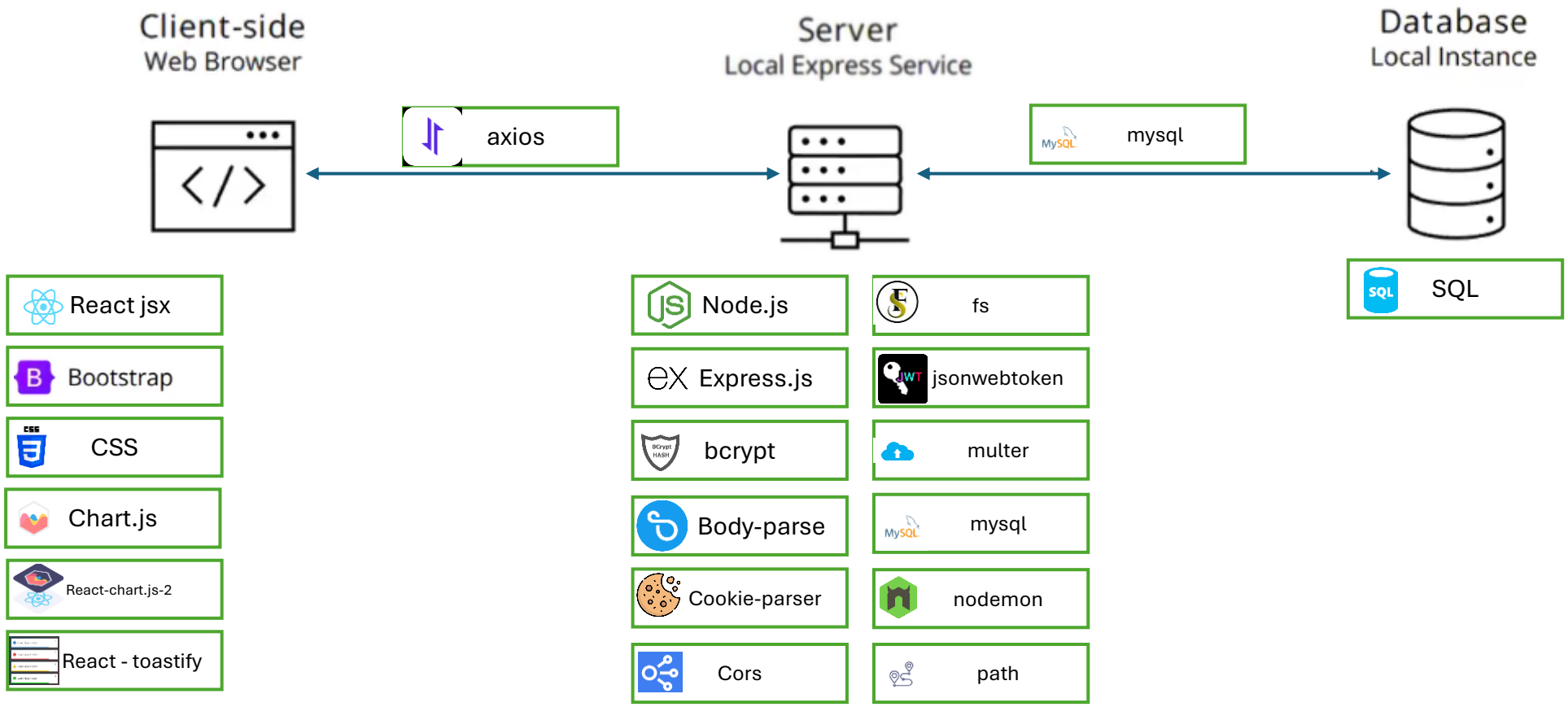
Full Stack Project

Laith khater 323965152

Rami hanna 323956557

Top-down View



About the project

- באפליקציה יש 2 סוגים של משתמשים יש login ו register ולפי המייל יודעים מי המרצה ומי הסטודנט :
- 1- מורה :

- יכול להוסיף כיתה חדשה עם ציונים
- יכול להוסיף סטודנט חדש לכיתה עם הציונים הרלוונטיים לכיתה הזאת
- יכול להוסיף עבודת בית עם תאריך וקובץ כלשהו
- יכול לעדכן הנוכחות לכל כיתה ולכל סטודנט
- יש אפשרות גם לראות מאזן הציונים לכל כיתה וגם יש גרף לכל כיתה (יש אפשרות לערוך הציונים גם)

2- סטודנט :

- יכול לראות הציונים שלו בכל הקורסים כולל הציון הסופי
- יש אפשרות גם להגיש את עבודות הבית

Technology Stack Overview

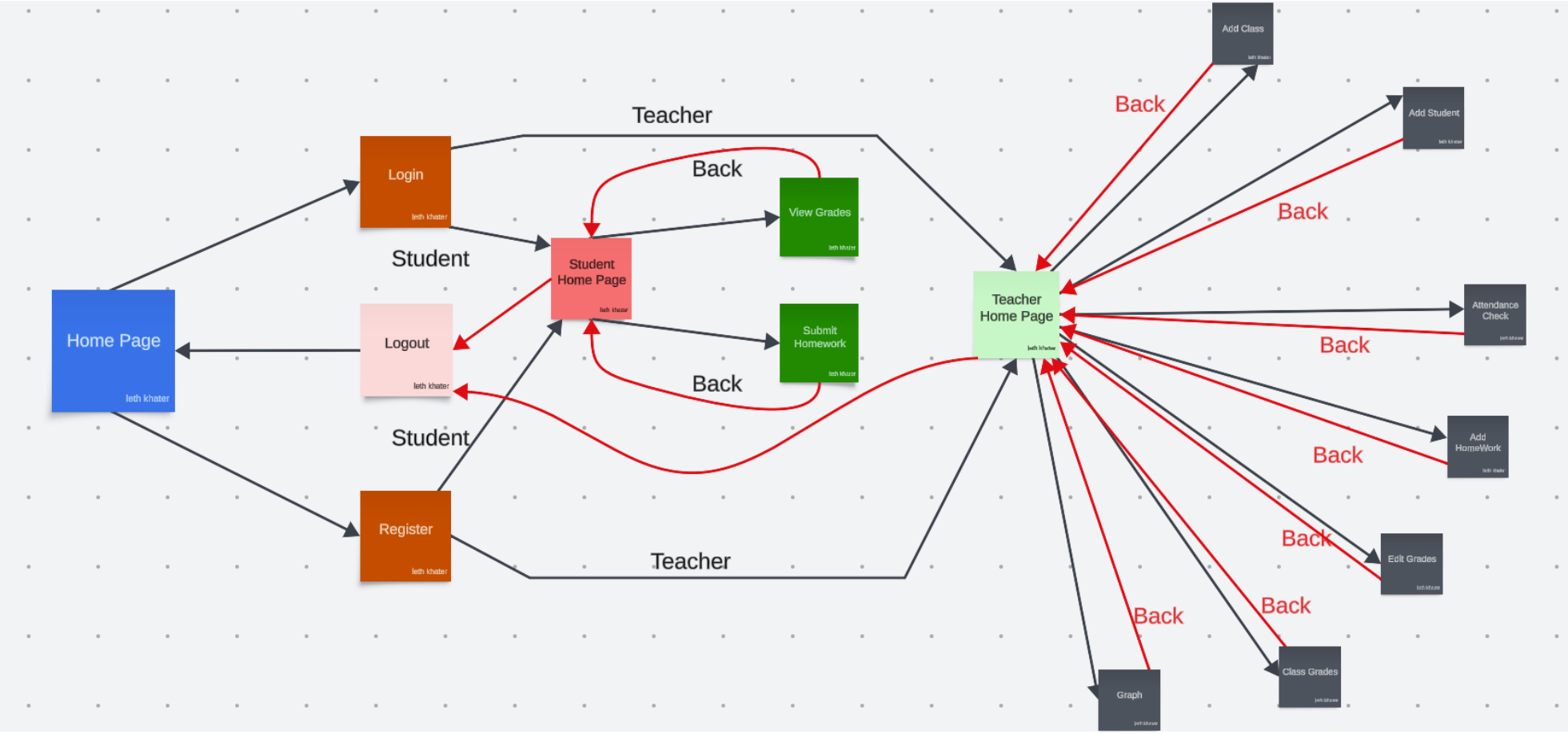
- We implemented the project using the following technologies:

- Client Side : HTML, CSS, JavaScript, Bootstrap
- Server Side : Node Js
- Database : phpMyAdmin (SQL Server)

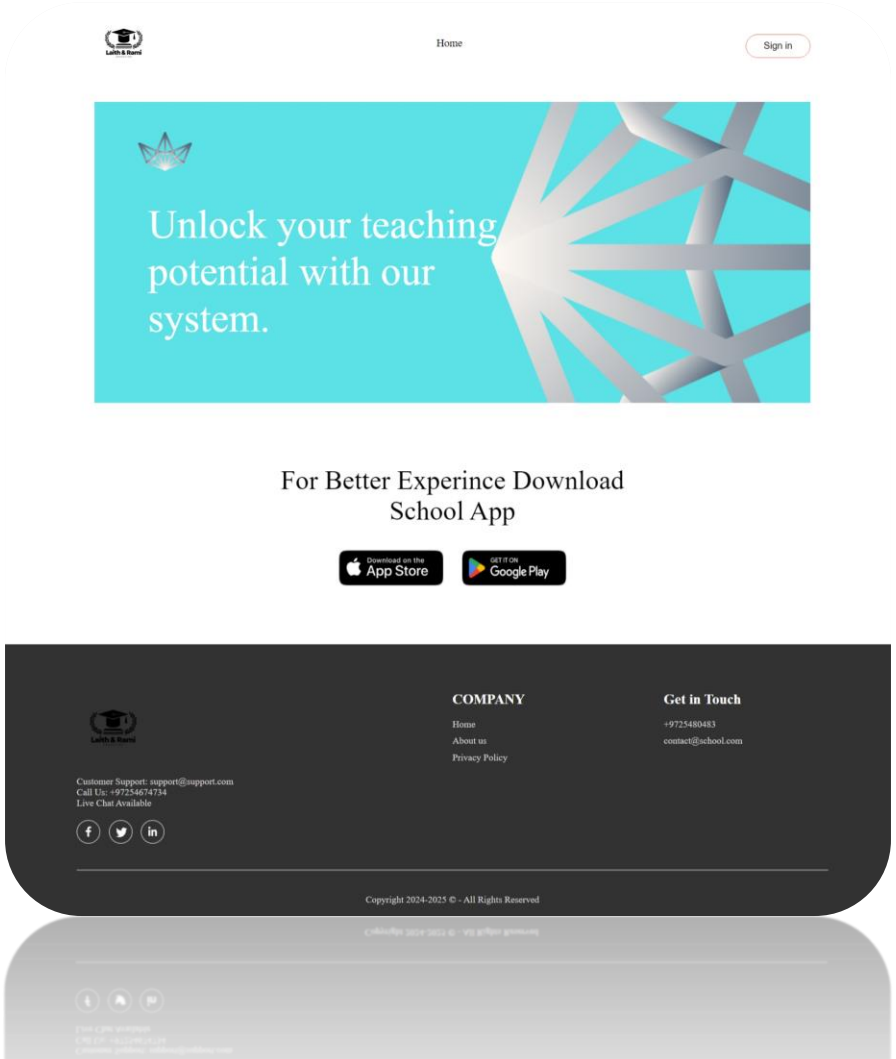
What the application do - Features

- Login
- Register
- Logging Exceptions
- Teacher - > add class
- Teacher -> add student
- Teacher -> add homework
- Teacher -> attendance check
- Teacher -> edit grades
- Teacher -> class grades
- Teacher -> grades graph
- Student -> see and submit homework
- Student -> see grades

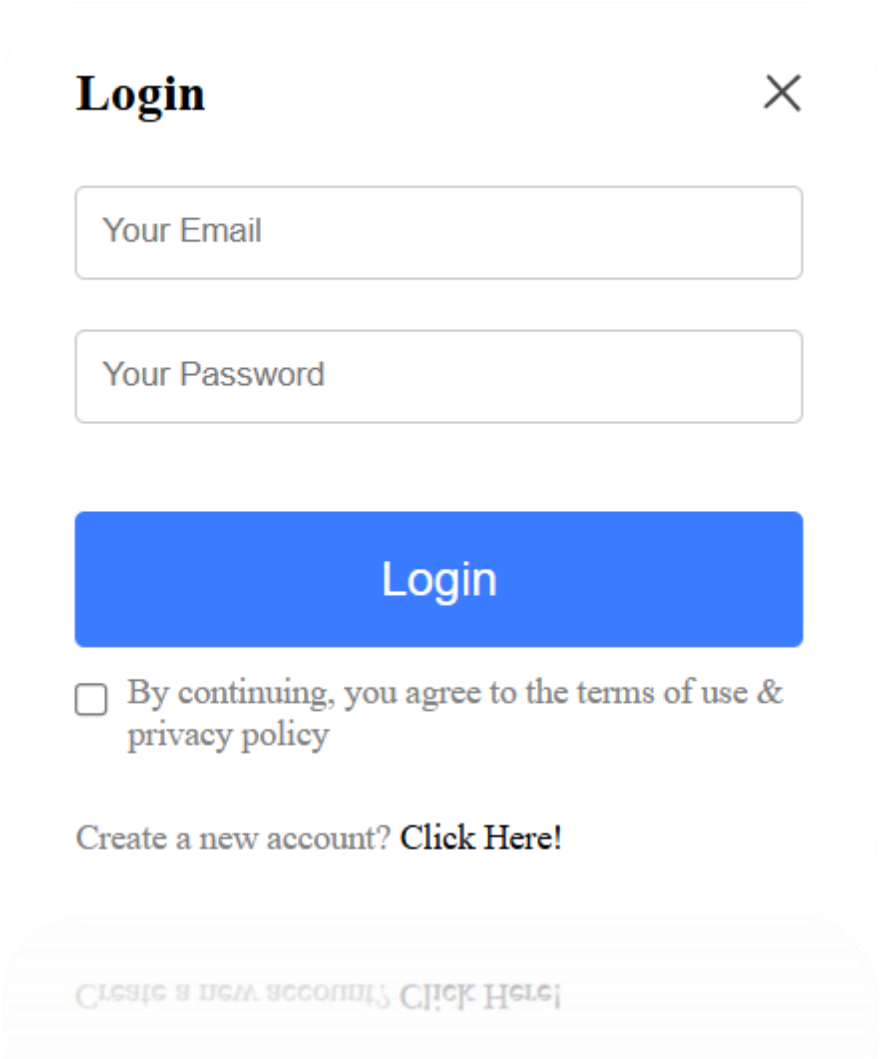
Use Case Diagram – Main Process



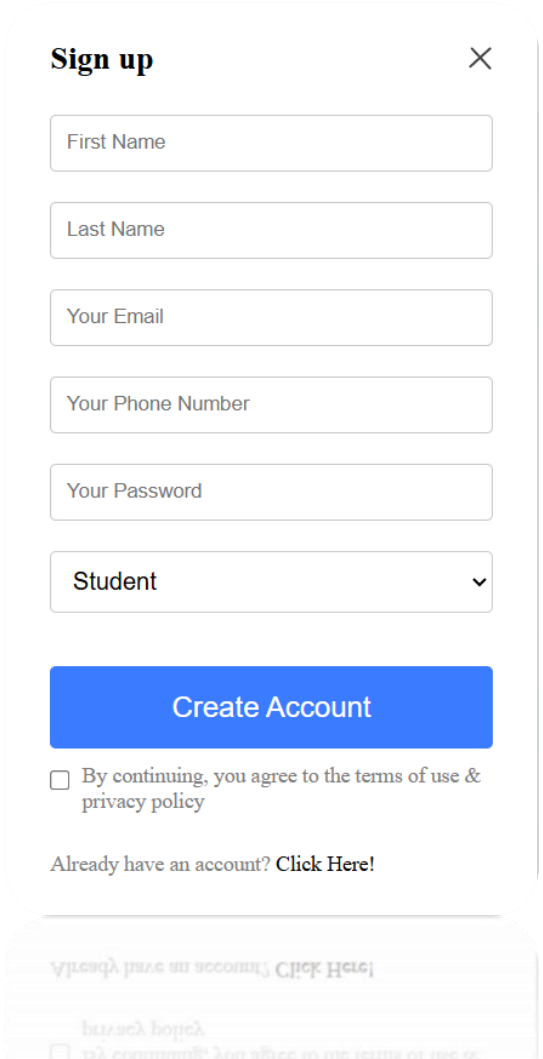
Home Page


Home
Sign in
Unlock your teaching potential with our system.
For Better Experince Download School App
Download on the App Store
GET IT ON Google Play
COMPANY
Home
About us
Privacy Policy
Get in Touch
+9725480483
contact@school.com
Customer Support: support@support.com
Call Us: +97254674734
Live Chat Available
Copyright 2024-2025 © - All Rights Reserved

Student / Teacher login page

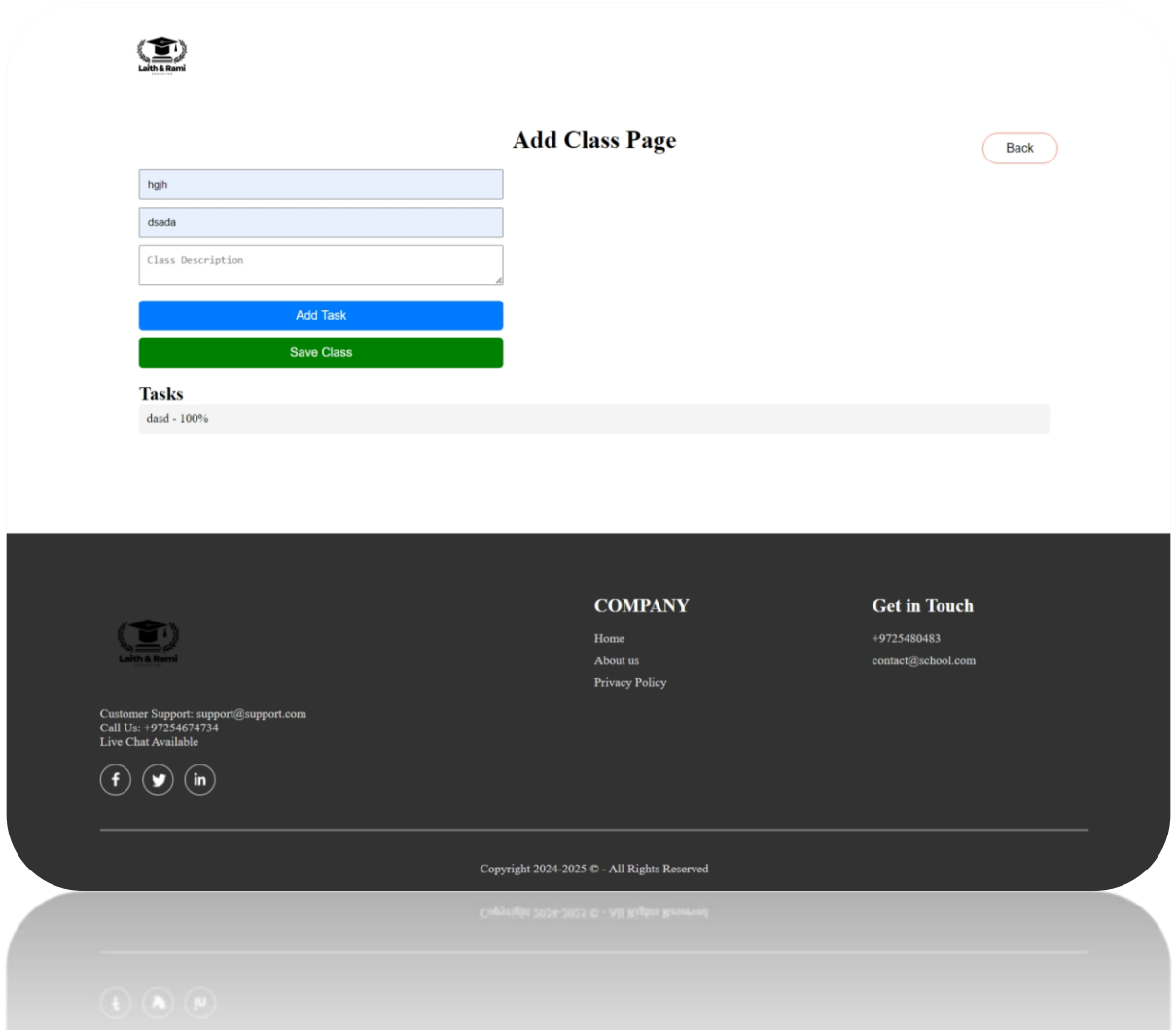


Student / Teacher register page

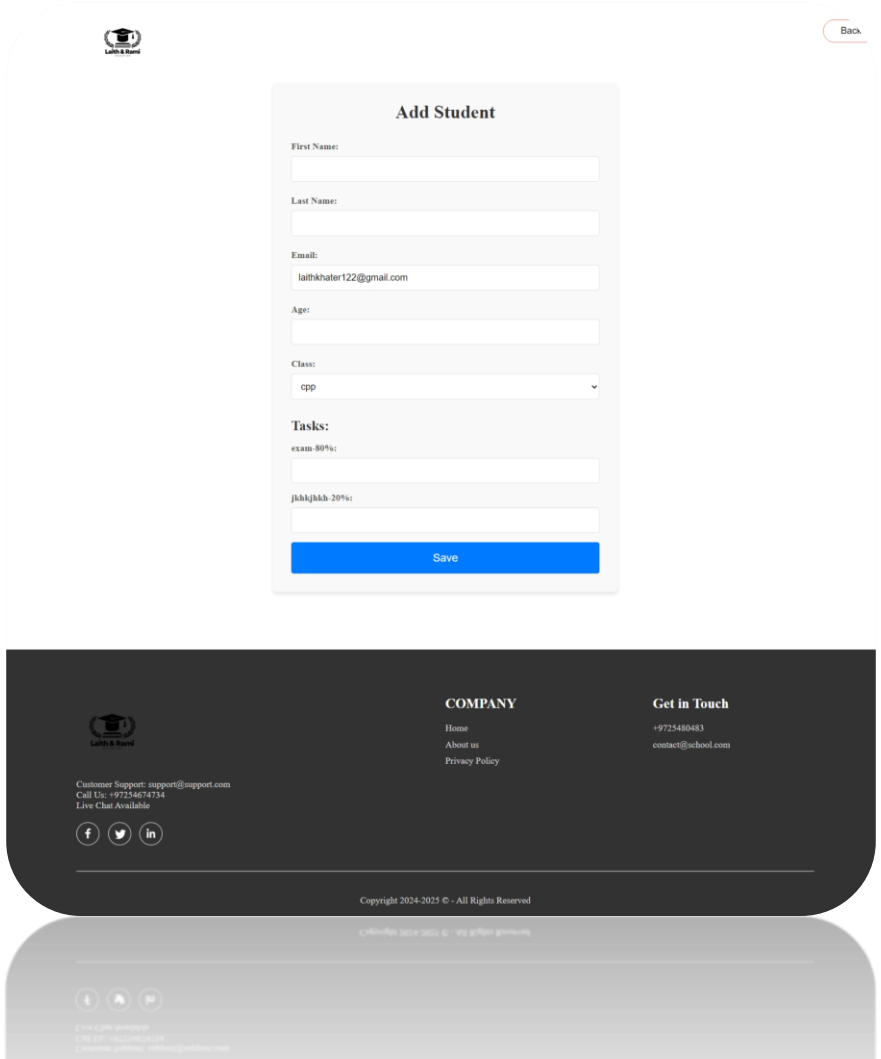


Teacher Main Page

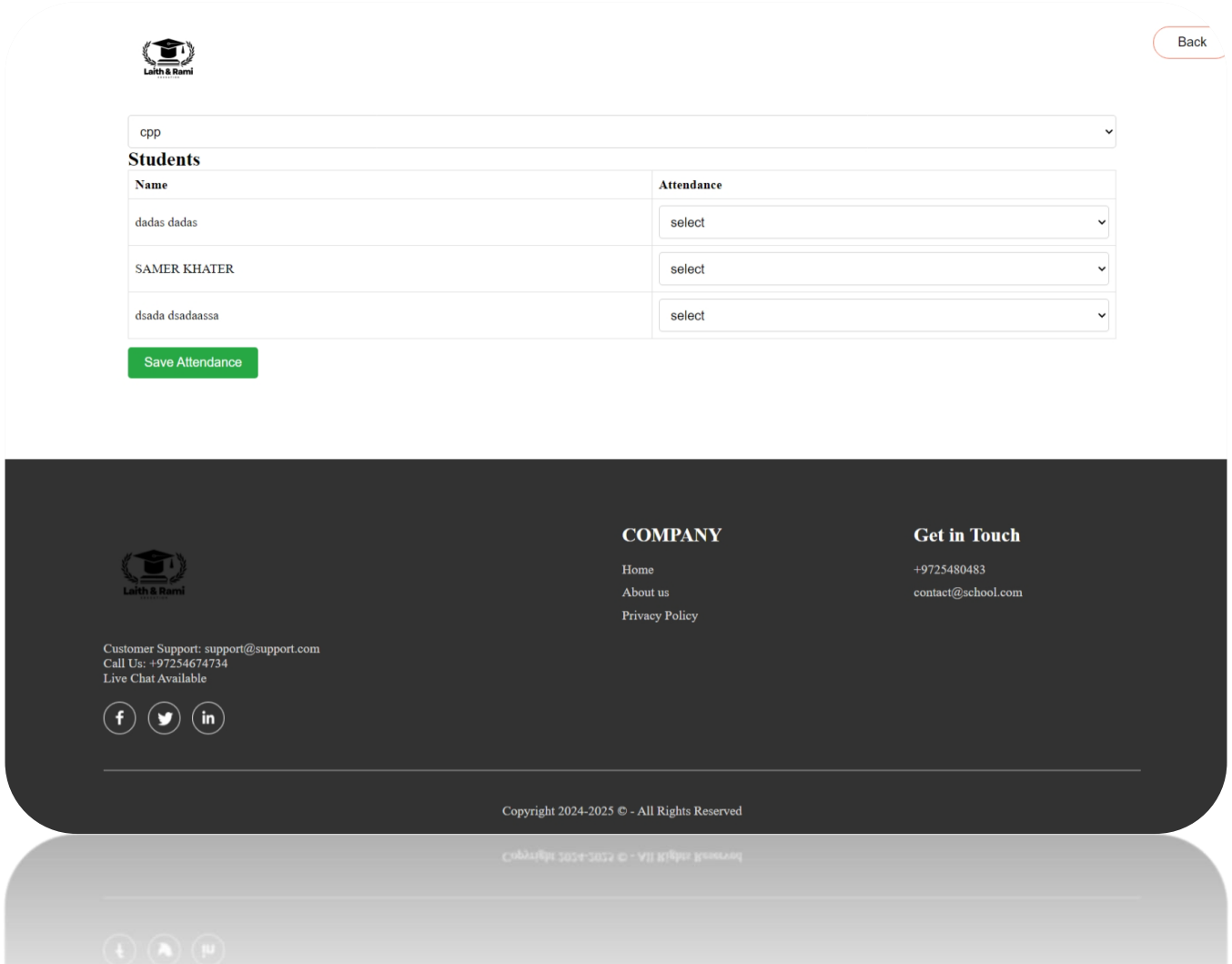
Teacher – Add Class Page



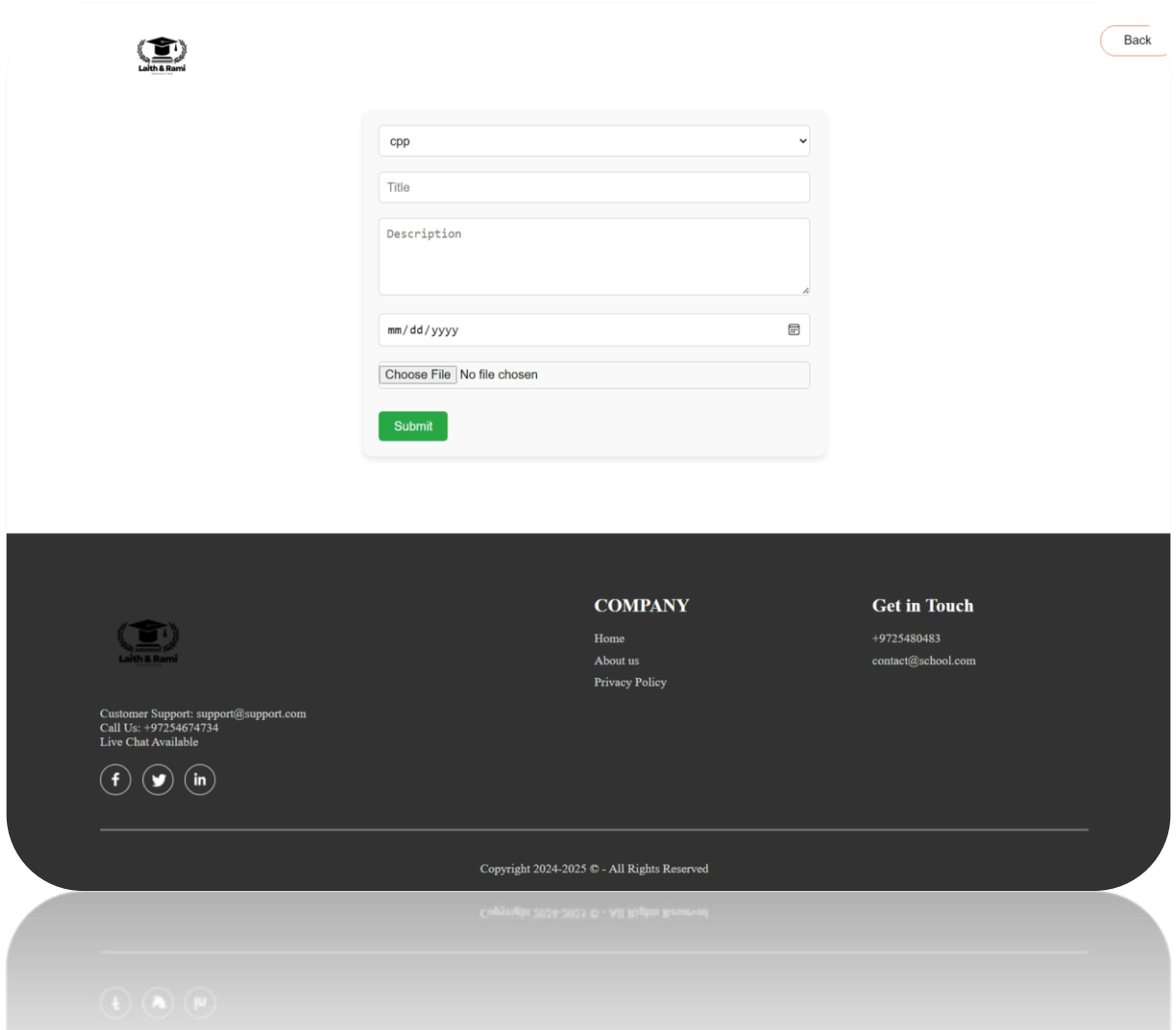
Teacher – Add Student Page



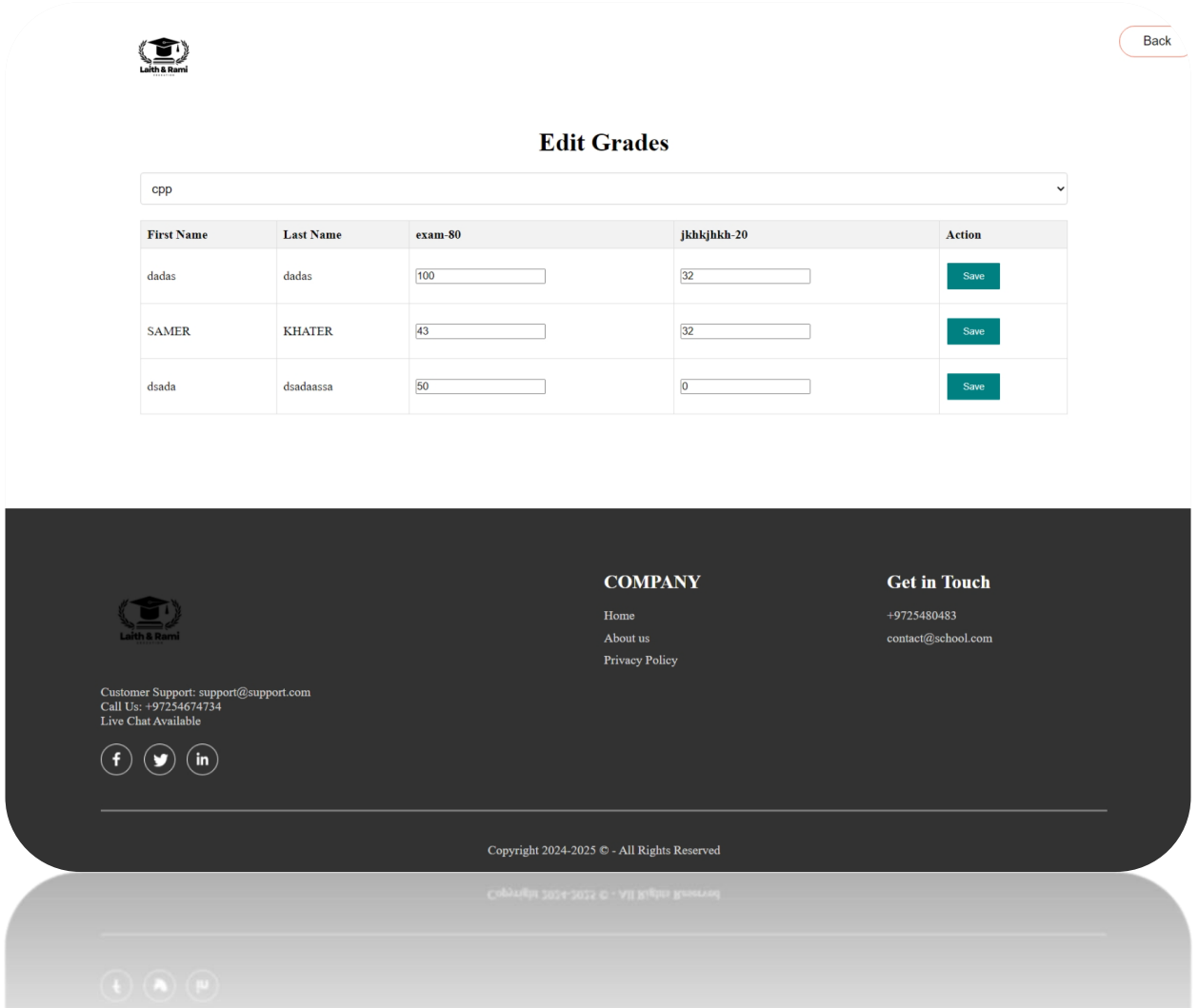
Teacher – Attendance Check Page



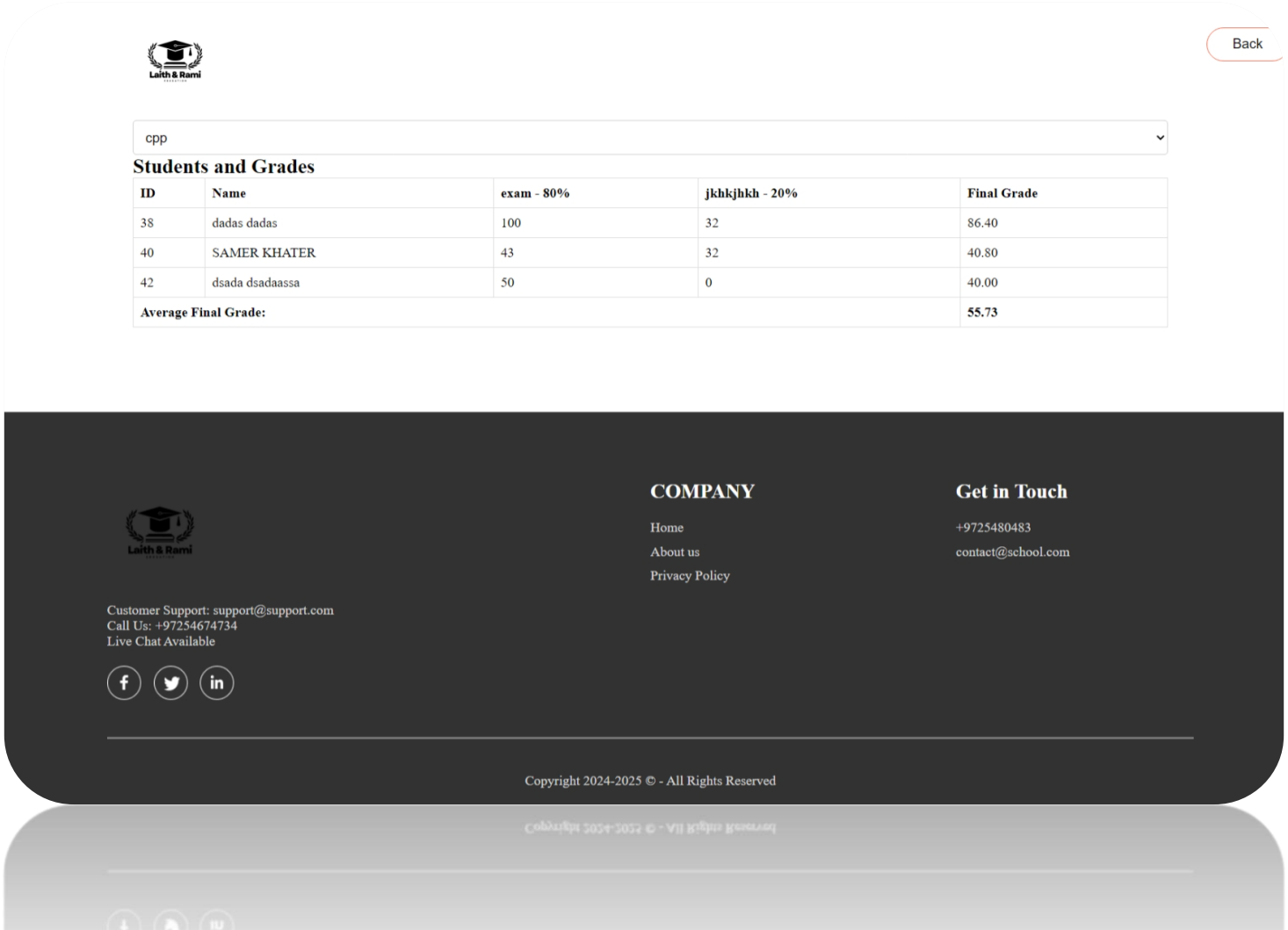
Teacher – Add Homework Page



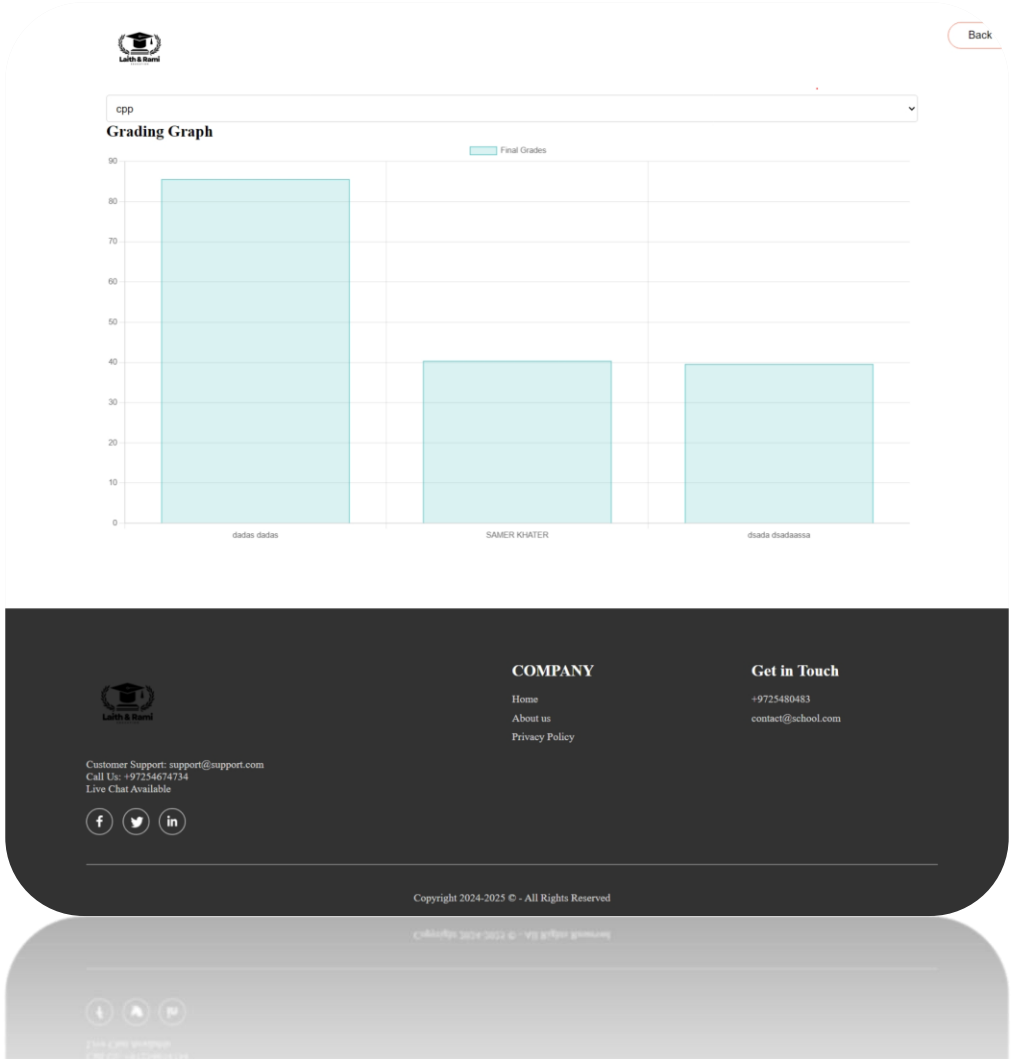
Teacher – Edit Grades Page



Teacher – View Grades Page

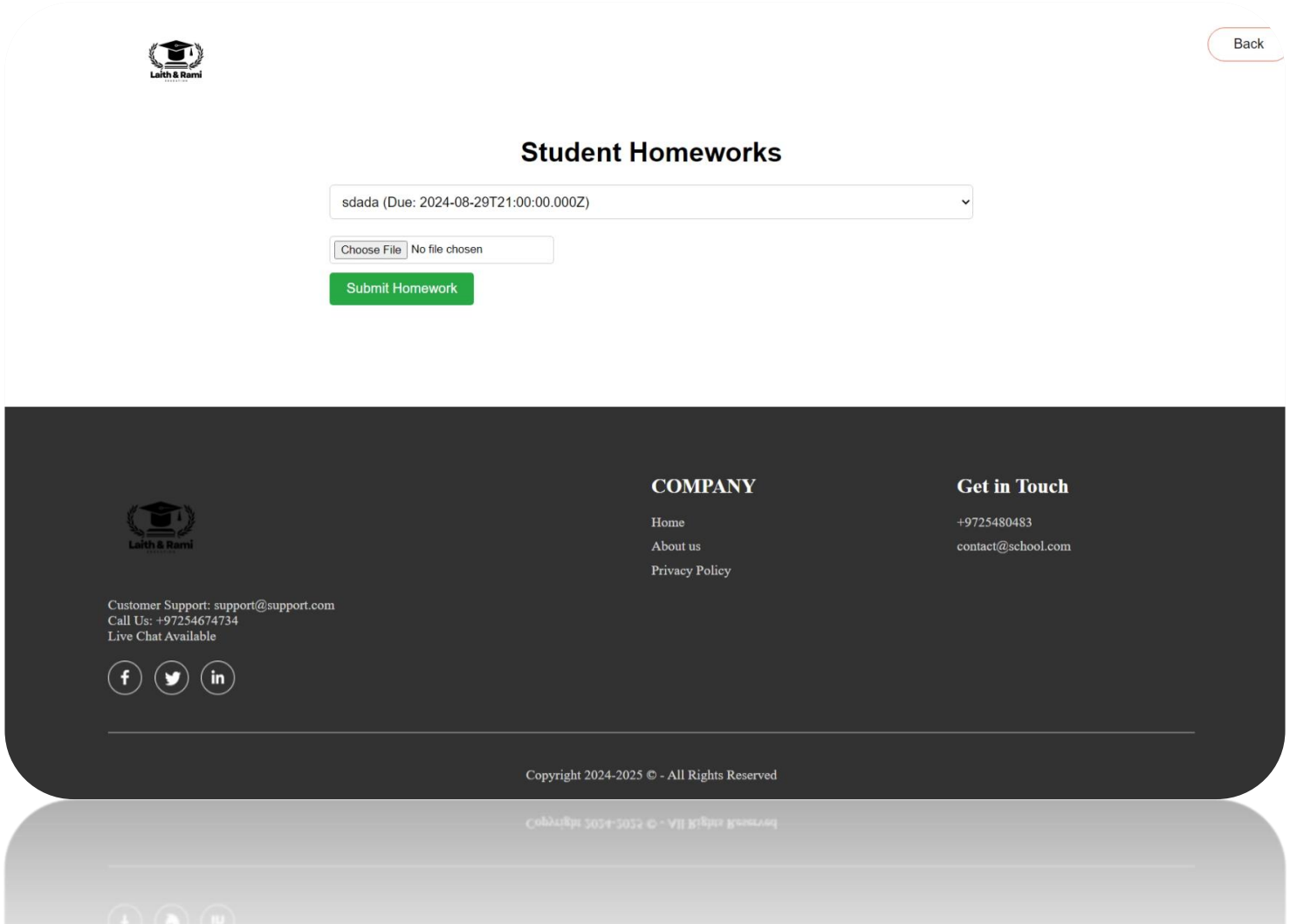


Teacher – Grades Graph Page

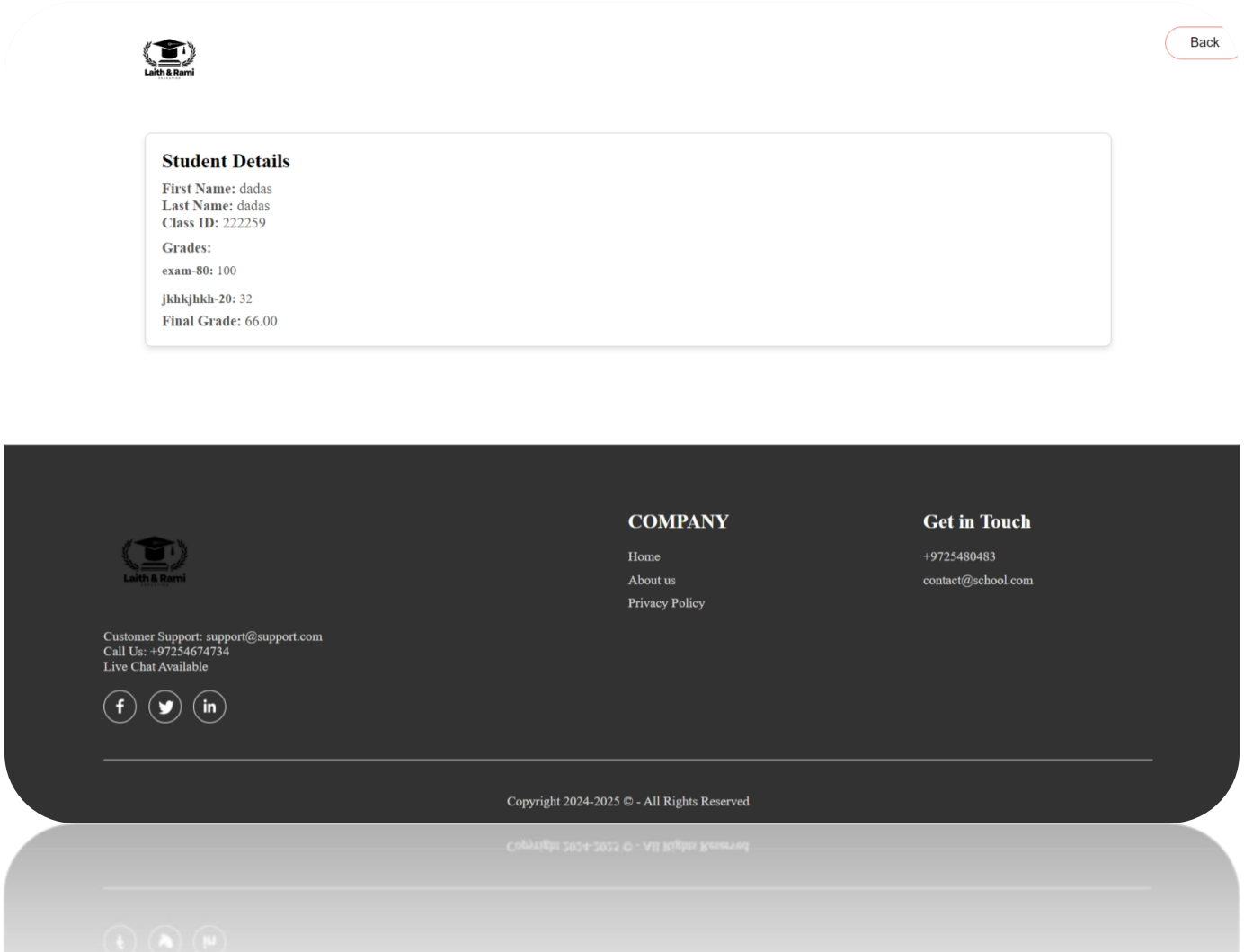


Student Main Page

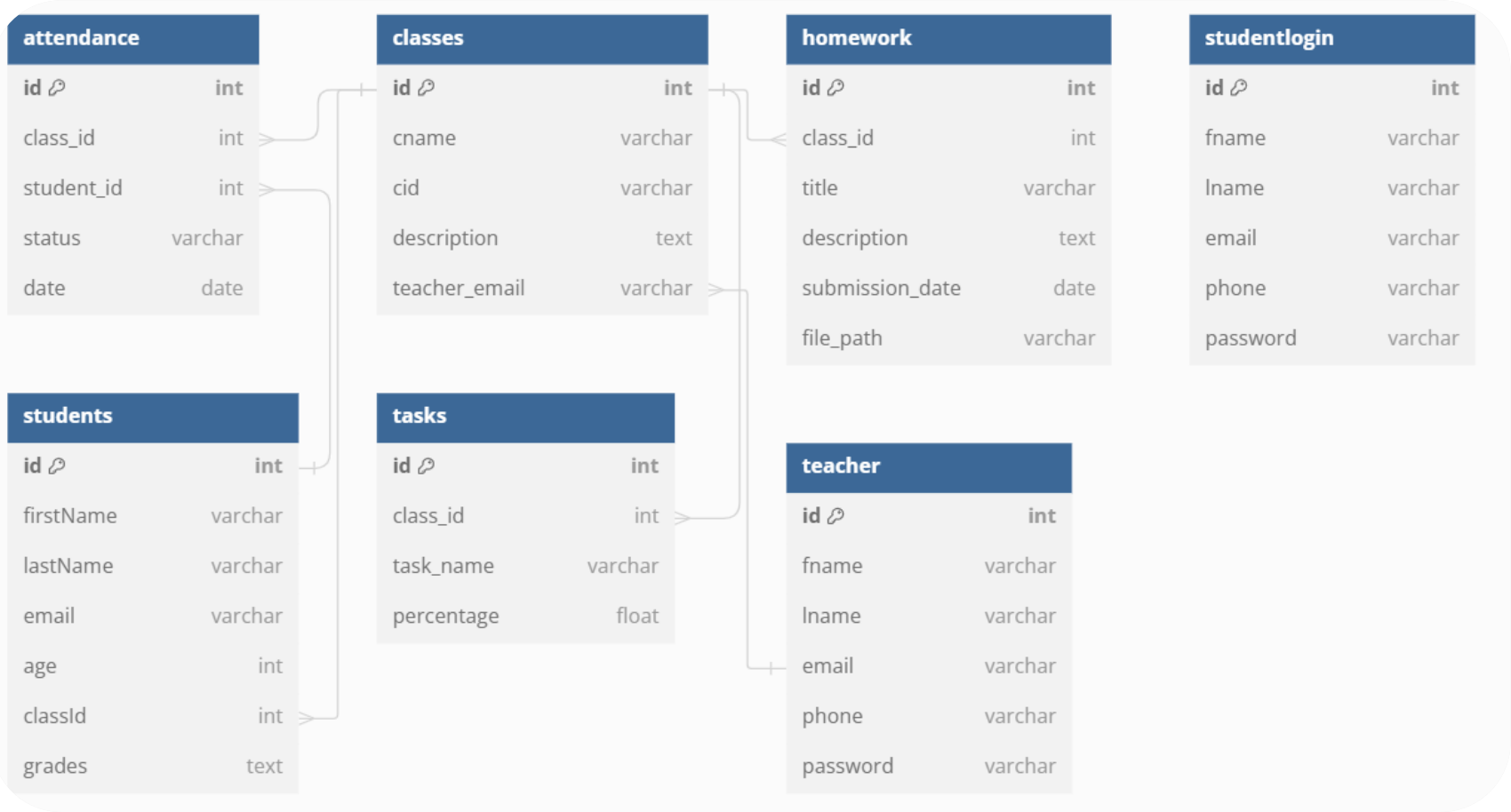
Student – HomeWorks Page



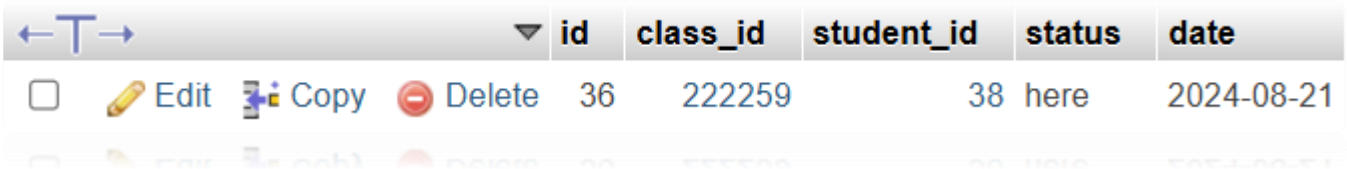
Student – Grades Page



Classes Diagram



DB – attendance table



	id	class_id	student_id	status	date
Edit Copy Delete	36	222259	38	here	2024-08-21

DB – classes table

	id	cname	cid	description	teacher_email
Edit Copy Delete	222259	cpp	1	wsdgfsd	laithkhater122@gmail.com
Edit Copy Delete	222260	hgjh	dsada	dsada	laithkhater122@gmail.com
Edit Copy Delete	222261	Math 101	MATH101	Introduction to Algebra	teacher@example.com
Edit Copy Delete	222263	hgjh	121212		laithkhater122@gmail.com

DB – homework table

	id	class_id	title	description	submission_date	file_path
Edit Copy Delete	9	222259	sdada	dasdada	2024-08-30	uploads\1724247109230-×©×□×¢×□×¨ ×□×□×¨×□ ×□×§×¨×□...

DB – student login table

	id	fname	lname	email	phone	password
Edit Copy Delete	11	dadas	dadas	laithkhater1221@gmail.com	0542392090	$2b$10$VhlOJowvSGuXrVr.lWjdAOelh15rq.y40mQN73sSokM...

DB – teacher table

id	fname	lname	email	phone	password
0	laith	khater	laithkhater122@gmail.com	123123123	$2b$10$blxVlm6jQkHrLVBDnVGFBuE6L40vd7U4wsYtlCptiHy...

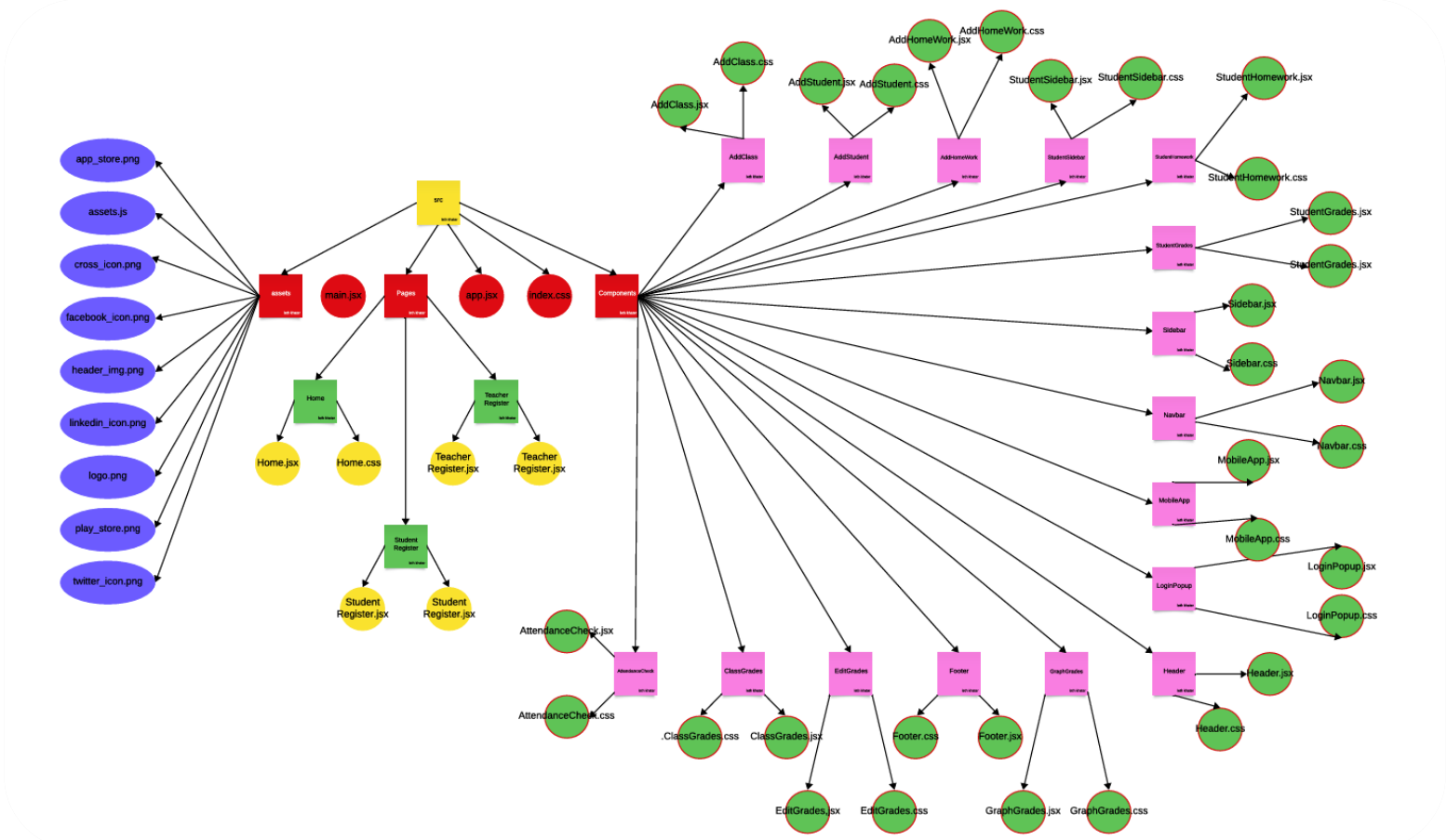
DB – students table

	id	firstName	lastName	email	age	classId	grades
Edit Copy Delete	38	dadas	dadas	laithkhater1221@gmail.com	34	222259	{"exam-80":"100","jkhkjhkh-20":"32"}
Edit Copy Delete	40	SAMER	KHATER	laithkhater1223@gmail.com	32	222259	{"exam-80":"43","jkhkjhkh-20":"32"}
Edit Copy Delete	41	dsada	dsadaassa	laithkhater1224@gmail.com	21	222260	{"dasd-100":"50"}
Edit Copy Delete	42	dsada	dsadaassa	laithkhater1227@gmail.com	21	222259	{"exam-80":"50","jkhkjhkh-20":"0"}

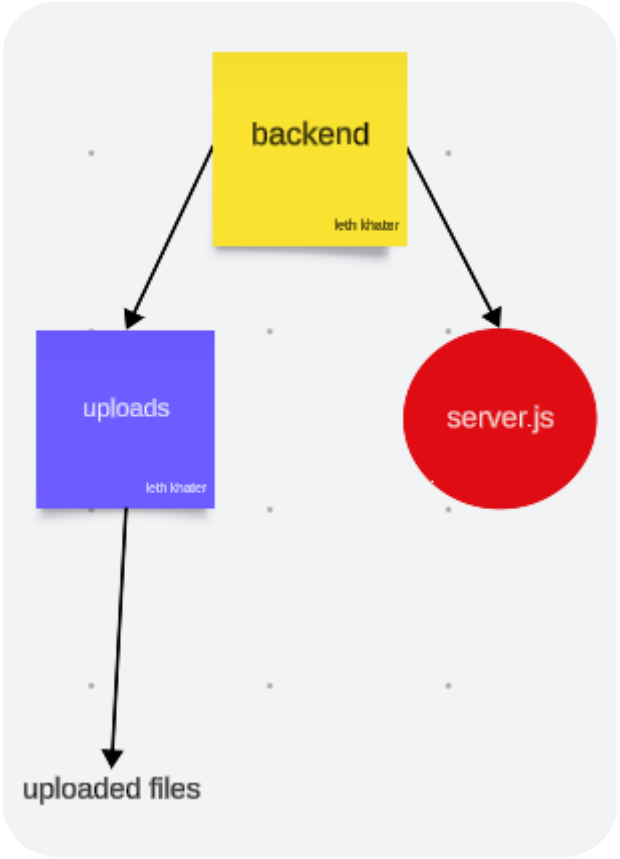
DB – tasks table

	id	class_id	task_name	percentage
Edit Copy Delete	60	222259	exam	80
Edit Copy Delete	61	222259	jkhkjhkh	20
Edit Copy Delete	62	222260	dasd	100
Edit Copy Delete	63	222261	Homework 1	20
Edit Copy Delete	64	222261	Quiz 1	10
Edit Copy Delete	65	222263	dasd	100

Client Side – Project Tree



Server Side – Project Tree



API Endpoints

Path	Method	Params	Returns

/register	POST	{
		"fname": "John",
		"lname": "Doe",
		"email": "john.doe@example.com",
		"phone": "123-456-7890",
		"password": "securepassword123",
		"role": "Student"
		}
			HTTP Status: 200 OK
			
			{
			"Message": "Student registered successfully!"
			}
			Or
			{
			"Message": "Teacher registered successfully!"
			}

/login	GET	**Query Parameters (from req.query):**
		- email: The email address of the user trying to log in.
		- password: The password provided by the user for authentication.
			Success teacher :
			{
			"success": true,
			"message": "Login successful",
			"role": "Teacher"
			}
			
			Success Student :
			{
			"success": true,
			"message": "Login successful",
			"role": "Student"
			}

API Endpoints

Path	Method	Params	Returns

/api/classes	POST	{
		"cname": "Mathematics 101",
		"id": 123,
		"description": "Introduction to basic mathematics.",
		"tasks": [
		{
		"taskName": "Homework 1",
		"percentage": 20
		},
		{
		"taskName": "Midterm Exam",
		"percentage": 30
		}
		],
		"teacher_email": "teacher@example.com"
		}
			If tasks are provided and successfully inserted:
			{
			"Message": "Class and tasks saved successfully!"
			}
			
			If no tasks are provided:
			{
			"Message": "Class saved successfully without tasks!"
			}

/api/classes	GET	**Request Parameters:** The endpoint does not expect any URL parameters, query parameters, or request body data. It simply responds to a GET request at the /api/classes route.	[{ "id": 1, "cname": "Mathematics", "cid": "MATH101", "description": "Introduction to Mathematics", "teacher_email": "teacher1@example.com"]

API Endpoints

/api/tasks/:classId	GET	**Path Parameter (req.params):**classId (string): This is the ID of the class for which tasks are being retrieved. It is specified as a part of the URL path.	[
			{
			"id": 1,
			"title": "Task 1",
			"description": "Description for task 1",
			"due_date": "2024-08-30",
			"class_id": 101
			}
			]

/api/students	POST	{
		"firstName": "John",
		"lastName": "Doe",
		"email": "john.doe@example.com",
		"age": 20,
		"classId": 101,
		"grades": {
		"math": 95,
		"science": 88
		}
		}
			Success Response:
			Status Code: 200 OK
			Body: { message: 'Student added successfully' }
			
			Error Response (Invalid classId):
			Status Code: 400 Bad Request Body: { error: 'Invalid classId' }

API Endpoints

/api/classes/:email	GET	This is a URL parameter (specified as :email in the route). It represents the email address of the teacher whose classes you want to retrieve.	[
			{
			"id": 1,
			"cname": "Mathematics",
			"cid": "MATH101",
			"description": "Basic Mathematics",
			"teacher_email": "teacher@example.com"
			}
			]

/api/students/:classId	GET	This is a required parameter in the URL path. It specifies the ID of the class for which you want to retrieve students.	[
			{
			"id": 1,
			"firstName": "John",
			"lastName": "Doe",
			"email": "john.doe@example.com",
			"age": 20,
			"classId": 10,
			"grades": "A, B, C"
			}
			]

API Endpoints

/api/saveAttendance	POST	[{ "class_id": 101, "student_id": 1, "status": "present", "date": "2024-08-24"}]	Success Response: Status Code: 200 OK JSON Body: { message: 'Attendance saved successfully' } Error Response: Status Code: 500 Internal Server Error JSON Body: { error: 'Failed to save attendance' }

/api/saveHomework	POST	{
		"title": "Math Homework 1",
		"description": "Complete the exercises on page 45",
		"submissionDate": "2024-09-15",
		"classId": "12345"
		}
			Success (200 OK):
			{
			"message": "Homework saved successfully"
			}
			
			Error (500 Internal Server Error):
			{
			"error": "Failed to save homework"
			}

API Endpoints

/api/attendance/:studentId	GET	The ID of the student whose attendance records are being queried. This value is provided in the URL path.	Successful Response :
			[
			{
			"student_id": 123,
			"date": "2024-08-01",
			"status": “here"
			}
			]
			
			Error Response :
			{
			"error": "Internal Server Error"
			}

/api/students/:id	PUT	{
		"grades": {
		"math": 90,
		"english": 85
		}
		}
			Success:
			{
			"message": "Grades updated successfully"
			}
			
			Error :
			{
			"error": "Internal Server Error"
			}

API Endpoints

/api/students	GET	**Query Parameter**: email	**Status Code**: 200 OK
			[
			{
			"id": 1,
			"firstName": "John",
			"lastName": "Doe",
			"email": "student@example.com",
			"age": 20,
			"classId": 101,
			"grades": {
			"math": 95,
			"science": 88
			}
			}
			]
			
			Status Code: 404 Not Found
			{
			"Error": "Student not found"
			}

/api/classes/:cid	GET	**Path Parameter**:
		cid - This is a dynamic path parameter representing the class ID. It is used to identify which class to fetch from the database.
			Status Code: 200 OK
			{
			"className": "<Class Name>"
			}
			
			Database Error:
			{
			"error": "Database error"
			}

API Endpoints

/api/homeworks/:classId	GET	**Path Parameter**: classId	[{ "id": 1, "class_id": 101, "file": "homework1.pdf", "due_date": "2024-09-01" }]

/api/submitHomework	POST	**Request Body**:
		homeworkId (string): Provided as part of the form-data in the request body.
		File:
		file (File): Uploaded with the request and accessible via req.file.
			Success Response:
			{
			"message": "Homework submitted successfully"
			}