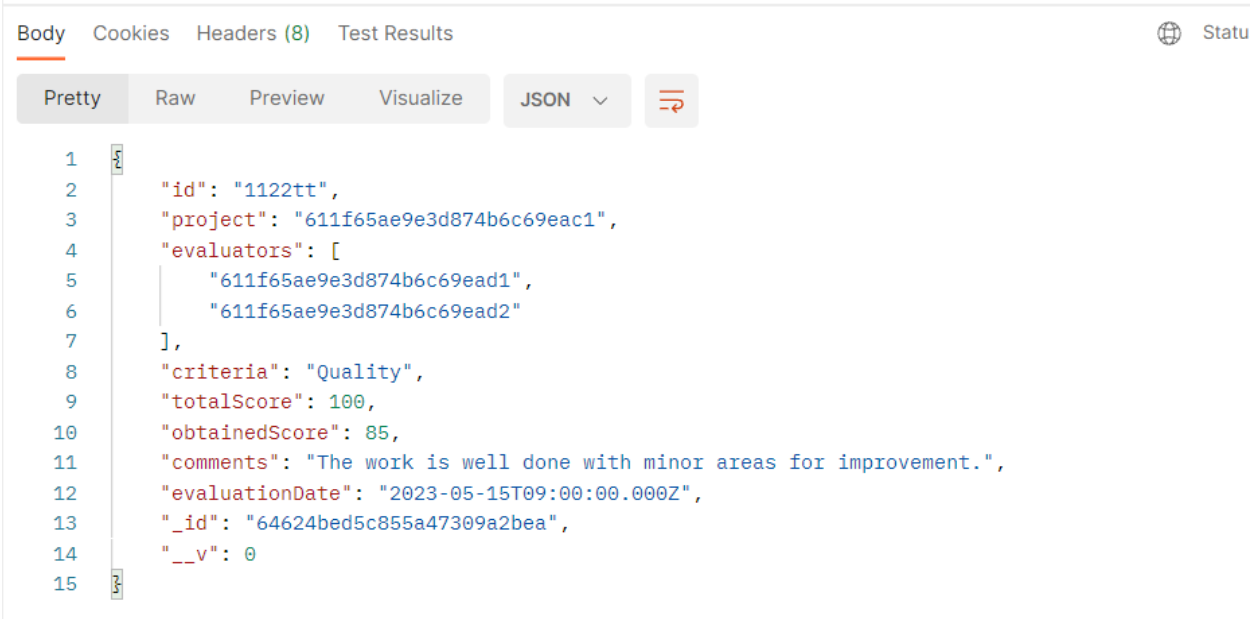
2. http://localhost:3001/deliverables/createDeliverable



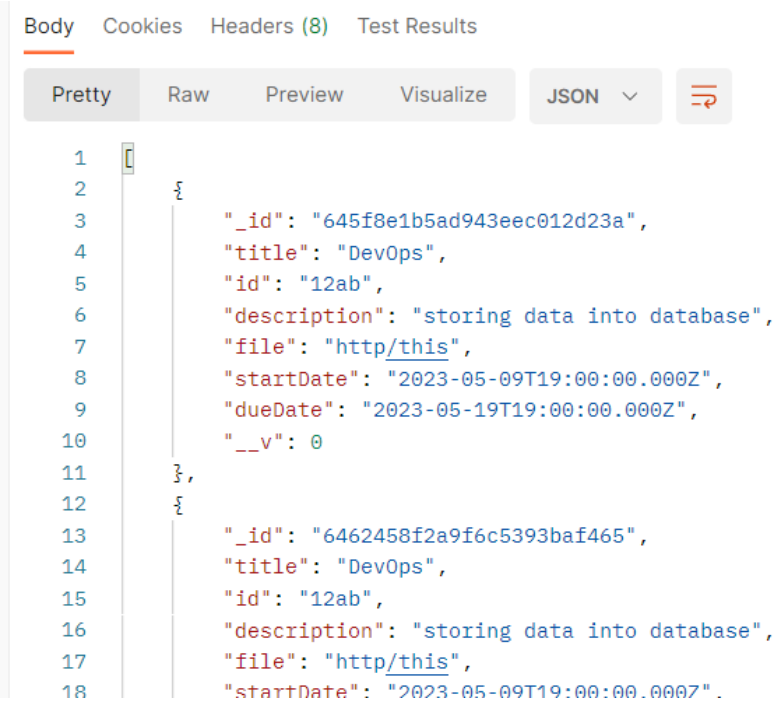
3. http://localhost:3001/deliverables/getAllDeliverables



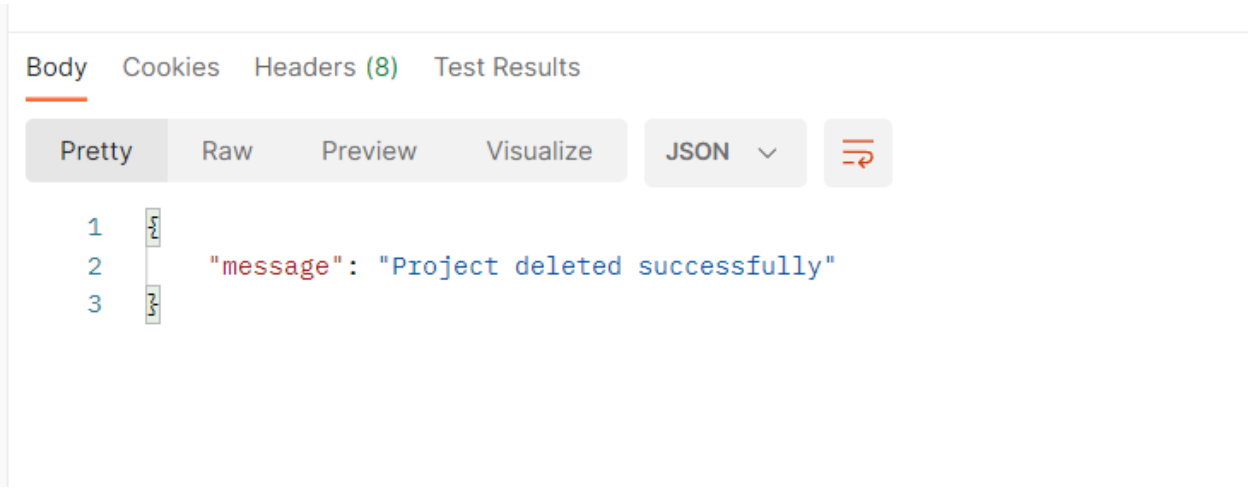
4. http://localhost:3001/evaluations/createEvaluation



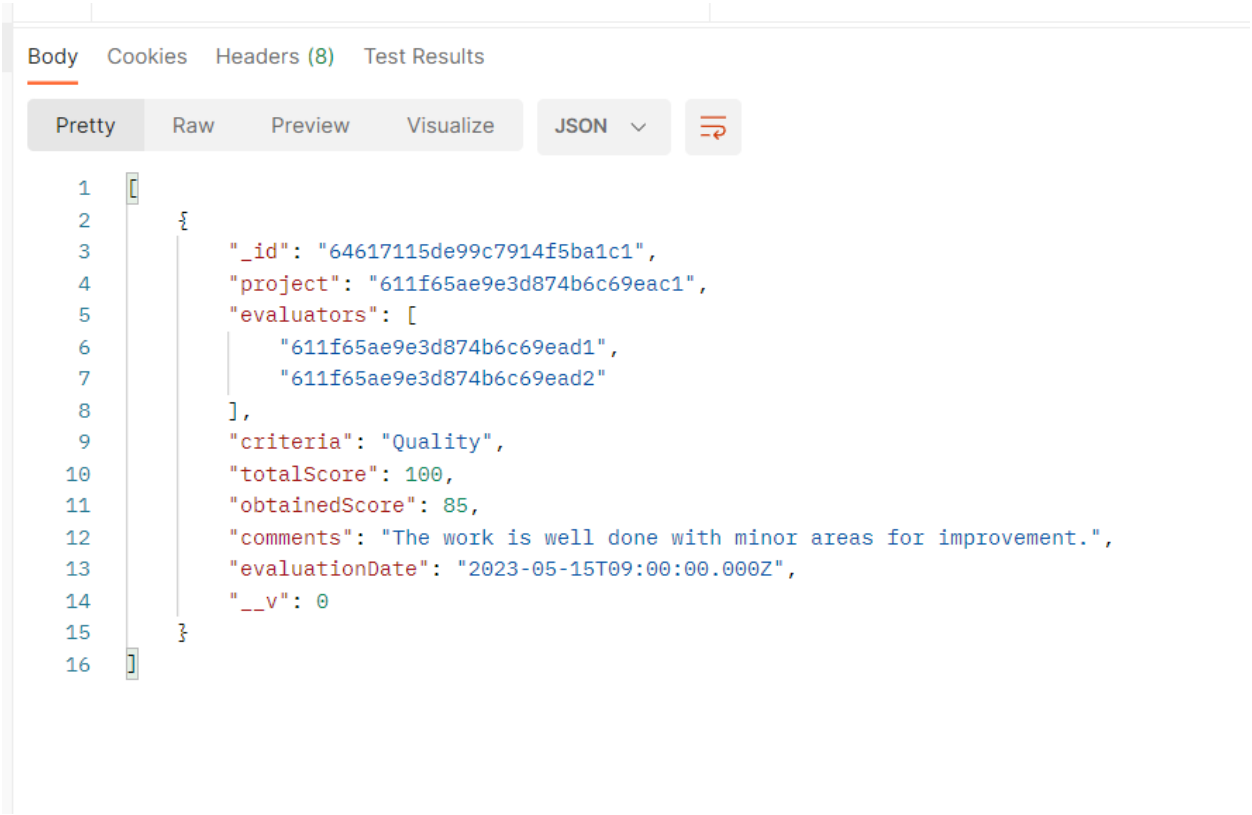
5. http://localhost:3001/users/updateTeamlead/:id?userId=123
6. http://localhost:3001/projects/getAllProjects



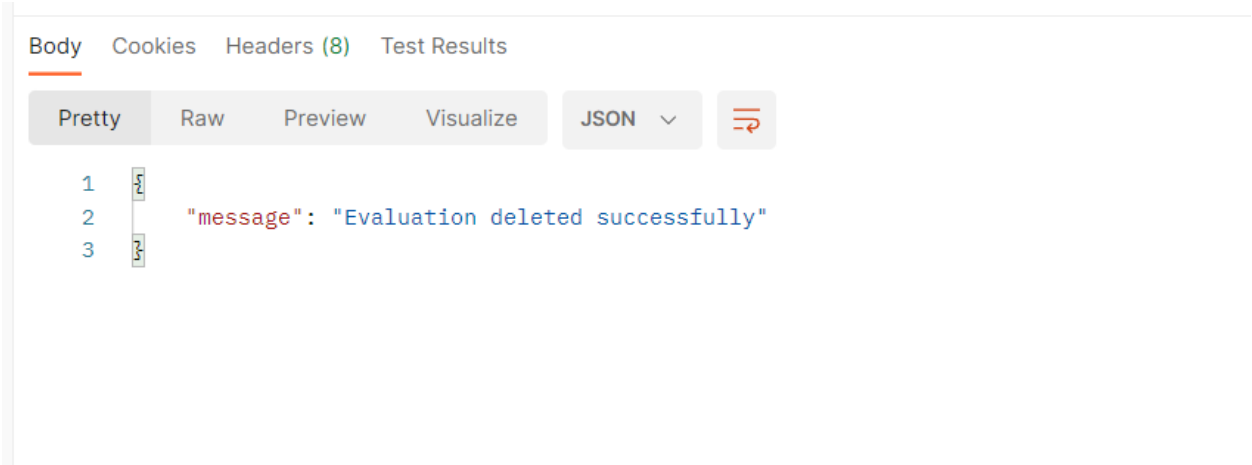
7. http://localhost:3001/projects/deleteProject/:id?id=12qwert



8. http://localhost:3001/evaluations/getAllEvaluations



9. http://localhost:3001/evaluations/deleteEvaluation/:id?id=1122tt



10. http://localhost:3001/deliverables/deleteDeliverable/:id


Body
Cookies
Headers (8)
Test Results
Status: 20
Pretty
Raw
Preview
Visualize
JSON
1 {
2 "message": "Deliverable deleted successfully"
3 }

Deliverable Model:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
const User = require("./userModel"); // Import the User schema file
const Project = require("./projectModel");
const DeliverableSchema = new Schema({
  id: {
    type : String,
    required: true,
  } ,
  project: {
    type: Schema.Types.ObjectId,
    ref: "Project",
    required: true,
  },
  title: {
    type: String,
    required: true,
  },
  description: {
    type: String,
    required: true,
  },
  files: [{
    type: String,
    required: true,
  }],
  authors: [{
    type: Schema.Types.ObjectId,
    ref: "User",
    required: true,
  }],
  creationDate: {
    type: Date,
    default: Date.now,
  },
  lastModifiedDate: {
    type: Date,
    default: Date.now,
  },
  status: {
    type: String,
    required: true,
  },
});
```

Evaluation Schema:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
const EvaluationSchema = new Schema({
  id: {
    type:String,
    required: true,
  },
  project: {
    type: Schema.Types.ObjectId,
    ref: "Project",
    required: true,
  },
  evaluators: [{
    type: Schema.Types.ObjectId,
    ref: "User",
    required: true,
  }],
  criteria: {
    type: String,
    required: true,
  },
  totalScore:{
    type: Number,
    require: true
  },
  obtainedScore: {
    type: Number,
    required: true,
  },
  comments: {
    type: String,
    required: true,
  },
  evaluationDate: {
    type: Date,
    default: Date.now,
  },
});
```

Project Model:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
const ProjectSchema = new Schema({
  title: {
    type: String,
    required: true,
  },
  id: {
    type: String,
    required: true,
  },
  description: {
    type: String,
    required: true,
  },
  file: {
    type: String,
    required: true,
  },
  startDate: {
    type: Date,
    default: Date.now,
  },
  dueDate: {
    type: Date,
    default: () => {
      const currentDate = new Date();
      currentDate.setDate(currentDate.getDate() + 5);
      return currentDate;
    },
  }
});
```

User Schema:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
const UserSchema = new Schema ({
    userId: {
        type: String,
        required: true
    },

    name: {
        type: String,
        required: true
    },
    email: {
        type: String,
        required: true
    },
    contact: {

        type: String,
        required: true
    },
    password: {
        type: String,
        required: true
    },
    gender: {
        type: String,
        required: true
    },
    profile_picture: {
        type: String,
        required: true
    },
    registerAs:{
        type: String,
        enum: ['student', 'teacher', 'admin'],
        required: true
    }
});
```

Mahad Rahat:

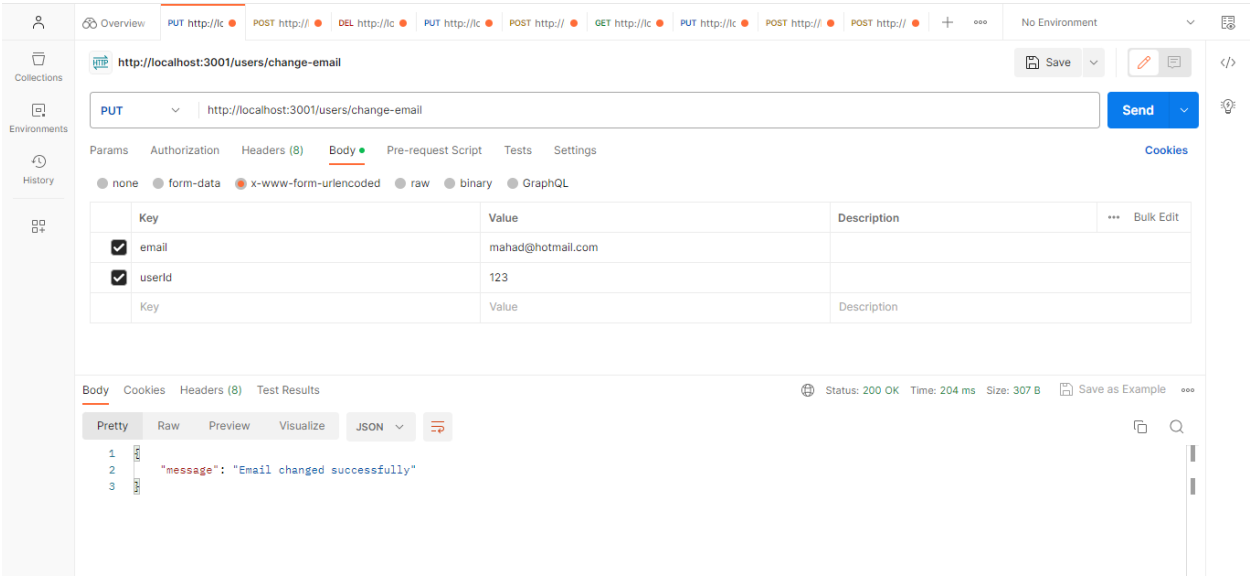
- **Postman documentation link of each member**

https://documenter.getpostman.com/view/27400726/2s93eeRpkf

- Sample request and response of the APIs (which you have developed)

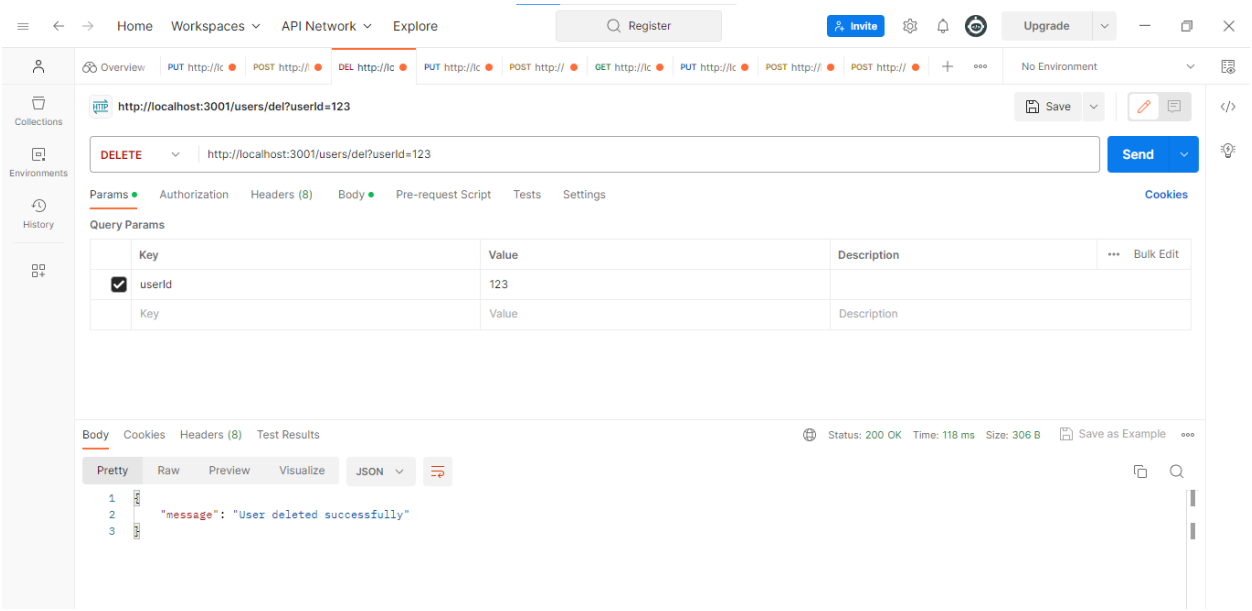
1. Changing email of a user

   http://localhost:3001/users/change-email



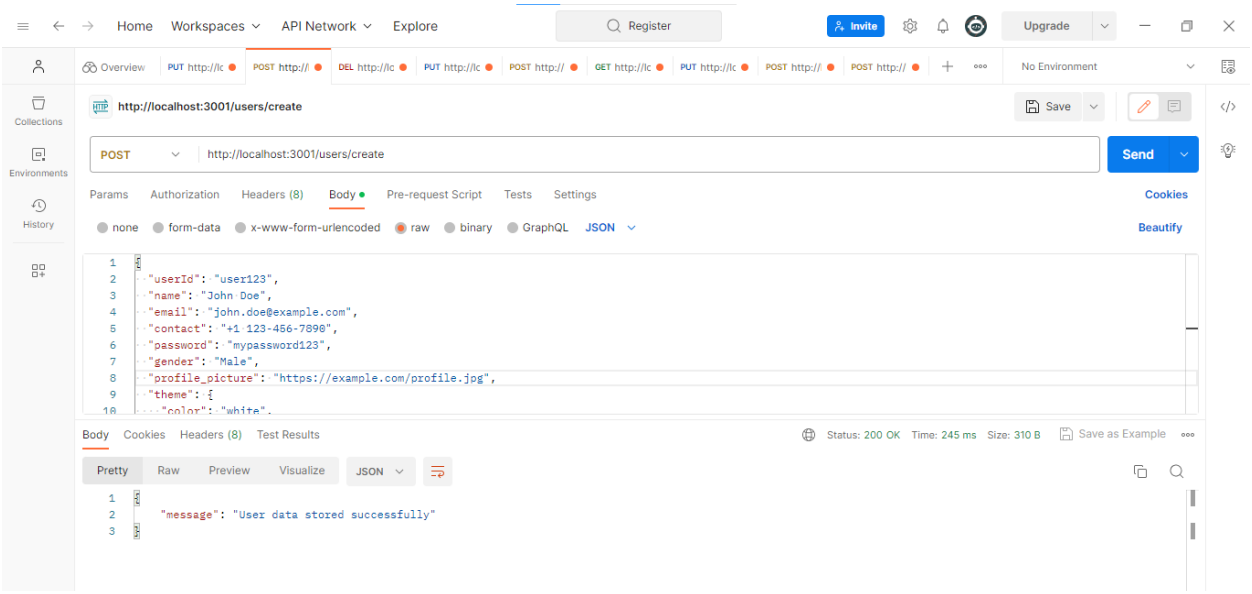
2. Deletion of a user

   http://localhost:3001/users/del?userId=123



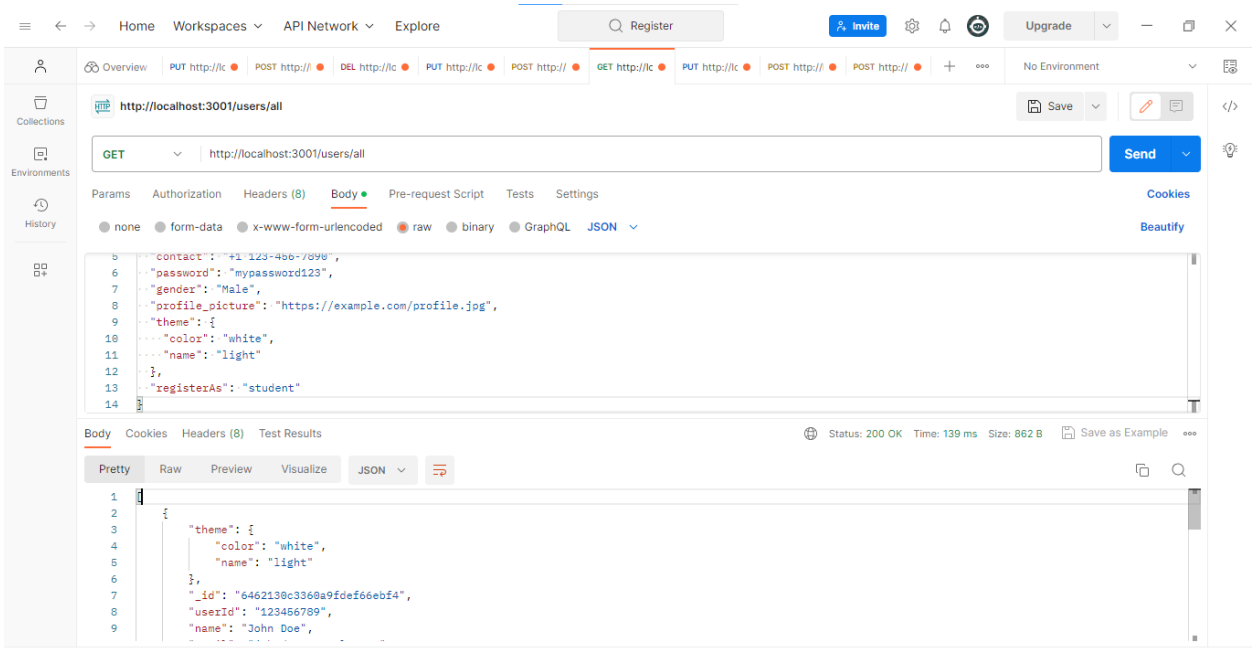
3. User Creation:

http://localhost:3001/users/create



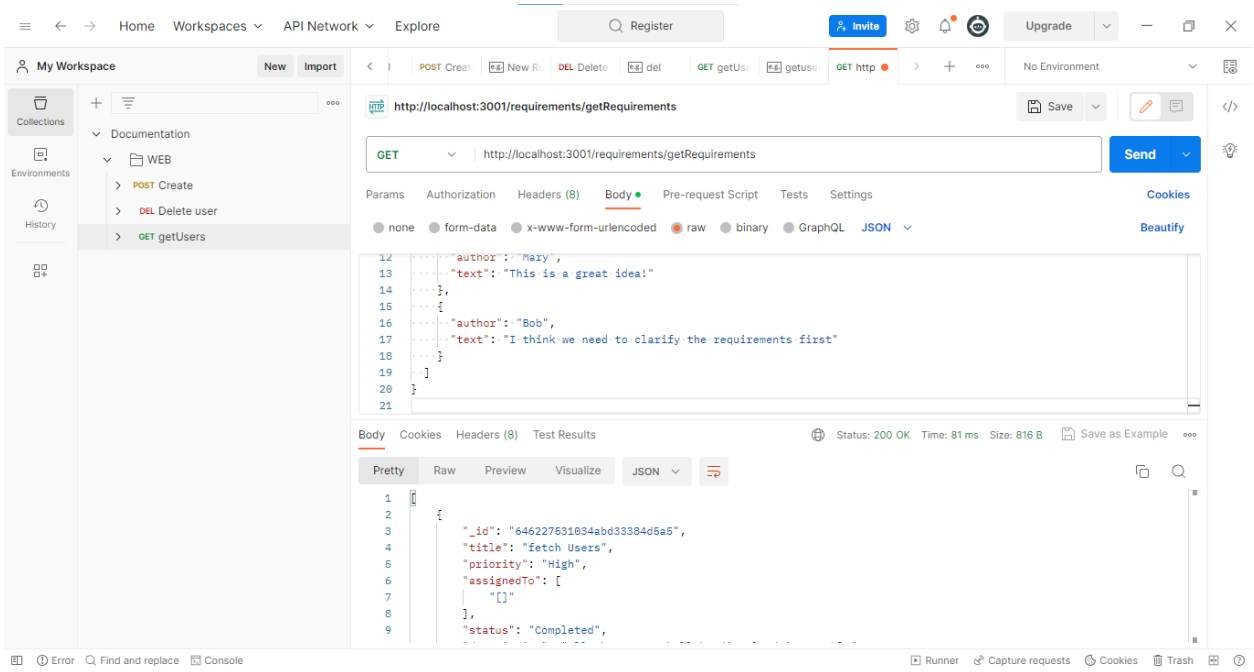
4. Getting all the users

http://localhost:3001/users/all

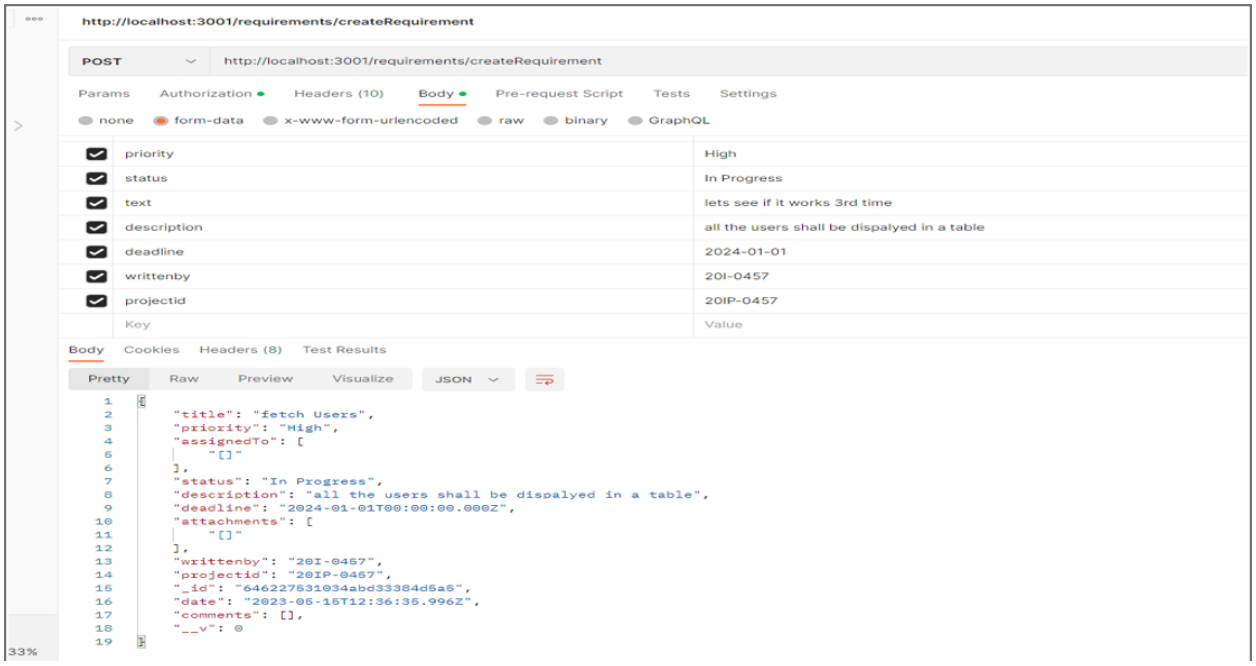


5. Getting all requirements

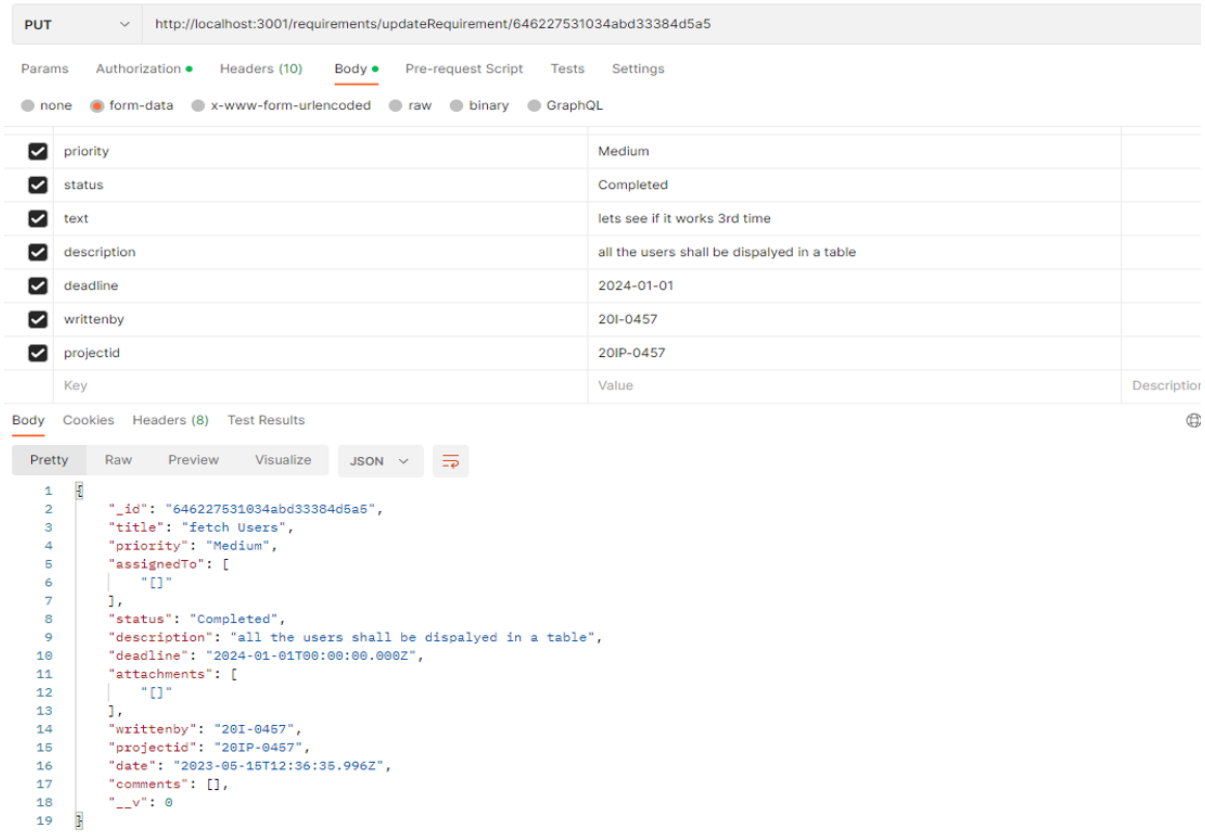
http://localhost:3001/requirements/getRequirements



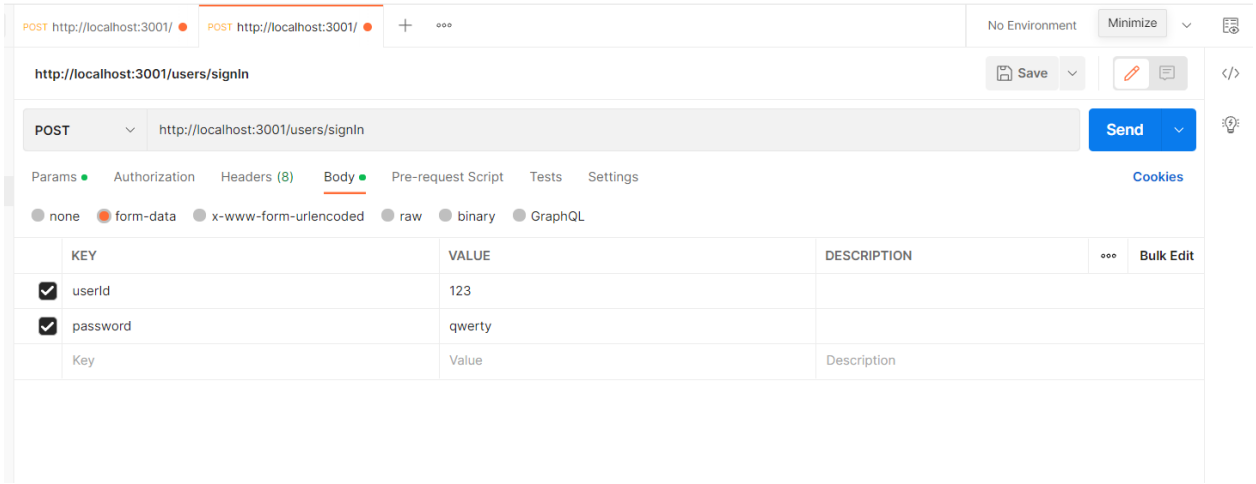
6: Creating Requirements:



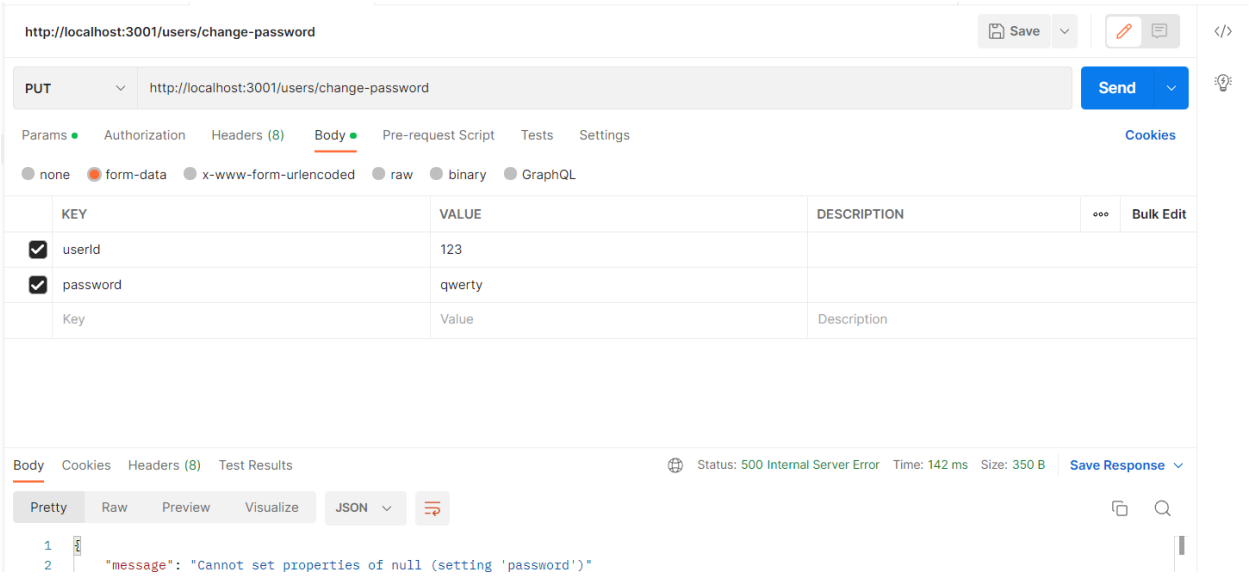
7. Update requirements



8: SignIn with jwt

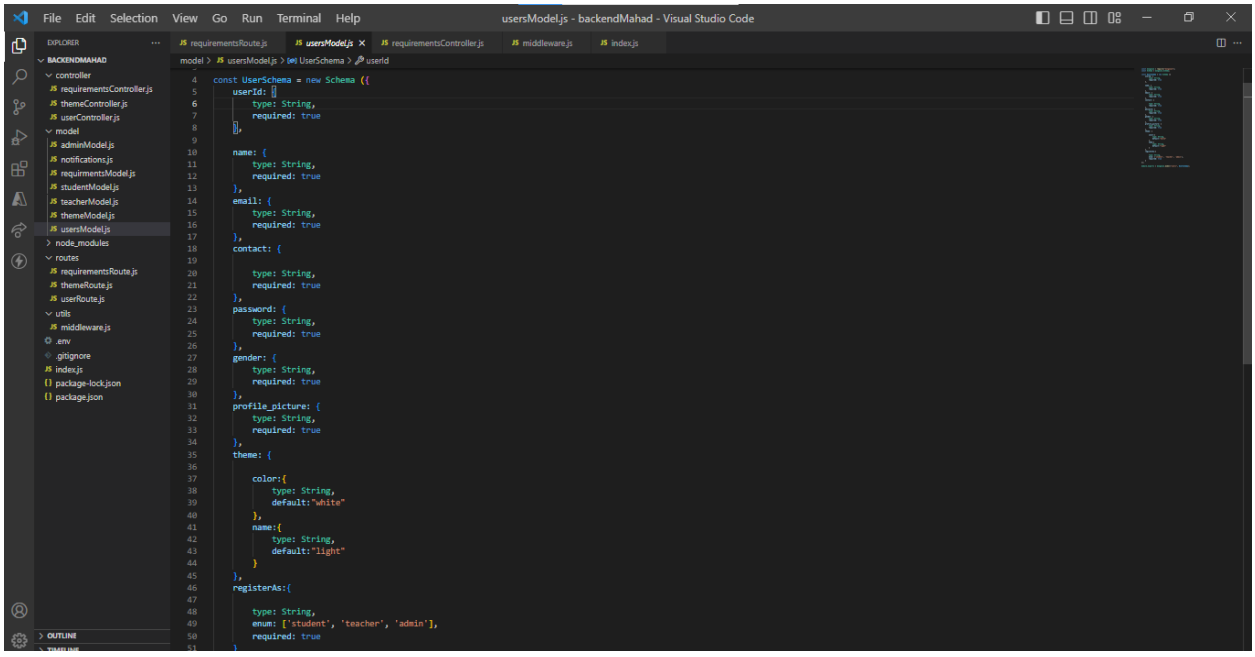


9. Change Password:

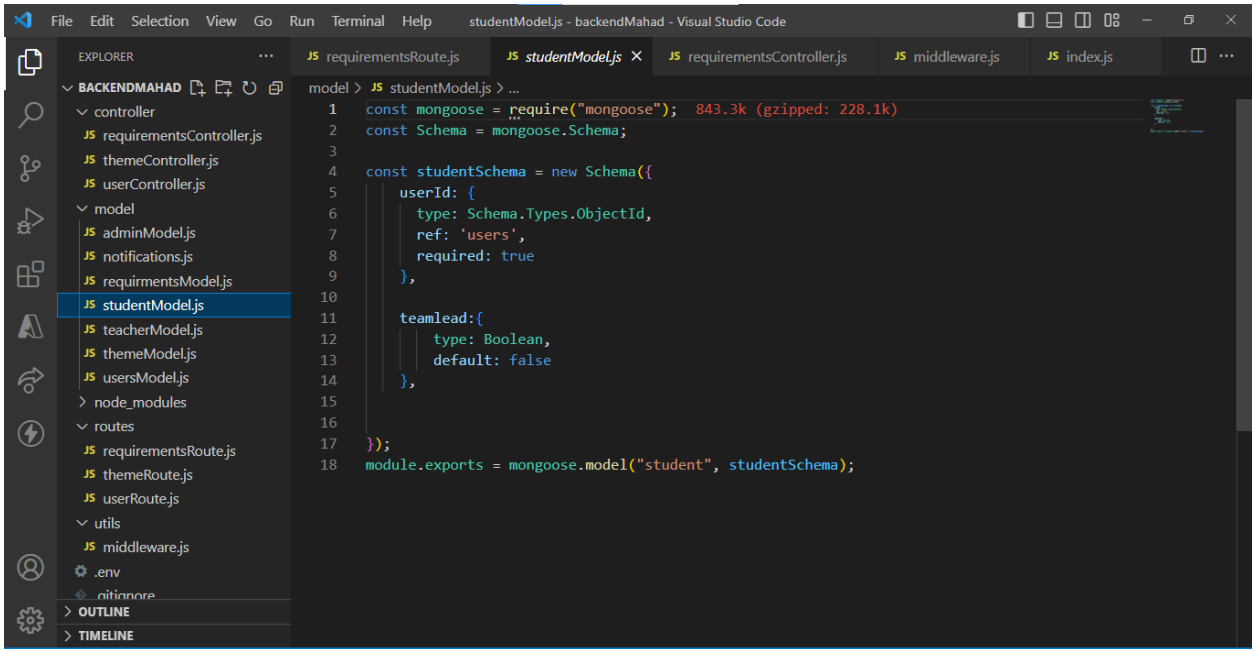


• Database Schemas.

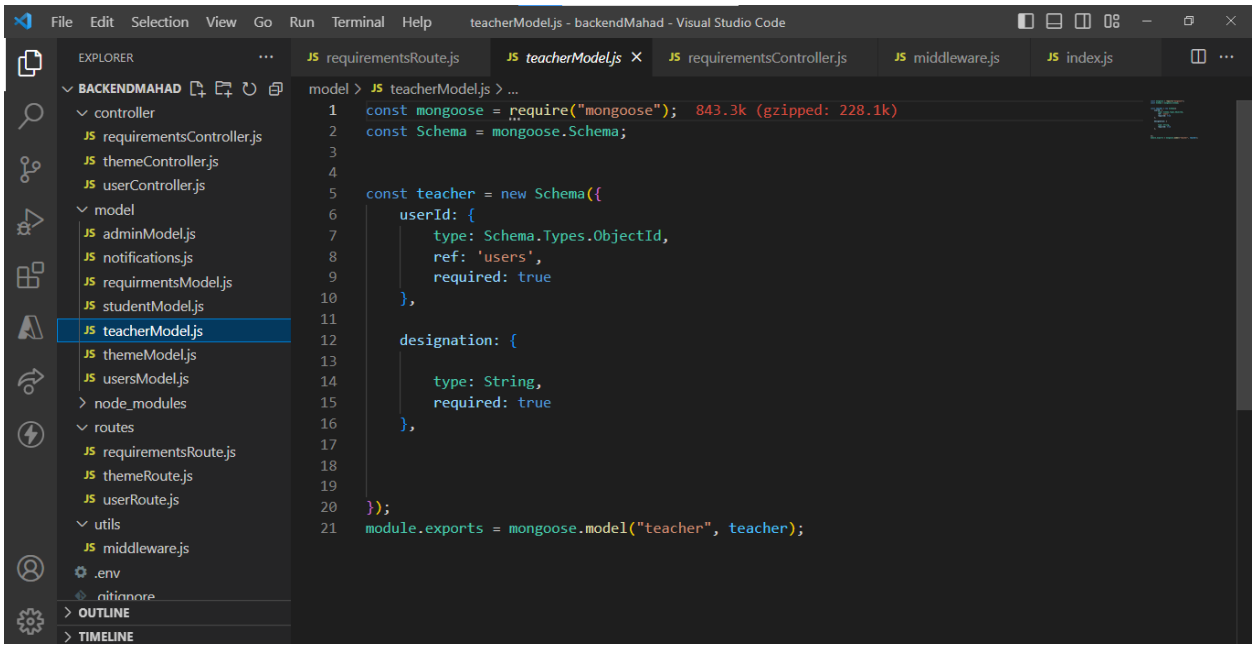
1. User Schema:



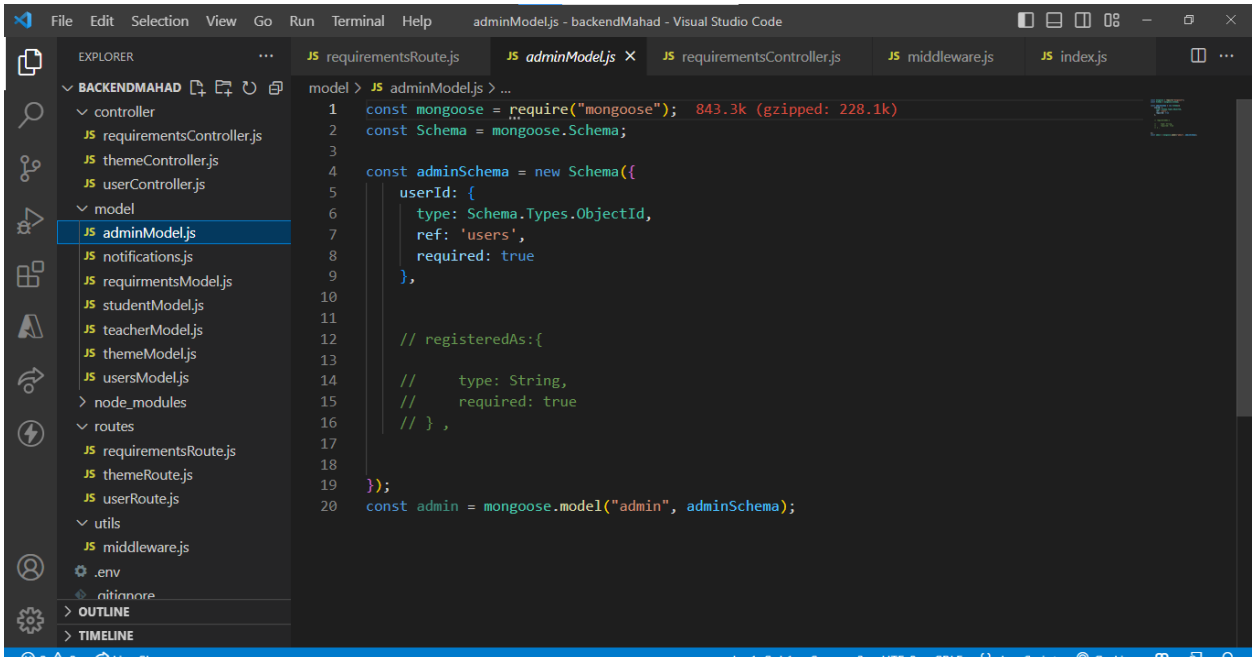
2. Student Schema:



3. Teacher Schema:



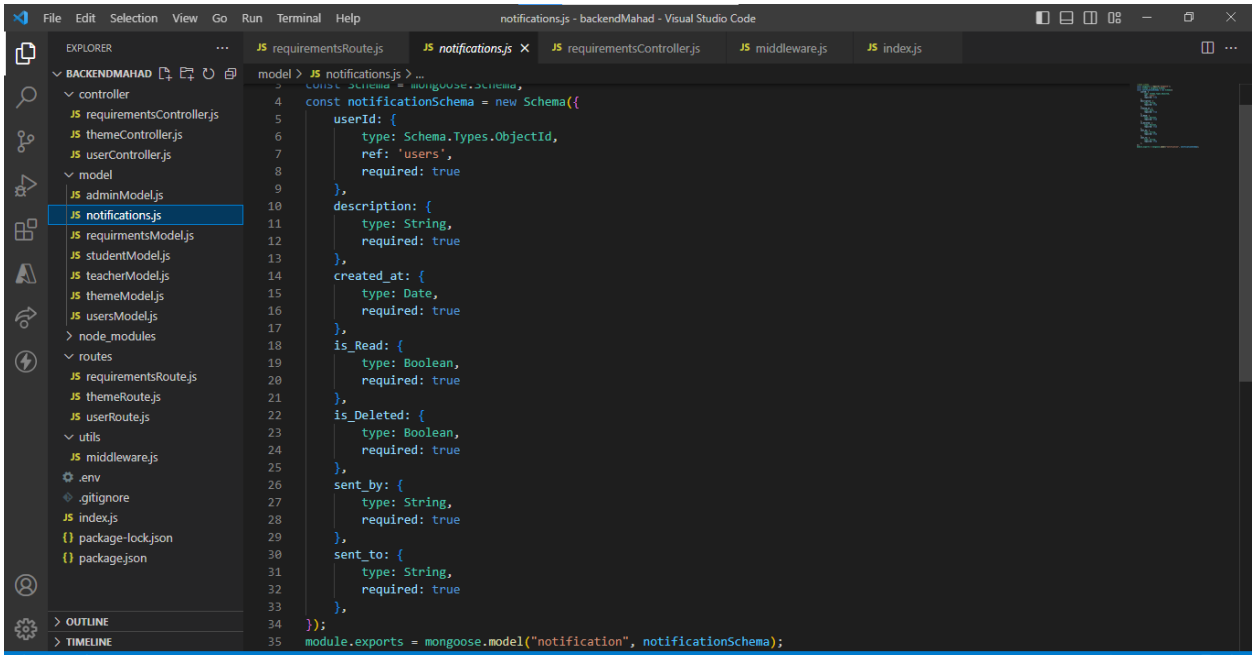
4. Admin Schema:



5. Requirement Schema:

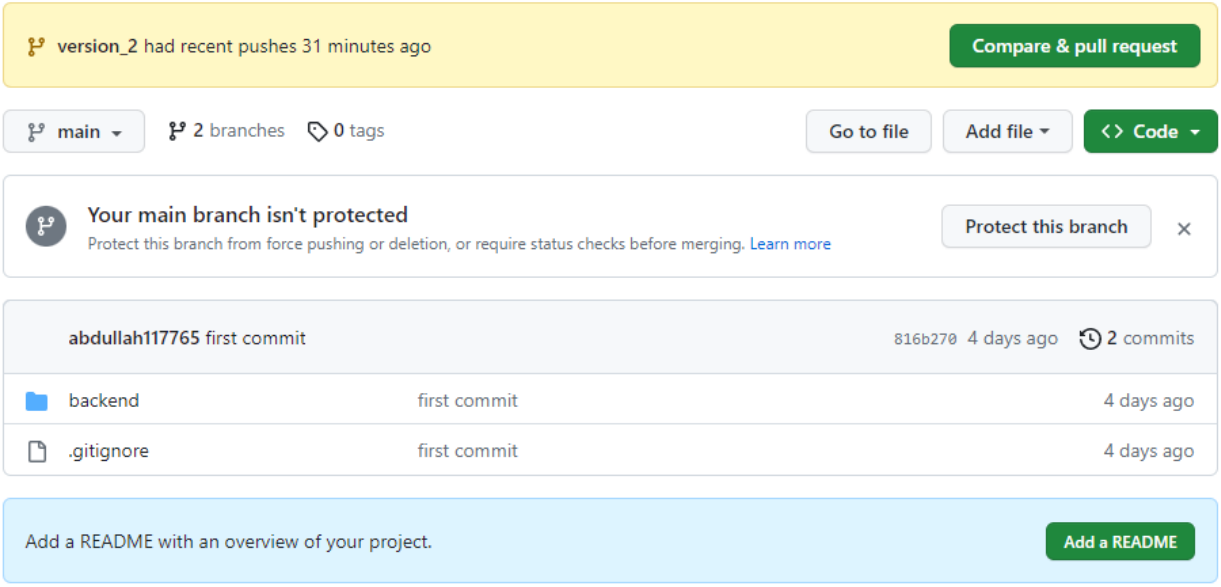
```
const commentSchema = new mongoose.Schema({
  content: {
    type: String,
    required: true,
  },
  createdBy: {
    type: String,
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
});

const requirementSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true,
  },
  priority: {
    type: String,
    enum: ['Low', 'Medium', 'High'],
    default: 'Low',
  },
  assignedTo: {
    type: Array,
    default:"[]"
  },
  status: {
    type: String,
    enum: ['Pending', 'In Progress', 'Completed'],
    default: 'Pending',
  },
  date: {
    type: Date,
    default: Date.now,
  },
  description: {
    type: String,
    required: true,
  },
  deadline: {
    type: Date,
    required: true,
  },
  attachments: {
    type: Array,
    default:"[]"
  },
  comments:{
        type: [commentSchema],

  }
});

const Requirement = mongoose.model('Requirement', requirementSchema);
```

7. Notification Schema:



Mian Abdullah:

Github Repository



https://github.com/abdullah117765/WebProject.git

Models

```
const mongoose = require('mongoose');  841.1k (gzipped: 227.5k)

const teamMemberSchema = new mongoose.Schema({
  student_id: {
    type: String,
    required: true,
    unique: true,
  },
  team_lead_id: {
    type: String,
    required: true,
  },
  student_name: {
    type: String,
    required: true,
  },
  student_email: {
    type: String,
    required: true,
    unique: true,
  },
  student_status: {
    type: String,
    required: true,
  },
  requirements: {
      type: Array,
    },
    projectid: {
      type: String,
      required: true,
    },
    student_role: {
      type: String,
      default:" "
    }

});

const TeamMember = mongoose.model('TeamMember', teamMemberSchema);
```

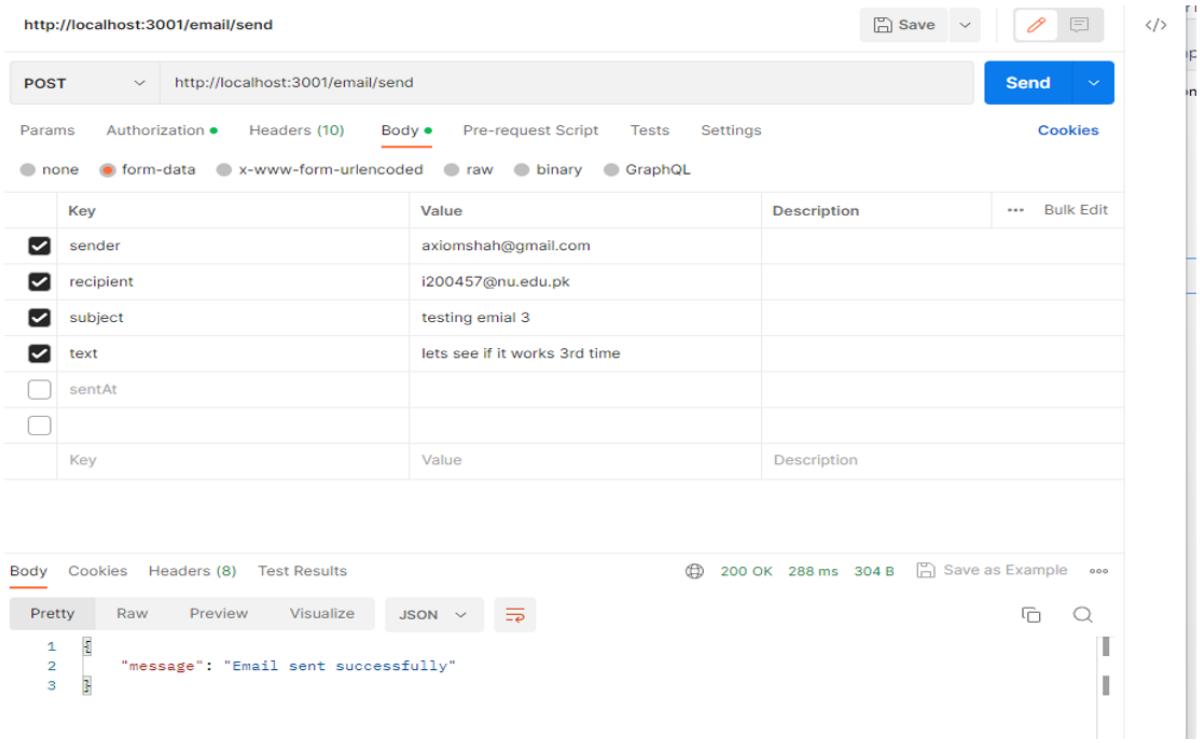
```
const mongoose = require('mongoose');  841.1k (gzipped: 227.5k)

const commentSchema = new mongoose.Schema({
  content: {
    type: String,
    required: true,
  },
  createdBy: {
    type: String,
  },
  createdAt: {
    type: Date,
    default: Date.now,
  },
});

const requirementSchema = new mongoose.Schema({


  title: {
    type: String,
    required: true,
  },
  priority: {
    type: String,
    enum: ['Low', 'Medium', 'High'],
    default: 'Low',
  },
  assignedTo: {
    type: Array,
    default:"[]"
  },
  status: {
    type: String,
    enum: ['Pending', 'In Progress', 'Completed'],
    default: 'Pending',
  },
```

```
  },
  date: {
    type: Date,
    default: Date.now,
  },
  description: {
    type: String,
    required: true,
  },
  deadline: {
    type: Date,
    required: true,
  },
  attachments: {
    type: Array,
    default:"[]"
  },
  comments:{
        type: [commentSchema],

  } ,
  writtenby: {//roll number
    type: String,
    required: true,
  },
  projectid: {// unique project id
    type: String,
    required: true,
  }
});
```

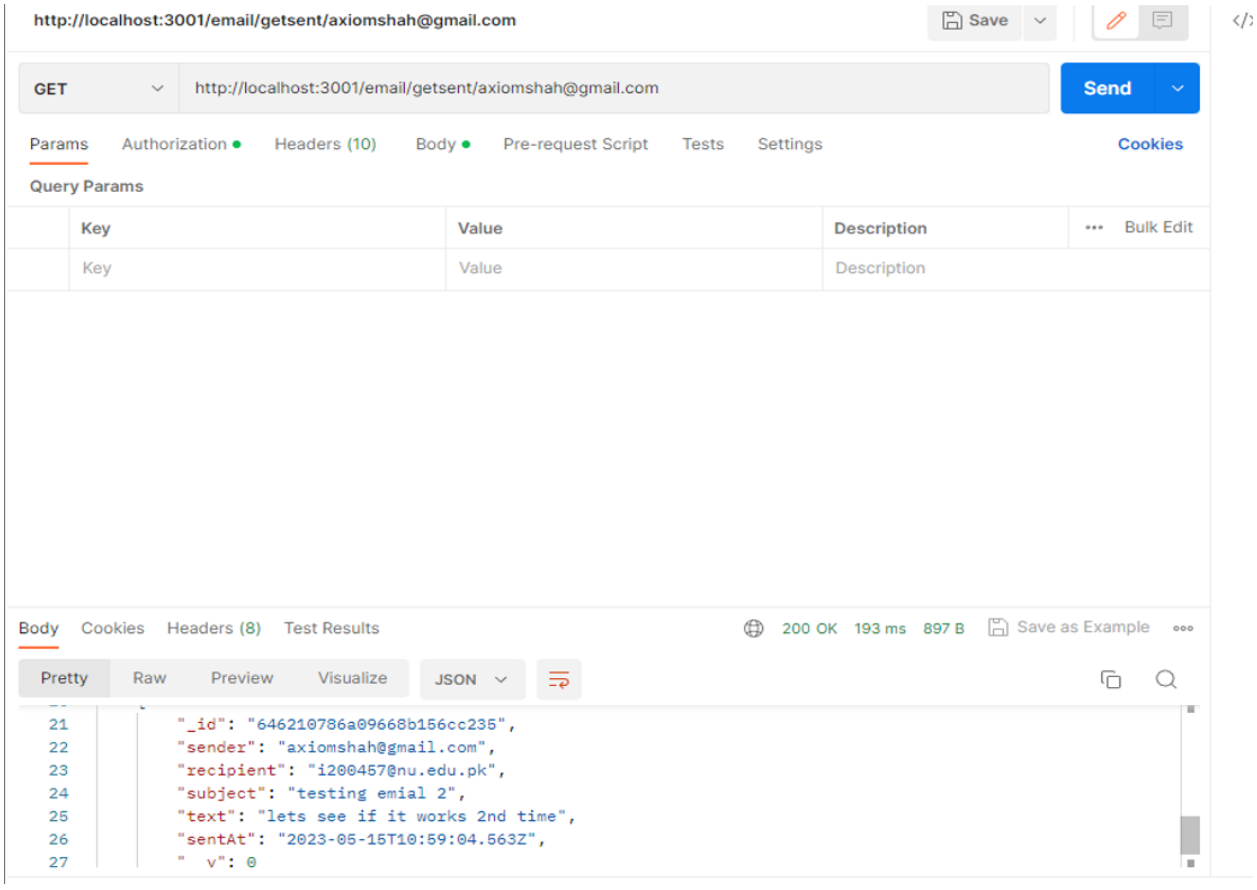
```
const mongoose = require('mongoose');  841.1k (gzipped: 227.5k)

const emailSchema = new mongoose.Schema({
  sender: {
    type: String,
    required: true,
  },
  recipient: {
    type: String,
    required: true,
  },
  subject: {
    type: String,
    required: true,
  },
  text: {
    type: String,
    required: true,
  },
  sentAt: {
    type: Date,
    default: Date.now,
  },
});

const Email = mongoose.model('Email', emailSchema);

module.exports = Email;
```

Email Module

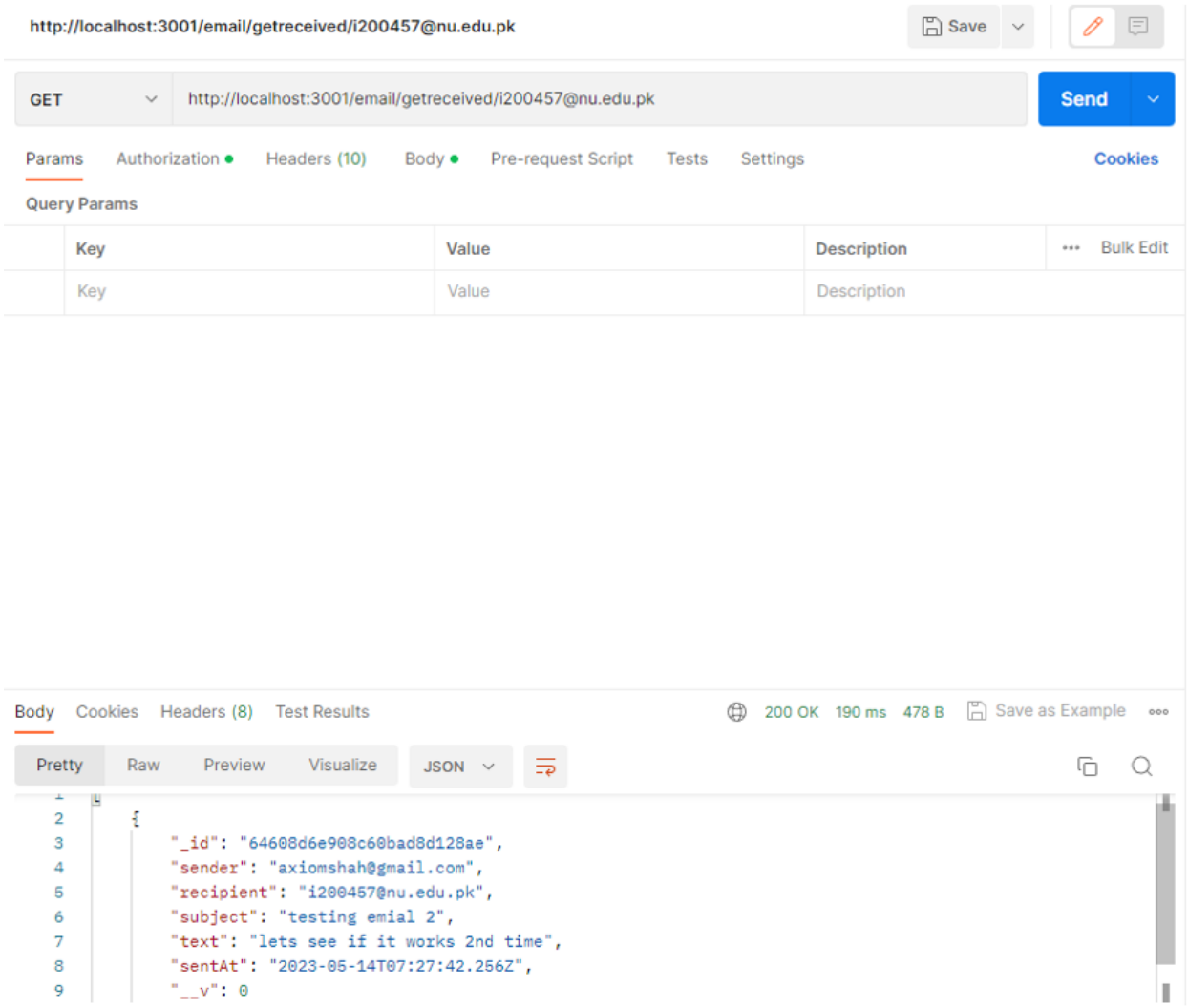
Send an email to a particular person



Get all emails that were send by a particular user

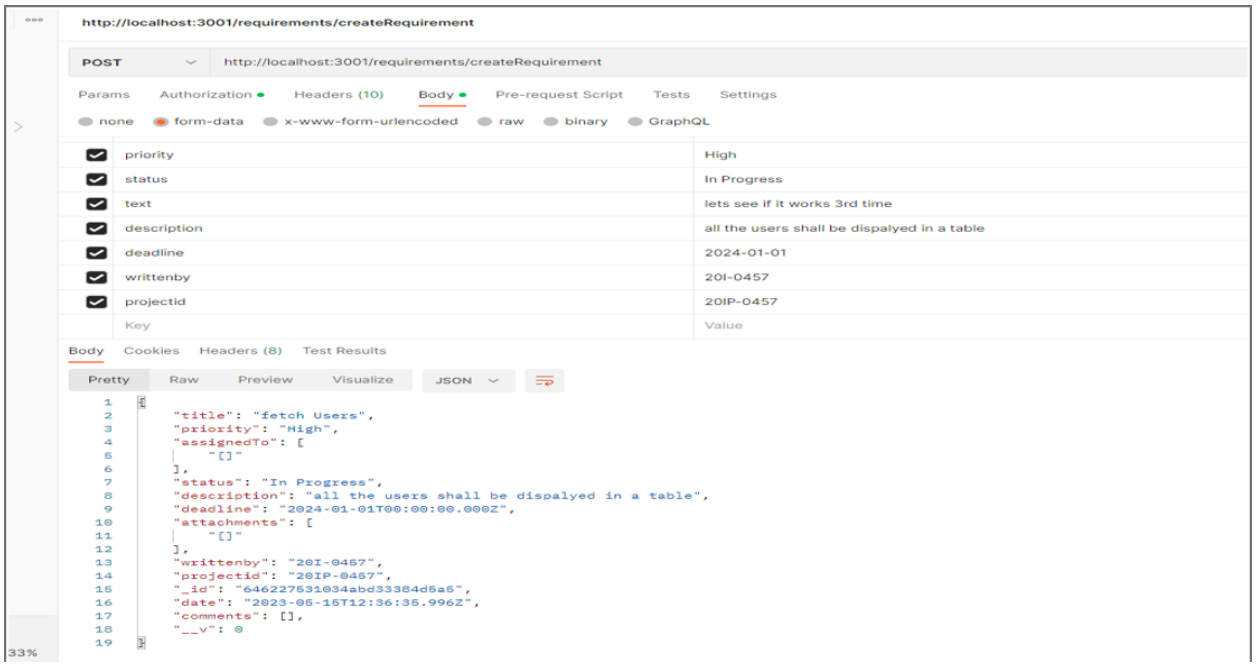


Get all emails that were receivied by a particular user

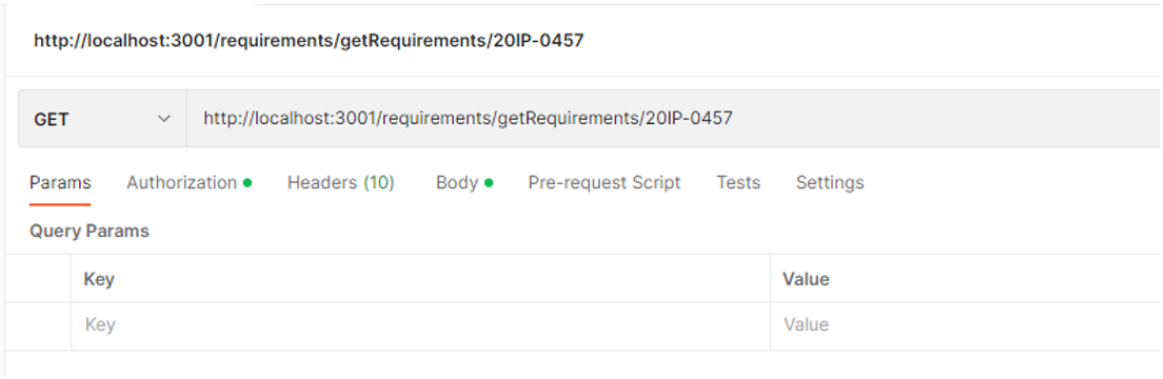


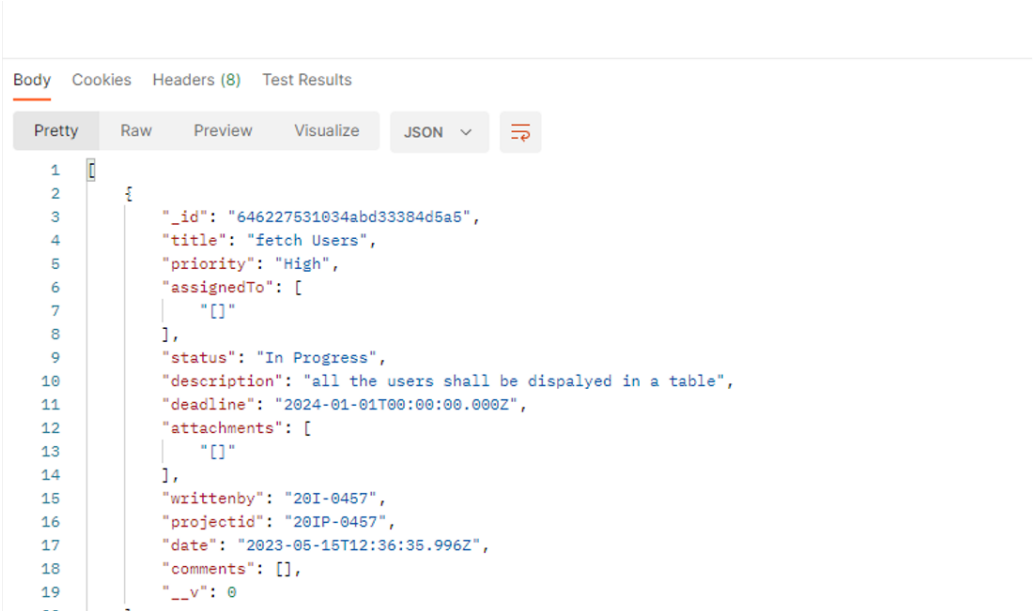
Requirement Module

Create a Requirement



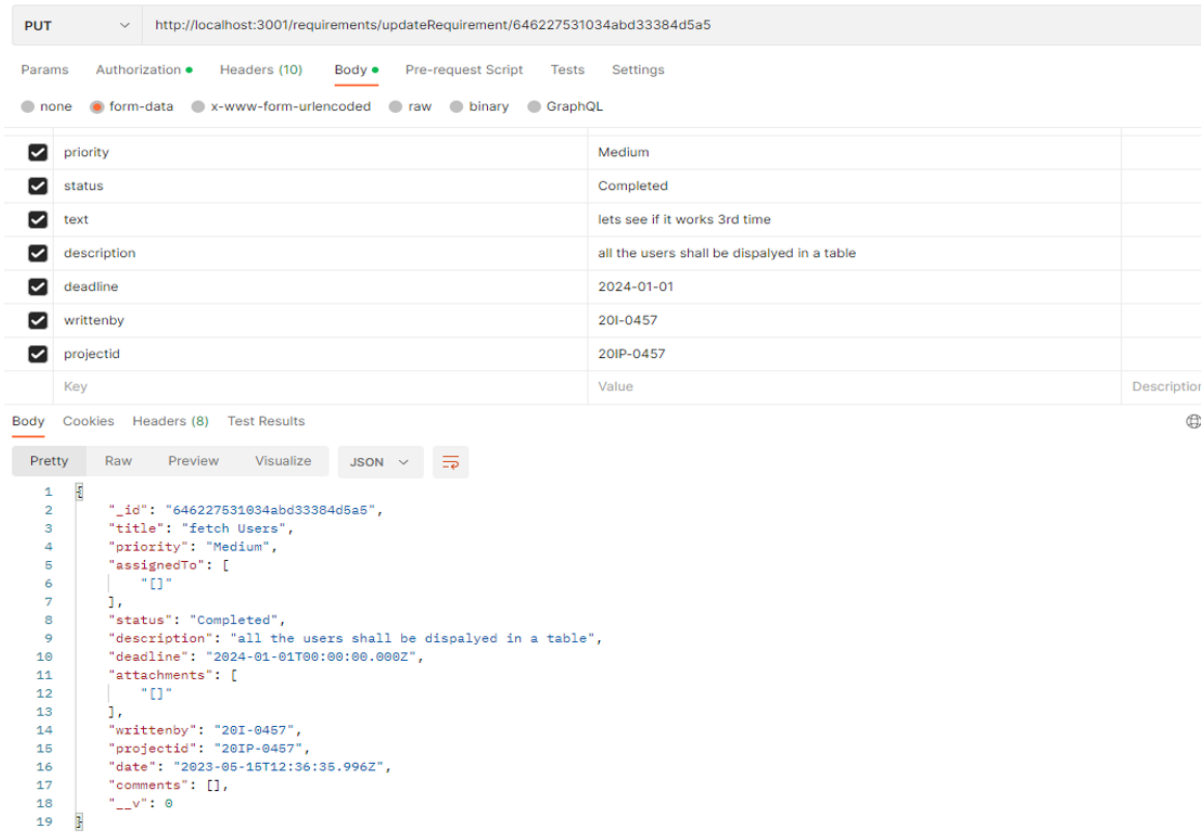
Find requirement by Projectid



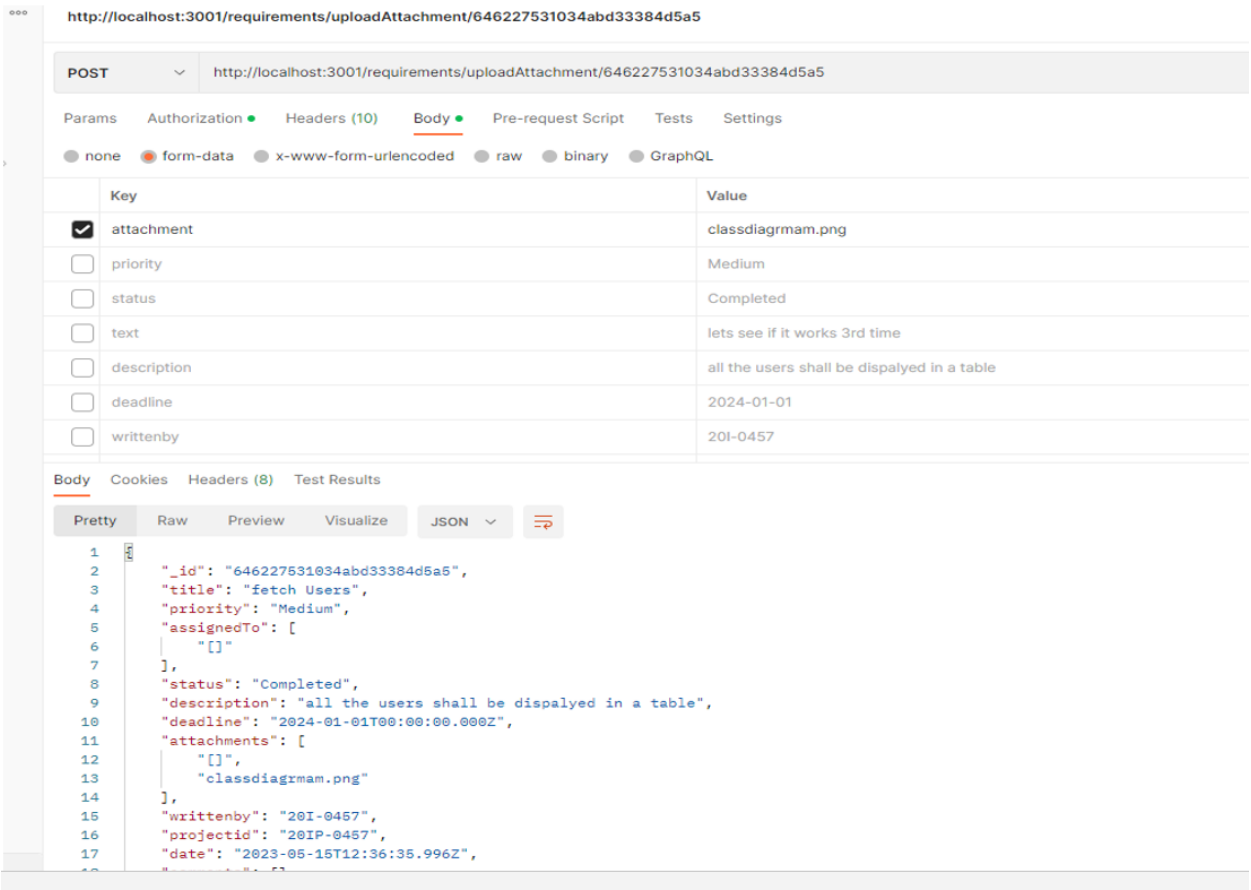


update requirement

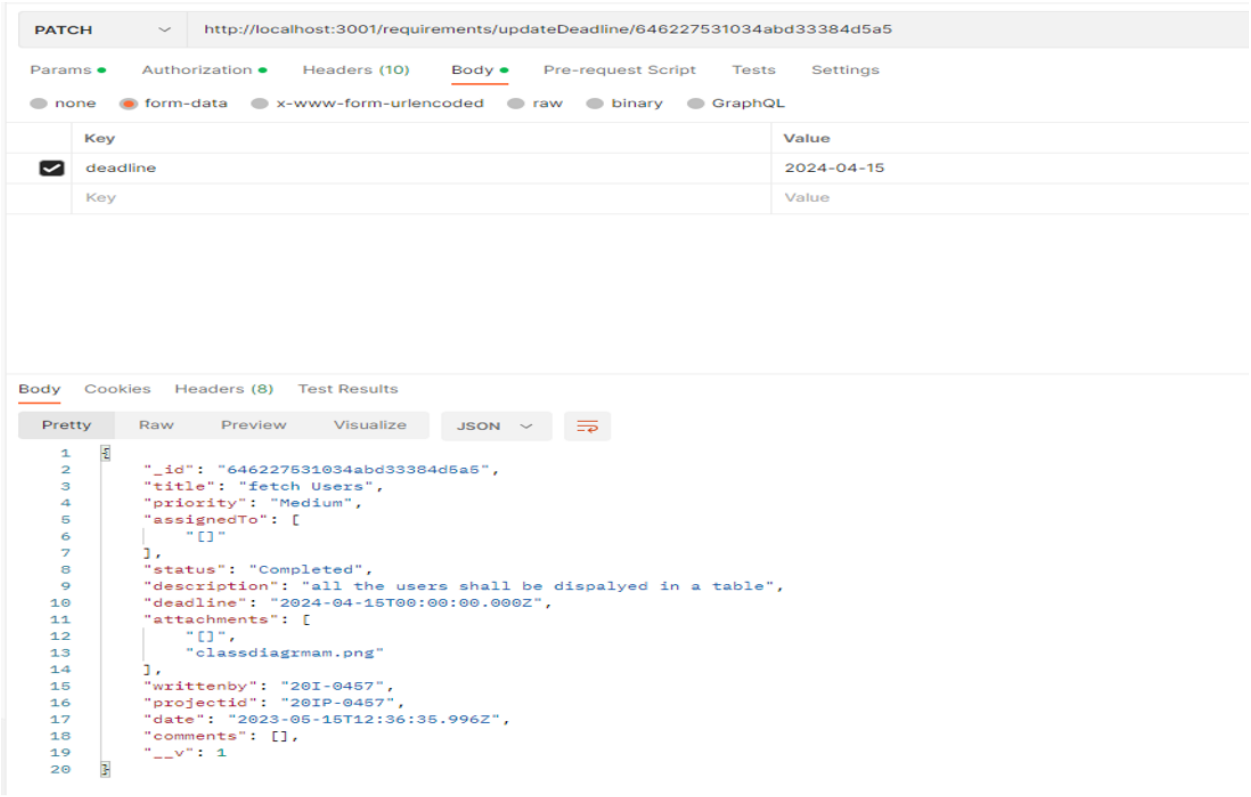
_Updating the priority to Medium and status to Complete


PUT
http://localhost:3001/requirements/updateRequirement/646227531034abd33384d5a5
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
priority
Medium
status
Completed
text
lets see if it works 3rd time
description
all the users shall be dispalyed in a table
deadline
2024-01-01
writtenby
20I-0457
projectid
20IP-0457
Key
Value
Description
Body
Cookies
Headers (8)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "646227531034abd33384d5a5",
"title": "fetch Users",
"priority": "Medium",
"assignedTo": [
"[]"
],
"status": "Completed",
"description": "all the users shall be dispalyed in a table",
"deadline": "2024-01-01T00:00:00.000Z",
"attachments": [
"[]"
],
"writtenby": "20I-0457",
"projectid": "20IP-0457",
"date": "2023-05-15T12:36:35.996Z",
"comments": [],
"__v": 0
}

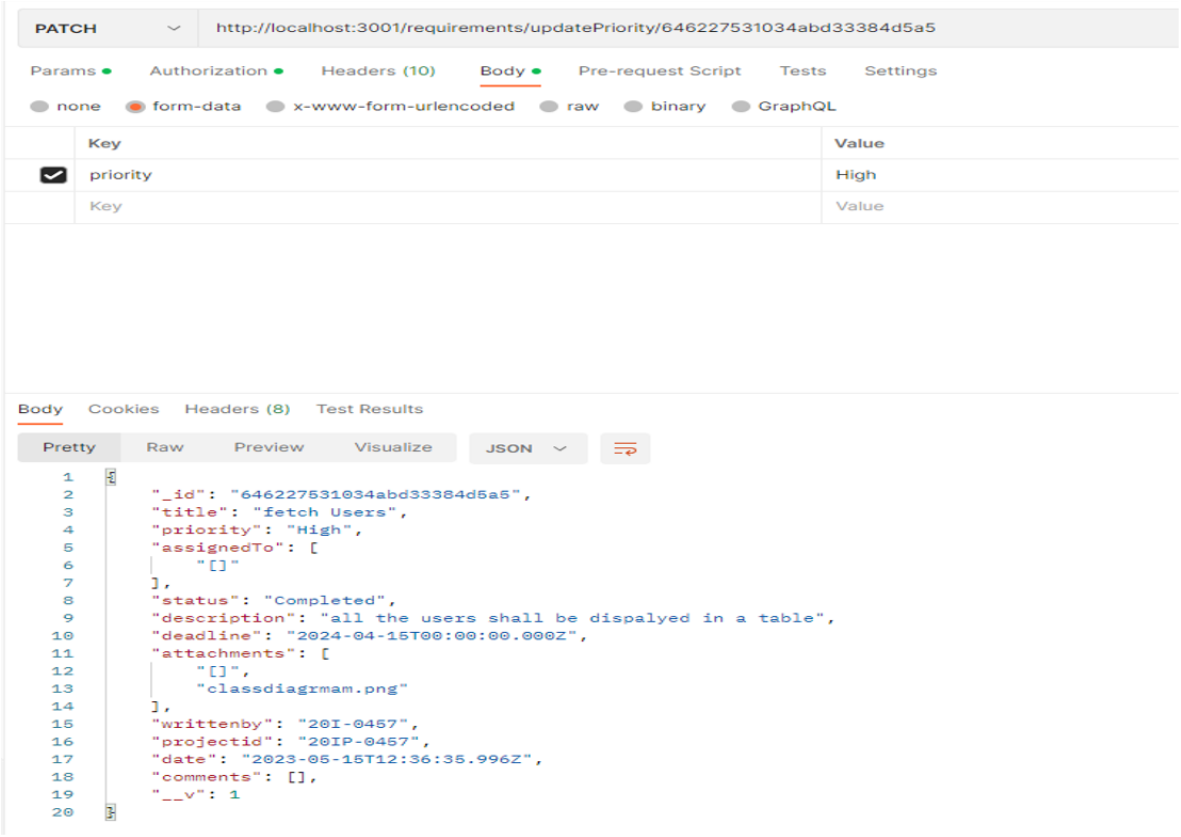
Upload attachment to a requirement



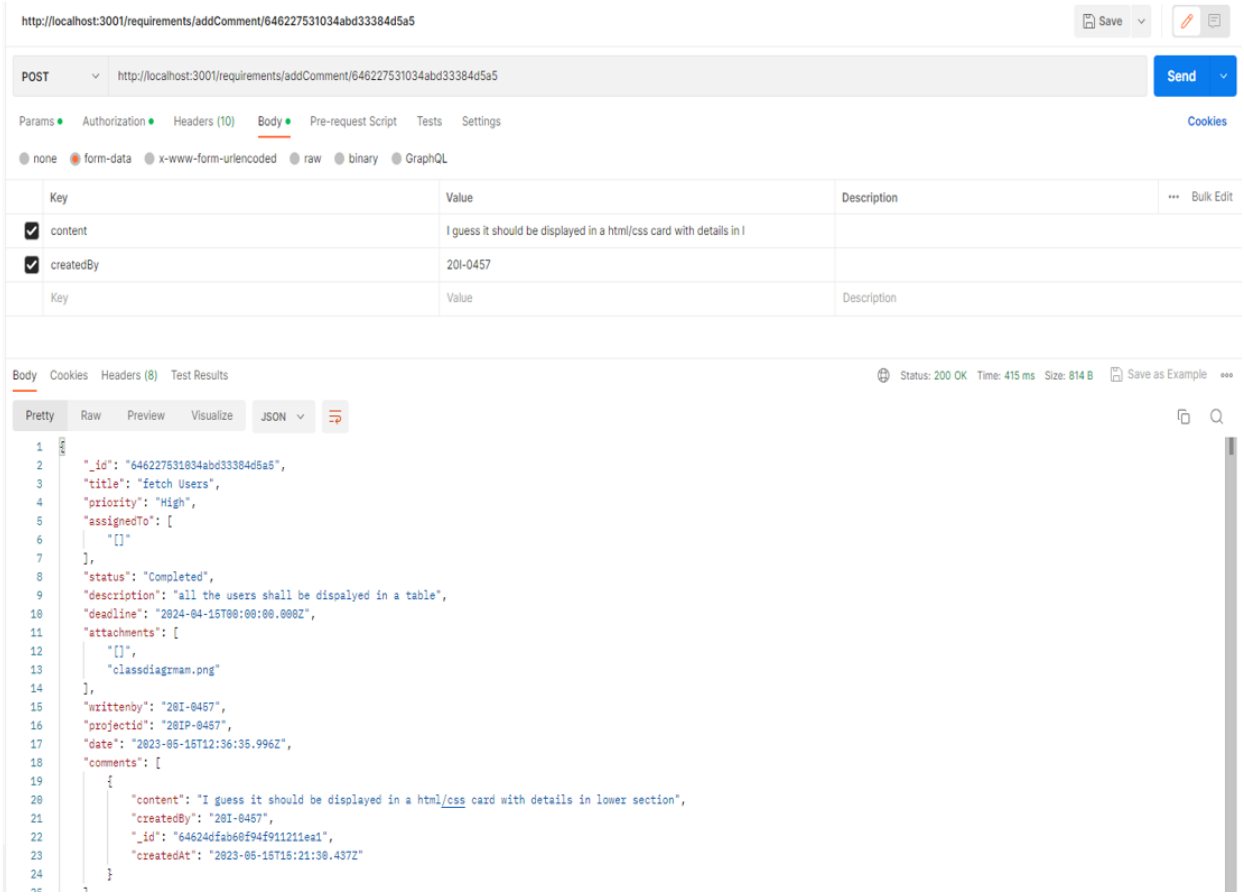
Update Deadline of requirement


PATCH
http://localhost:3001/requirements/updateDeadline/646227531034abd33384d5a5
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
Key
Value
deadline
2024-04-15
Key
Value
Body
Cookies
Headers (8)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "646227531034abd33384d5a5",
"title": "fetch Users",
"priority": "Medium",
"assignedTo": [
"[]"
],
"status": "Completed",
"description": "all the users shall be dispalyed in a table",
"deadline": "2024-04-15T00:00:00.000Z",
"attachments": [
"[]",
"classdiagrmam.png"
],
"writtenby": "20I-0457",
"projectid": "20IP-0457",
"date": "2023-05-15T12:36:35.996Z",
"comments": [],
"__v": 1
}

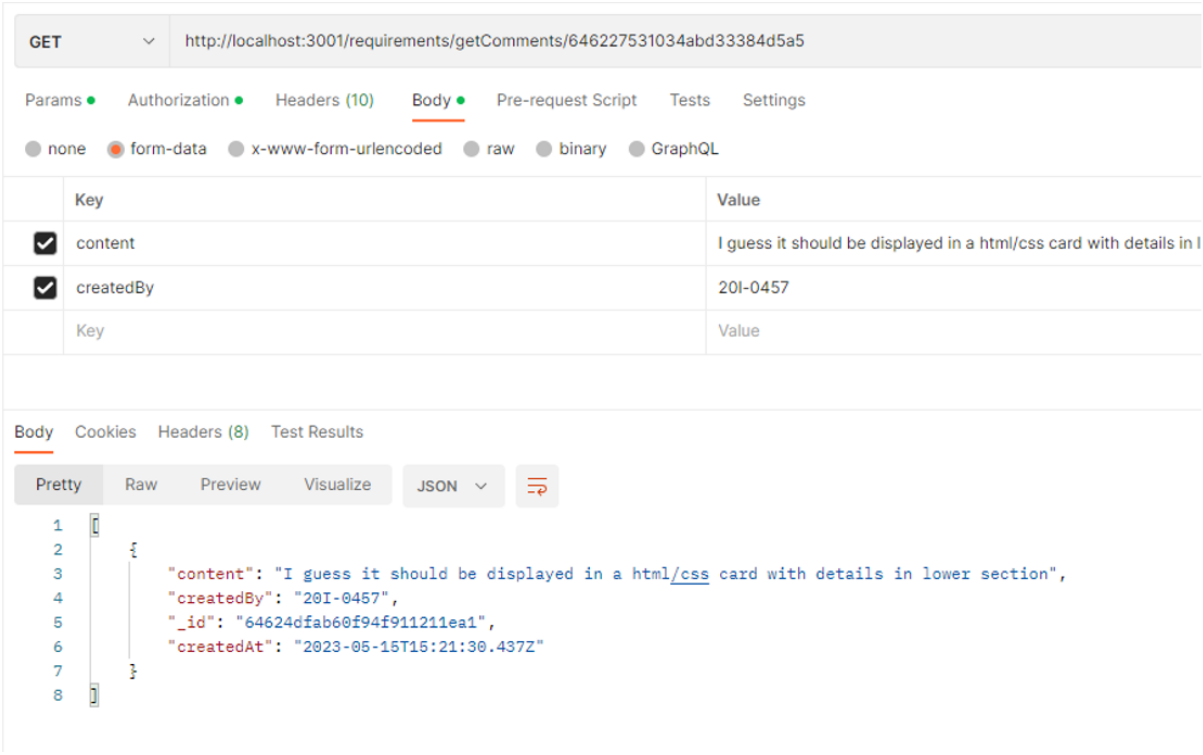
Update Priority of requirement



Add comments to requirement

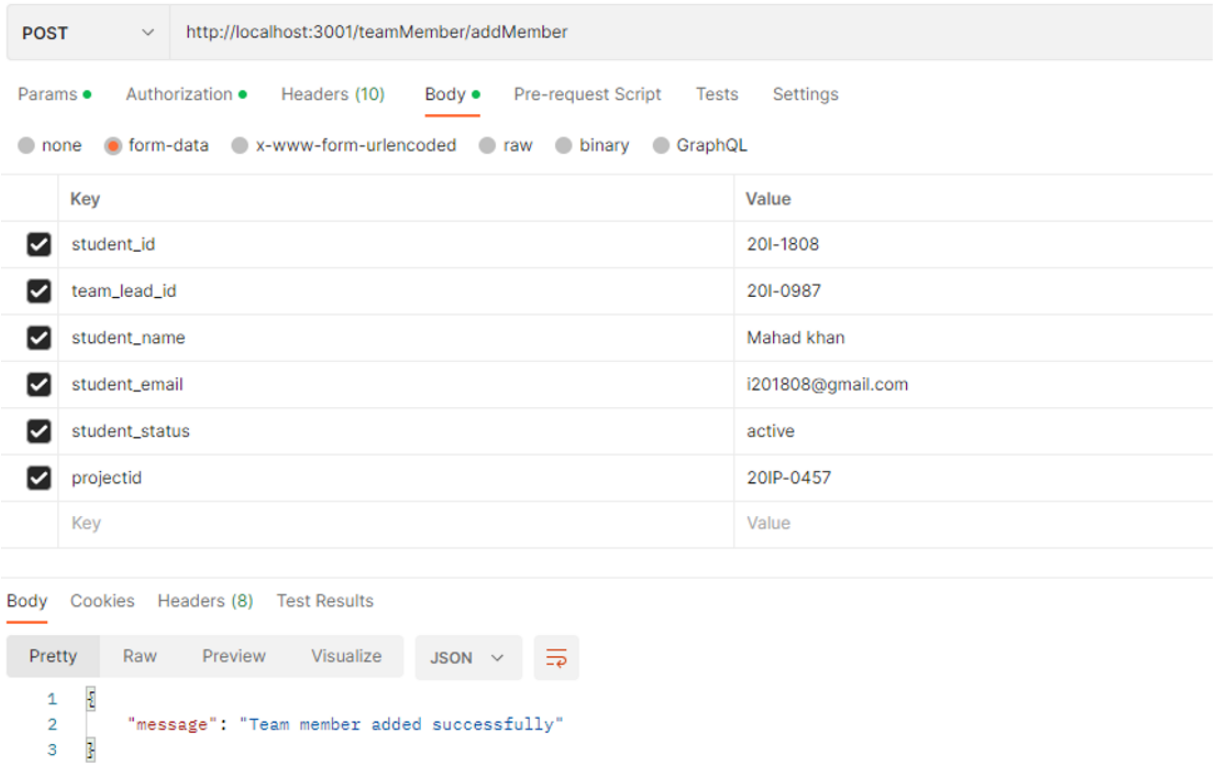


Display comments of a requirement

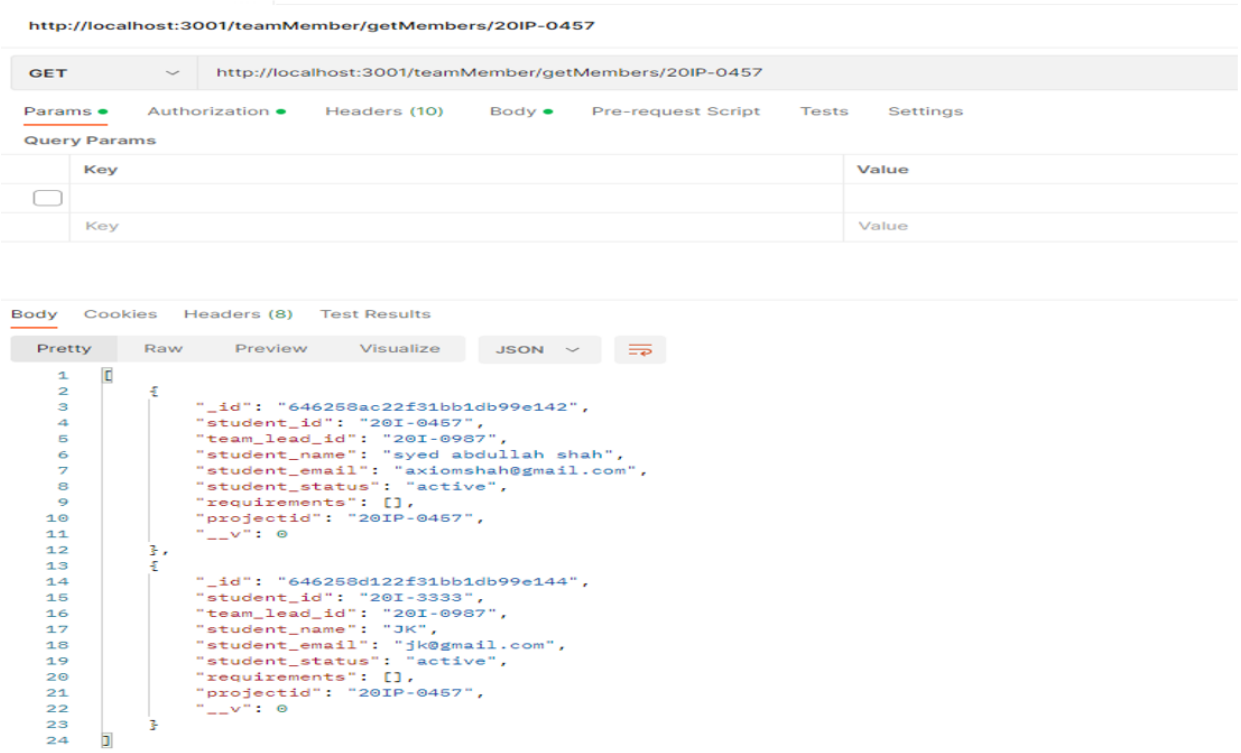


Team Member Module

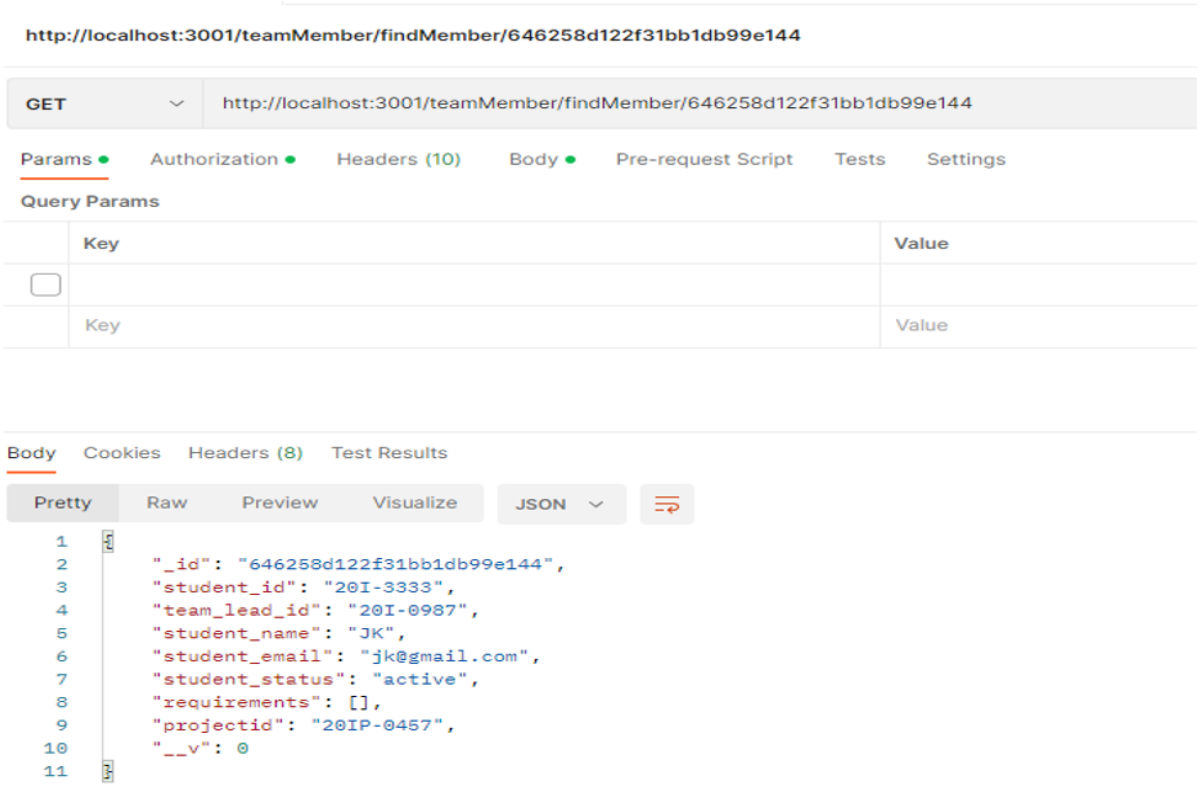
Adding team member



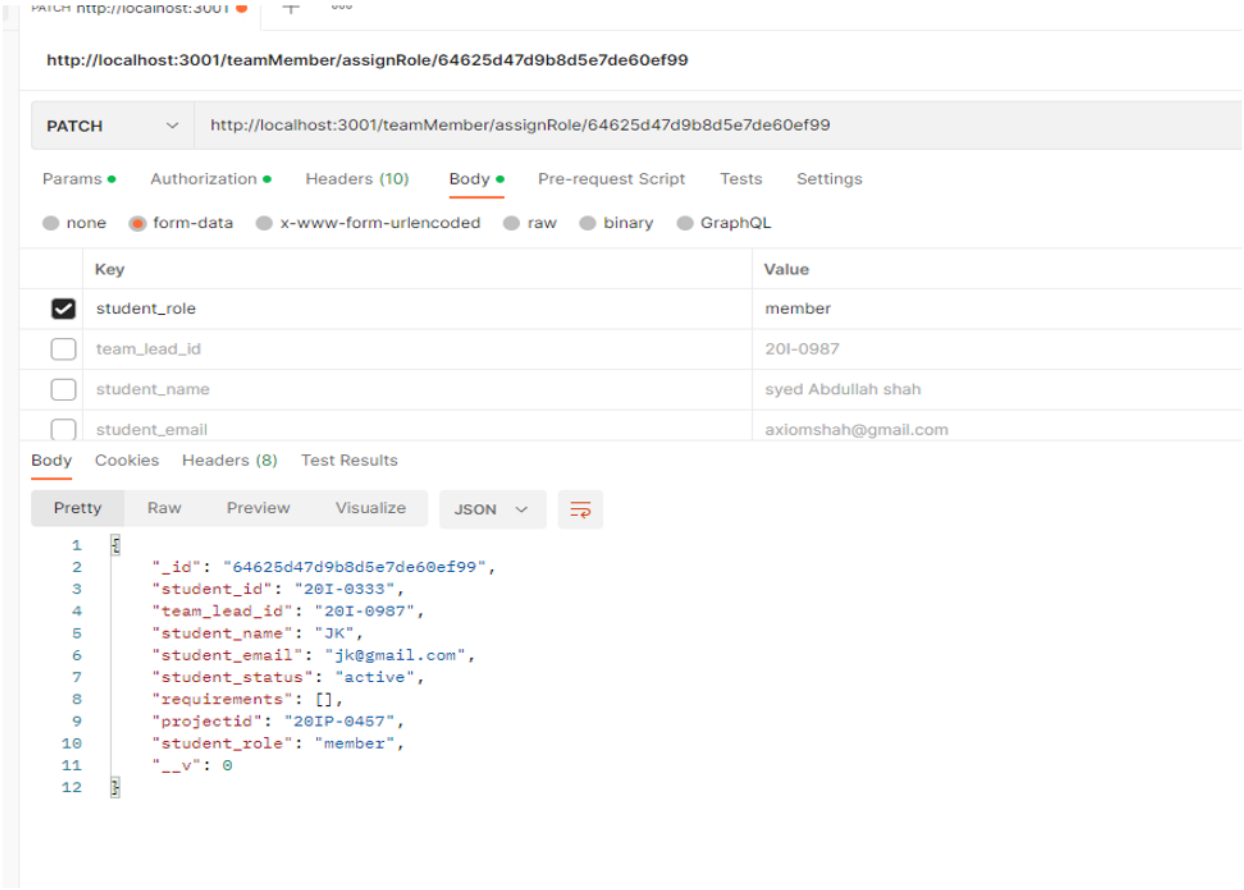
Getting all team member by Projectid



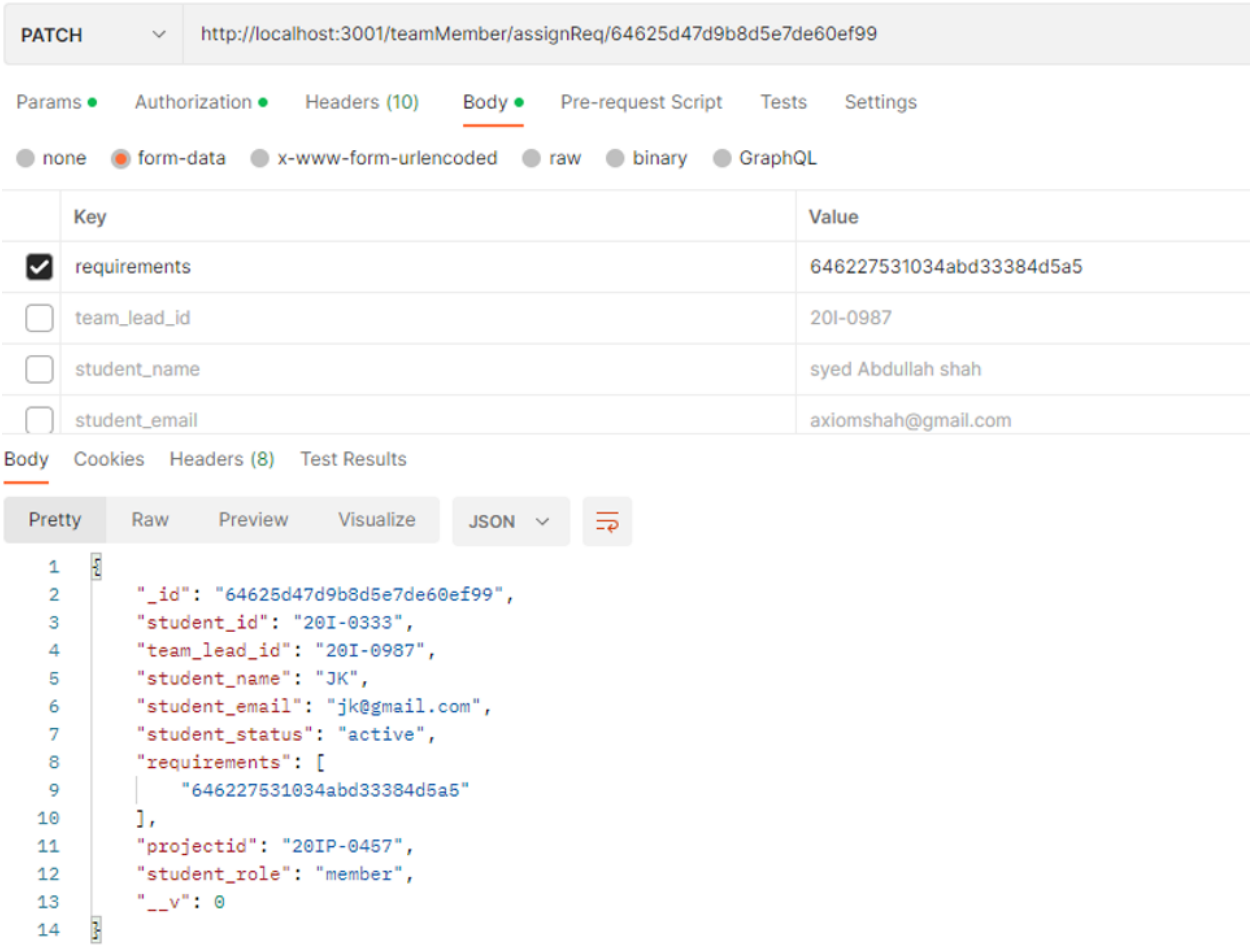
Getting a single team member by id



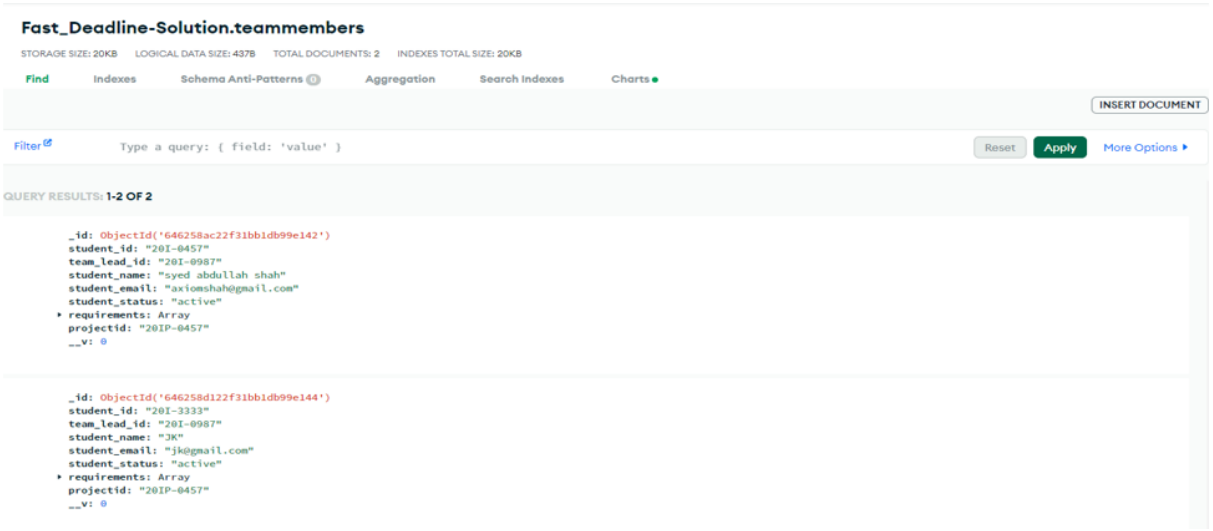
Updating the role of a team member

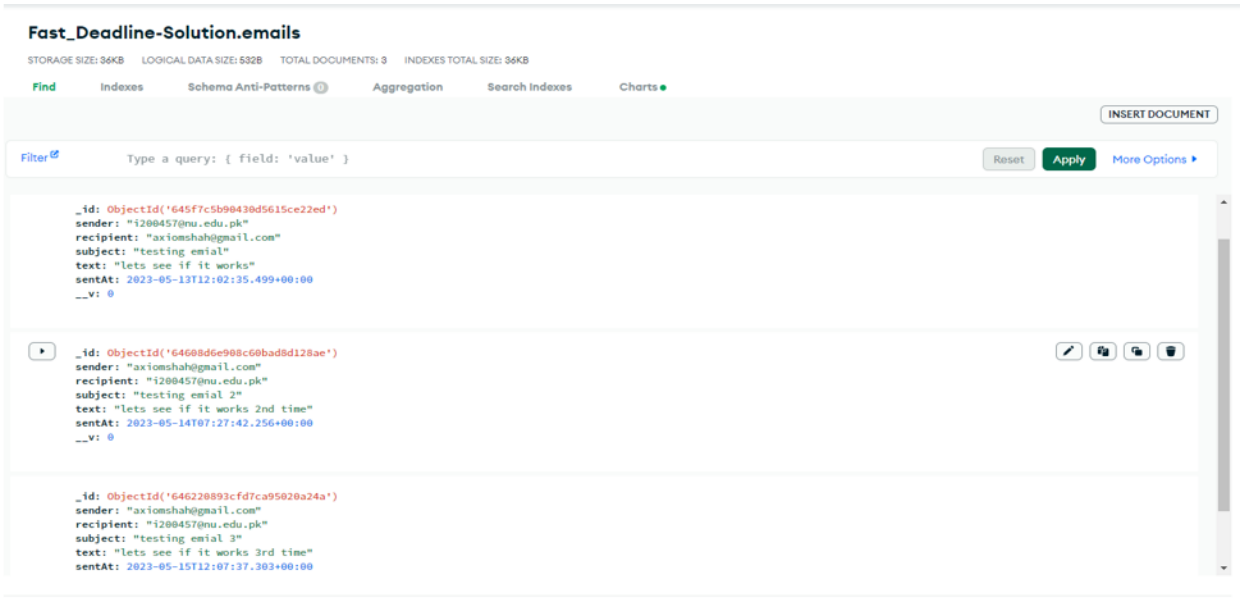

PATCH http://localhost:3001
http://localhost:3001/teamMember/assignRole/64625d47d9b8d5e7de60ef99
PATCH
http://localhost:3001/teamMember/assignRole/64625d47d9b8d5e7de60ef99
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
Key
Value
student_role
member
team_lead_id
20I-0987
student_name
syed Abdullah shah
student_email
axiomshah@gmail.com
Body
Cookies
Headers (8)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "64625d47d9b8d5e7de60ef99",
"student_id": "20I-0333",
"team_lead_id": "20I-0987",
"student_name": "JK",
"student_email": "jk@gmail.com",
"student_status": "active",
"requirements": [],
"projectid": "20IP-0457",
"student_role": "member",
"__v": 0
}

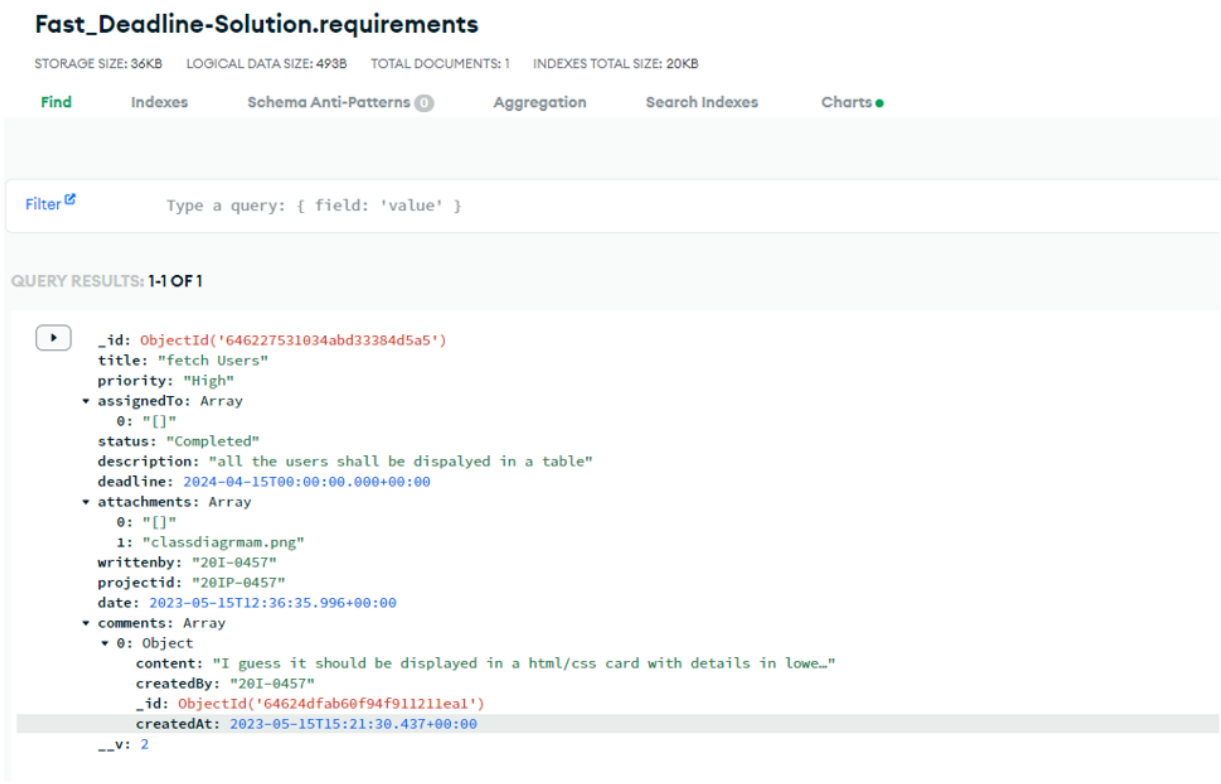
Assigning a requriement to a team member



DatabaseMongodb


Fast_Deadline-Solution.teammembers
STORAGE SIZE: 20KB LOGICAL DATA SIZE: 437B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 20KB
Find Indexes Schema Anti-Patterns Aggregation Search Indexes Charts
INSERT DOCUMENT
Filter Type a query: { field: 'value' } Reset Apply More Options
QUERY RESULTS: 1-2 OF 2
_id: ObjectId('646258ac22f31bb1db99e142')
student_id: "20I-0457"
team_lead_id: "20I-0987"
student_name: "syed abdullah shah"
student_email: "axiomshah@gmail.com"
student_status: "active"
requirements: Array
projectid: "20IP-0457"
__v: 0
_id: ObjectId('646258d122f31bb1db99e144')
student_id: "20I-3333"
team_lead_id: "20I-0987"
student_name: "JK"
student_email: "jk@gmail.com"
student_status: "active"
requirements: Array
projectid: "20IP-0457"
__v: 0


Fast_Deadline-Solution.emails
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 532B TOTAL DOCUMENTS: 3 INDEXES TOTAL SIZE: 36KB
Find Indexes Schema Anti-Patterns Aggregation Search Indexes Charts
INSERT DOCUMENT
Filter Type a query: { field: 'value' } Reset Apply More Options
_id: ObjectId('645f7c5b90430d5615ce22ed')
sender: "i200457@nu.edu.pk"
recipient: "axiomshah@gmail.com"
subject: "testing emial"
text: "lets see if it works"
sentAt: 2023-05-13T12:02:35.499+00:00
__v: 0
_id: ObjectId('64608d6e908c60bad8d128ae')
sender: "axiomshah@gmail.com"
recipient: "i200457@nu.edu.pk"
subject: "testing emial 2"
text: "lets see if it works 2nd time"
sentAt: 2023-05-14T07:27:42.256+00:00
__v: 0
_id: ObjectId('646220893cfd7ca95020a24a')
sender: "axiomshah@gmail.com"
recipient: "i200457@nu.edu.pk"
subject: "testing emial 3"
text: "lets see if it works 3rd time"
sentAt: 2023-05-15T12:07:37.303+00:00


Fast_Deadline-Solution.requirements
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 493B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB
Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes Charts
Filter
Type a query: { field: 'value' }
QUERY RESULTS: 1-1 OF 1
_id: ObjectId('646227531034abd33384d5a5')
title: "fetch Users"
priority: "High"
assignedTo: Array
0: "[]"
status: "Completed"
description: "all the users shall be dispalyed in a table"
deadline: 2024-04-15T00:00:00.000+00:00
attachments: Array
0: "[]"
1: "classdiagrmam.png"
writtenby: "20I-0457"
projectid: "20IP-0457"
date: 2023-05-15T12:36:35.996+00:00
comments: Array
0: Object
content: "I guess it should be displayed in a html/css card with details in lowe…"
createdBy: "20I-0457"
_id: ObjectId('64624dfab60f94f911211ea1')
createdAt: 2023-05-15T15:21:30.437+00:00
__v: 2