

Task Overview: API CRUD operations for student data with normalized tables, and implementing pagination logic using Node.js and PostgreSQL, with React.js frontend

Task 1 : Database Schema Design

- Provide a normalized database schema for storing student information and their marks. Ensure that the schema follows best practices for normalization.
- Include tables for students and marks.
- Specify the relationships between these tables using foreign key constraints.

Task 2 : API CRUD Operations

- Implement RESTful API endpoints using Node.js for CRUD operations on student data.
- Create endpoints for:
 - Creating a new student record
 - Retrieving a list of all students
 - Retrieving a single student by ID with marks
 - Updating a student's information
 - Deleting a student record
- Use Postman or any other application to test API endpoints and ensure they function correctly.

Task 3 : API Pagination Logic

- Extend the API created in Task 2 to support pagination for retrieving lists of students.
- Implement pagination logic using query parameters such as page and limit.
- Ensure that the API returns paginated results along with metadata like total count of records.

Task 4 : React.js Frontend Integration to interact with NodeJs APIs

- Design Bootstrap forms using React.js to interact with CRUD operations on

Add New Member

Member Name*

Enter Member Name

Member Email*

Enter Member Email

Member Age*

Enter Age

Member Parent Id

Enter Parent Id

Submit

All Members

QA

Add New Member

Id	Member Name	member Email	Age	Action
24	Qa Nadsoft	qa+upll-test-client-4@nadsoftdev.com	28	
23	Qa Nadsoft	qa+upll-test-client-3@nadsoftdev.com	32	
15	qa test	qa+ertg@nadsoftdev.com	36	
4	ts vb	qa*72l@nadsoftdev.com	30	

Show 10 entries

First Previous 1 Next Last

student data.

- Implement a Bootstrap list view to display paginated student records.

All Members

QA

Add New Member

Id	Member Name	member Email	Age	Action
24	Qa Nadsoft	qa+upll-test-client-4@nadsoftdev.com	28	
23	Qa Nadsoft	qa+upll-test-client-3@nadsoftdev.com	32	
15	qa test	qa+ertg@nadsoftdev.com	36	
4	ts vb	qa*72l@nadsoftdev.com	30	

Show 10 entries

First Previous 1 Next Last

- Integrate SweetAlerts for user feedback on create, update, and delete operations.

All Members

QA

Add New Member

Id	Member Name	member Email	Age	Action
24	Qa Nadsoft	qa+upll-test-client-4@nadsoftdev.com	28	
23	Qa Nadsoft	qa+upll-test-client-3@nadsoftdev.com	32	
15	qa test	qa+ertg@nadsoftdev.com	36	
4	ts vb	qa*72l@nadsoftdev.com	30	

Show 10 entries

First Previous 1 Next Last

Are you sure?

If You delete this Member Then this action can not be undone.

Yes, delete it! Cancel

Additional Deliverables:

- **Code Zip:** Provide a zip file containing the Node.js project code, including all necessary files and dependencies.
- **Database Schema SQL Script:** Include SQL scripts to create the necessary tables and relationships in the PostgreSQL database.
- **API Collection:** Export the Postman collection containing requests for testing the API endpoints. This collection should include sample requests for each CRUD operation and pagination logic.