

Task Management Application

Name : **Madham Mounika**

EmpID : **70983735**

Given Task : The project revolves around creating a real-time task management application that allows users to create, edit, delete, and prioritize tasks. Users will also have the ability to collaborate on tasks with other users, making it ideal for team-based project or personal use.

1. Users Create tasks with due dates, descriptions, and priorities
2. Edit and update tasks.
3. Mark tasks as complete.
4. Delete tasks.
5. Share tasks with other users
6. Assign tasks to the team members
7. Dashboard to see the overview of the tasks
8. Filters tasks based on status, date
9. Users can search tasks based on keywords

Technologies Used:

- **Frontend** : Angular(VScode)
- **Backend** : Spring Boot(IntelliJ)
- **Database** : PostgreSQL
- **API** : Swagger, postman

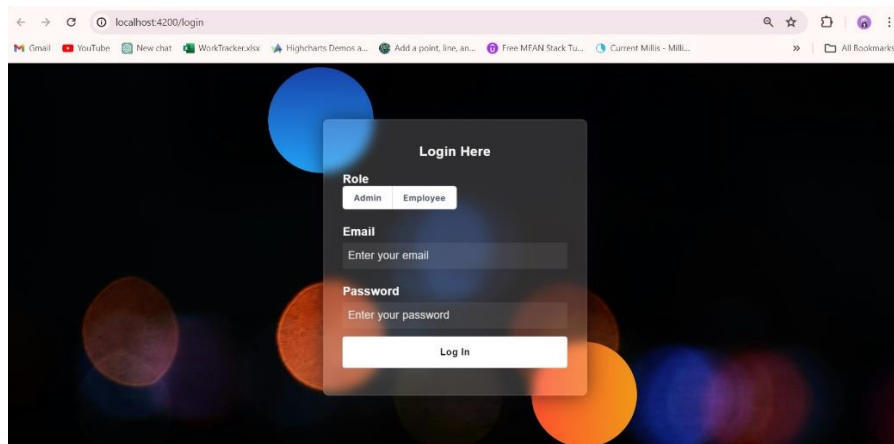
Configurations:

- Angular CLI: 17.3.5
- Angular: 17.3.5
- Node: 20.12.2
- Swagger:2
- java 17
- spring boot 2.4.5

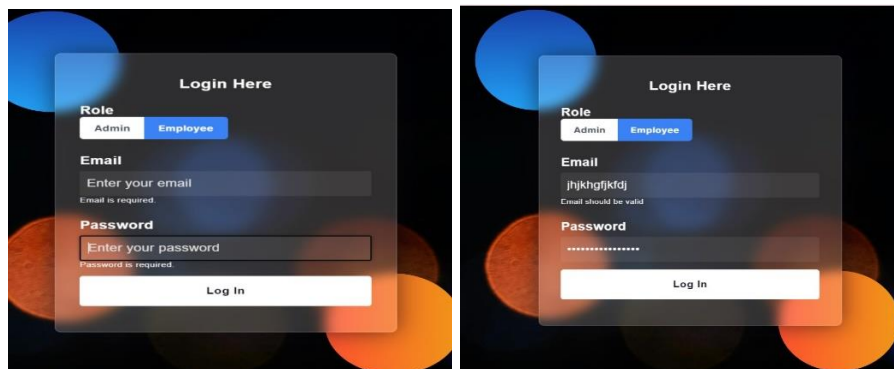
Implementation Part:

Frontend:

- Only admin can able to register the employee & admin credentials are automatically loaded in database.

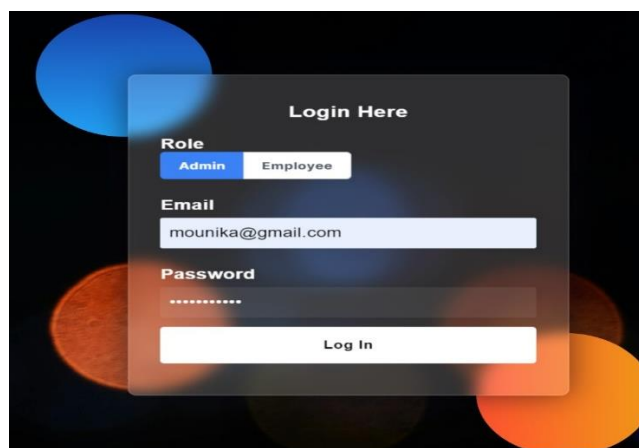


- If login credentials are invalid or not provided then shows the message

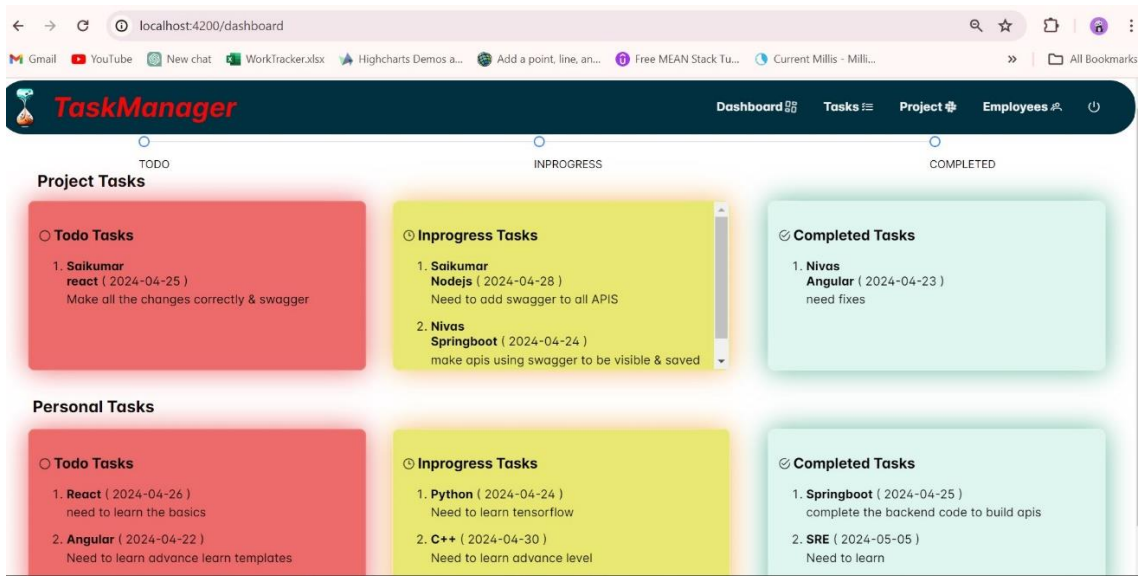


1.Admin Dashboard/ Manager Dashboard:

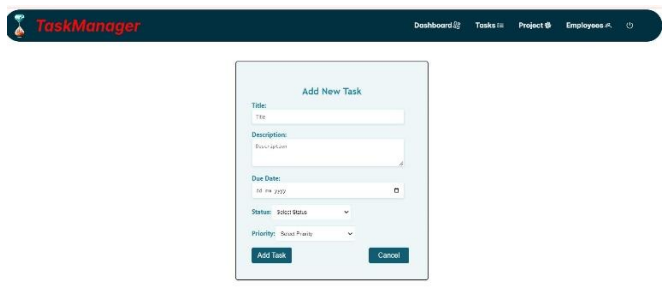
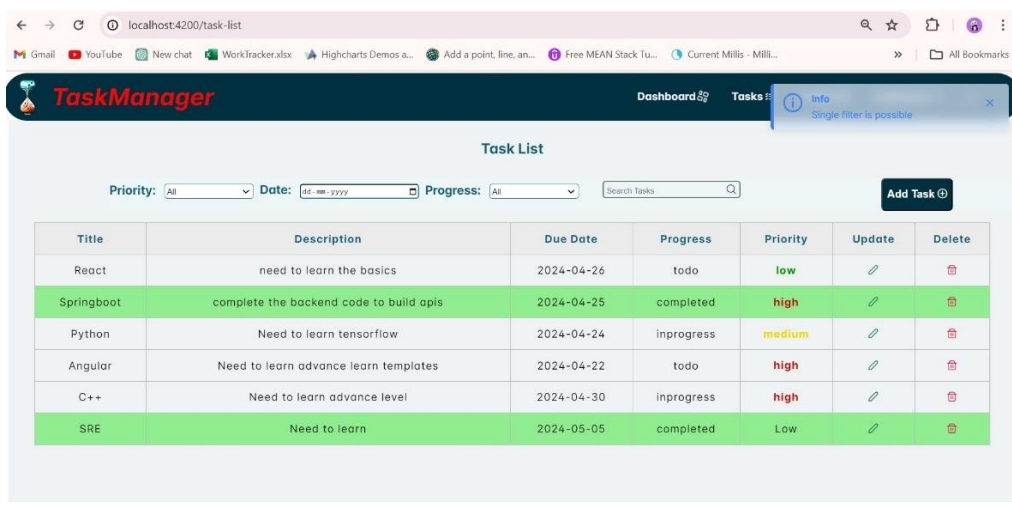
- Credentials(admin,mounika@gmail.com,Hitachi@007)



- After admin successfully logged in , he/she can able to see dashboard with over view of tasks organized in such a way that his/her personally added tasks & himself/herself added tasks to other employees in the project tasks.



- In Tasks, he/she can add their personal tasks as a goal settings.(add, edit, update, delete)
- Here based on the filters user can filter the data based on the Priority, Status, Date, Search.



- Next Page is Project List, admin only can add the tasks & assign to the individual employee tasks.
- Admin can only able to edit all the details, admin can only have permission to delete the tasks.
- Here based on the filters user can filter the data based on the Progress, Employee, Date, Search

TaskManager Dashboard Tasks Project Employees

Project List View Projects Add Task/Project

Employee: All Date: dd-mm-yyyy Progress: All Search Tasks

Employee ID	FullName	ProjectName	Title	Description	Due Date	Progress	Priority	Actions
70983734	Saikumar	Meerkat	Terraform	Need to create terraform for ecs	2024-04-30	inprogress	Medium	Edit Delete
70983734	Saikumar	Meerkat	Nodejs	Need to add swagger to all APIS	2024-04-28	Inprogress	Low	Edit Delete
70983734	Saikumar	Meerkat	react	Make all the changes correctly & swagger	2024-04-25	Todo	Low	Edit Delete
70983740	Nivas	Meerkat	Springboot	make apis using swagger to be visible & saved	2024-04-24	Inprogress	High	Edit Delete
70983740	Nivas	Meerkat	Angular	need fixes	2024-04-23	Completed	Medium	Edit Delete

- For assigning task to employee

TaskManager Dashboard Tasks Info Only admin can add tasks

Post Task

Title:

Description:

Due Date: dd-mm-yyyy

Priority:

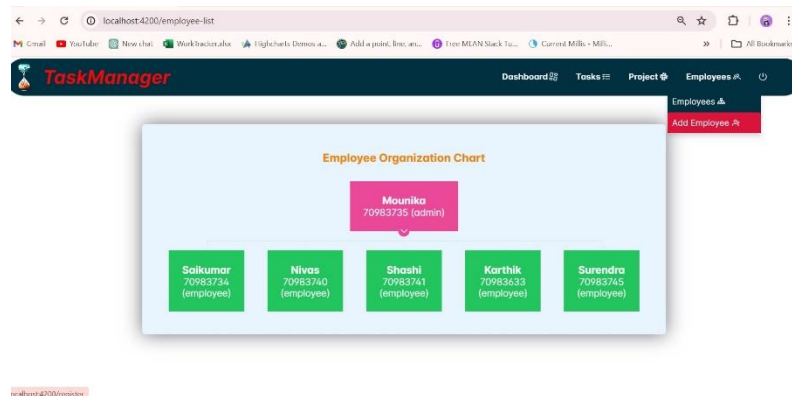
Select Employee:

Status:

ProjectName:

- Admin give the title mean short name for task, description, deadline, priority(high, medium, low), selecting employee, status(todo, inprogress, complete) to assign.

- Next page is employee chart to view all the employees in organization format.



- Admin have permission to add employee into the project.

Register Form

Role: Email:

Employee Id: Password:

Employee FullName: Confirm Password:

Warn: role employee fixed

Info: Only admin can add employee

- Next page is to view the profile & have the option to update & logout in 2ways.

Profile

HITACHI
Web Developer

Profile

Update

Logout

IDENTITY

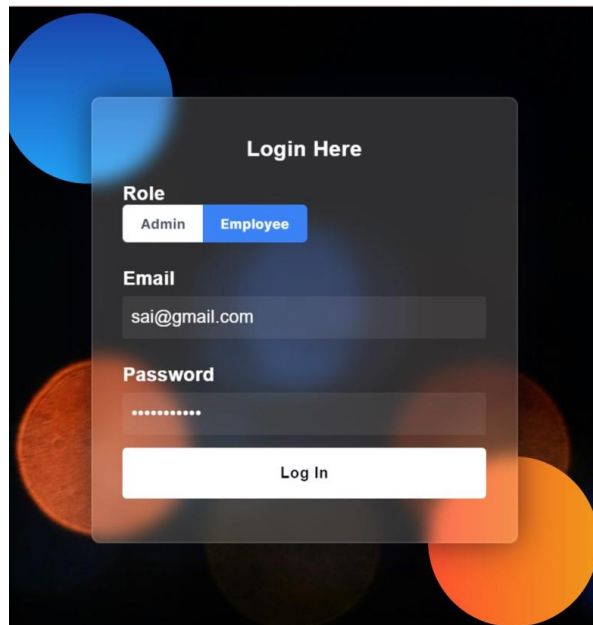
Employee ID : 70983735
Full Name : Mounika
Email : mounika@gmail.com
Role : admin
Password : Hitachi@007
Job : Web Developer
Skill : Full Stack, HTML, CSS, Java, Angular, Springboot

About me:

I design and develop services for customers of all sizes, specializing in creating stylish, modern websites, web services and online stores. My passion is to design digital user experiences through the bold interface and meaningful interactions.

2. Employee Dashboard:

- Login page Credentials(employee, sai@gmail.com, Hitachi@007)
- This credentials are created by the admin.



- After login, dashboard is same as admin displaying personal & project tasks.
- Here personal tasks are ones tasks & project tasks are admin created one.
- After that coming to the Personal Tasks can add ,delete, update, get same & can have filters to filter data based on priority, status, date, search.
- Coming to next page Project list, employee can have all tasks assigned by the admin to employees.

localhost:4200/project-list

TaskManager

Dashboard Tasks Project Employees

View Projects

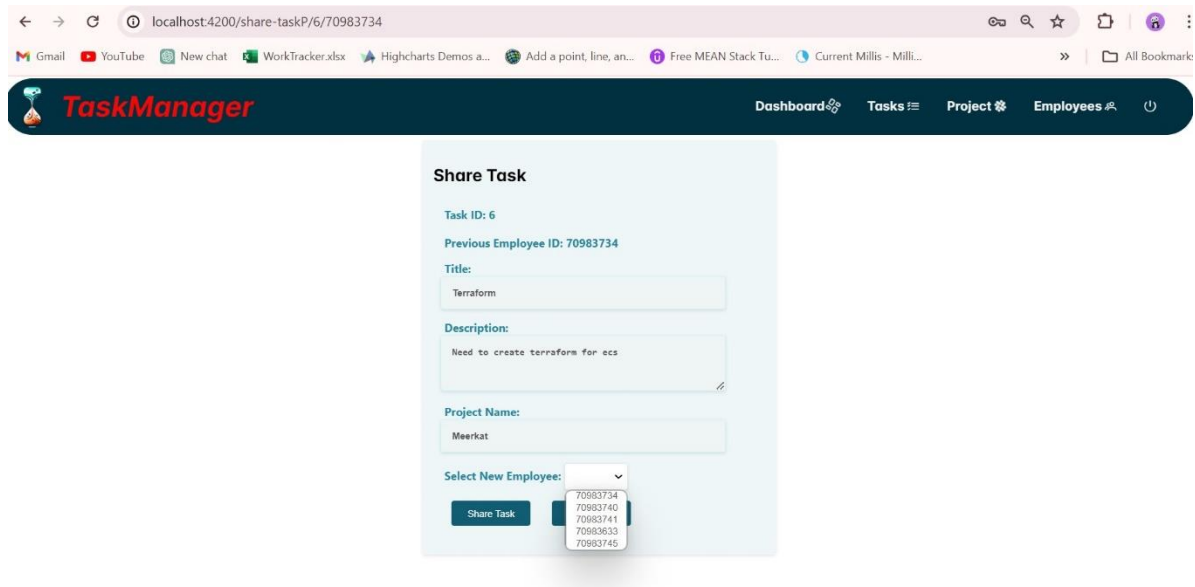
Project List

Employee: All Date: dd-mm-yyyy Progress: All Search Tasks

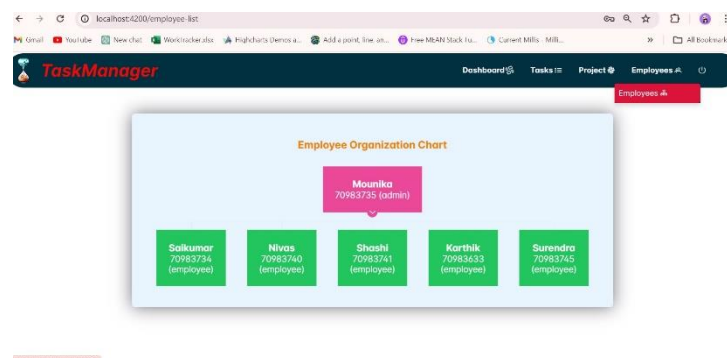
Employee ID	FullName	ProjectName	Title	Description	Due Date	Progress	Priority	Actions	Share
70983734	Saikumar	Meerkat	Terraform	Need to create terraform for ecs	2024-04-30	Inprogress	Medium		
70983734	Saikumar	Meerkat	Nodejs	Need to add swagger to all APIS	2024-04-28	Inprogress	Low		
70983734	Saikumar	Meerkat	react	Make all the changes correctly & swagger	2024-04-25	Todo	Low		
70983740	Nivas	Meerkat	Springboot	make apis using swagger to be visible & saved	2024-04-24	Inprogress	High		
70983740	Nivas	Meerkat	Angular	need fixes	2024-04-23	Completed	Medium		

localhost:4200/project-list

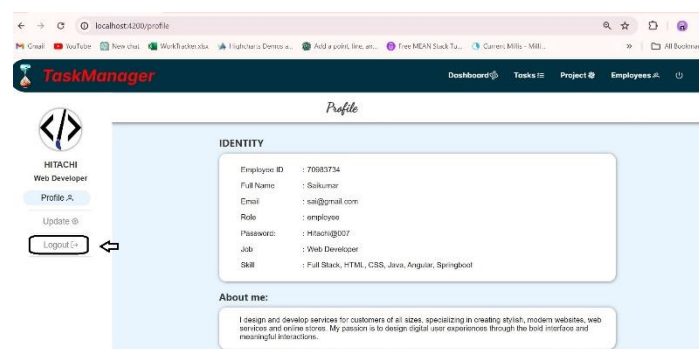
- Here employee can able to edit the progress of employee's tasks .
- Here employee have option to share the complete task details to other employee using share option.



- Once shared that task, that task can be displayed to admin with employee credentials.
- After that coming to Employee list page, employee can have option only to view all the employees .

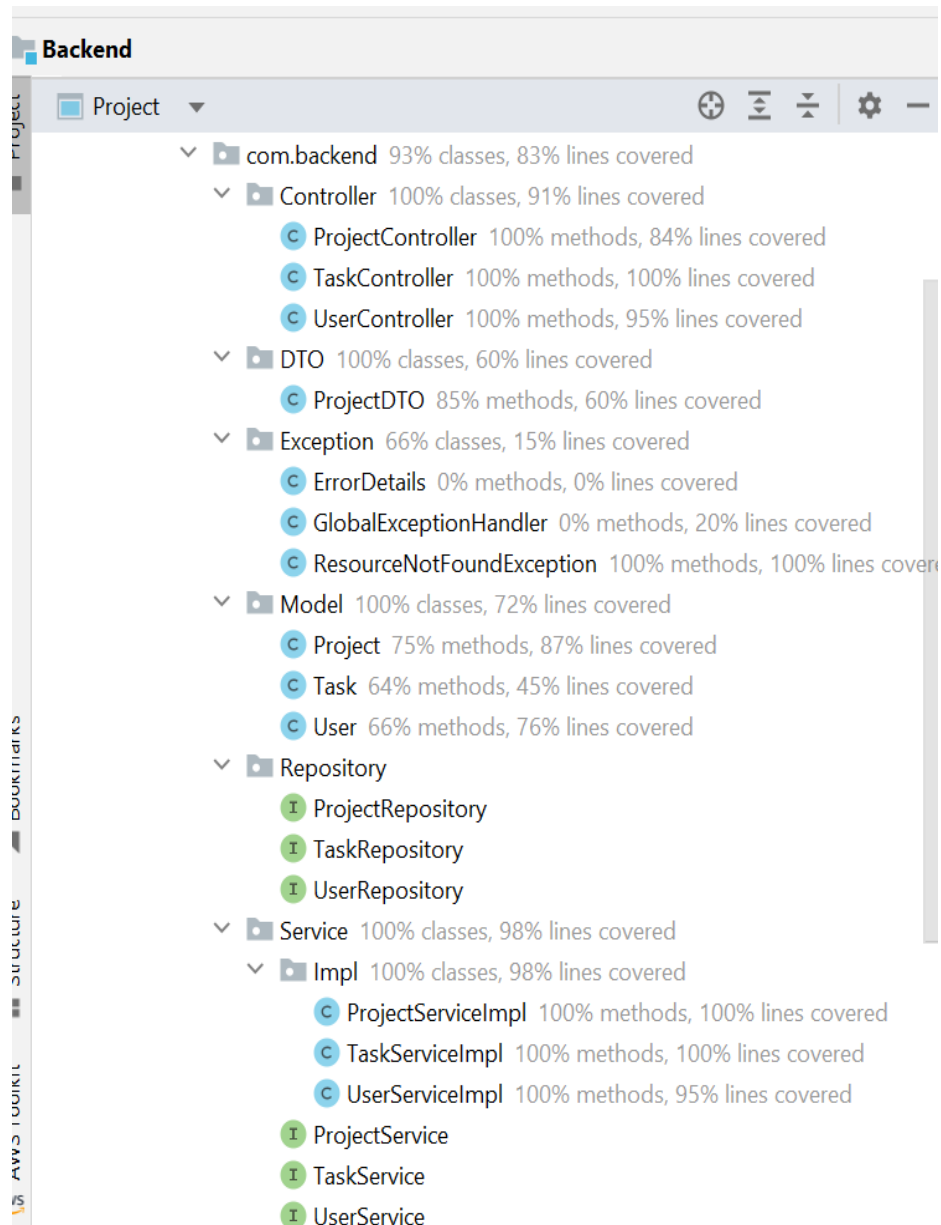


- Next page is profile with all employee data & logout option.



Backend:

Project Structure:



- **Database with 3 tables :**

1. Employee Table :

Procedures

1.3 Sequences

Tables (3)

employees

Columns

Constraints

Indexes

RLS Policies

Rules

Triggers

project_tasks

tasks

Data OutputMessagesNotifications

	id [PK] bigint	eid bigint	email character varying (255)	ename character varying (255)	password character varying (255)	role character varying (255)
1	1	70983735	mounika@gmail.com	Mounika	Hitachi@007	admin
2	2	70983734	sai@gmail.com	Saikumar	Hitachi@007	employee
3	3	70983740	nivas@gmail.com	Nivas	Hitachi@007	employee
4	4	70983741	shashi@gmail.com	Shashi	Hitachi@007	employee
5	8	70983633	karthik@gmail.com	Karthik	Hitachi@007	employee
6	9	70983745	surendra@gmail.com	Surendra	Hitachi@007	employee

2. Personal Tasks Table:

Materialized Views

Operators

Procedures

1.3 Sequences

Tables (3)

employees

project_tasks

tasks

Columns

Constraints

Indexes

RLS Policies

Rules

Triggers

Trigger Functions

Data OutputMessagesNotifications

	id [PK] bigint	description character varying (255)	date timestamp without time zone	eid bigint	priority character varying (255)	status character varying (255)	title character varying (255)
1	1	Need to learn advance learn templates	2024-04-22 05:30:00	70983735	high	todo	Angular
2	2	Need to learn advance level	2024-04-30 05:30:00	70983735	high	inprogress	C++
3	3	Need to learn tensorflow	2024-04-24 05:30:00	70983735	medium	inprogress	Python
4	4	To learn web UI framework react from basics	2024-04-30 05:30:00	70983734	high	todo	React
5	5	To learn nodejs, express as backend framework	2024-05-04 05:30:00	70983734	medium	completed	Nodejs
6	6	learn the SRE training	2024-04-30 00:00:00	70983740	high	inprogress	SRE
7	7	need to create apis for frontend angular	2024-04-28 05:30:00	70983740	high	inprogress	Springboot
8	8	need to learn the basics	2024-04-26 00:00:00	70983735	low	todo	React
9	9	complete the backend code to build apis	2024-04-25 00:00:00	70983735	high	completed	Springboot
10	10	Need to learn	2024-05-05 05:30:00	70983735	Low	completed	SRE

3.Project Tasks Table:

1.3 Sequences

Tables (3)

employees

project_tasks

Columns

Constraints

Indexes

RLS Policies

Rules

Triggers

Data OutputMessagesNotifications

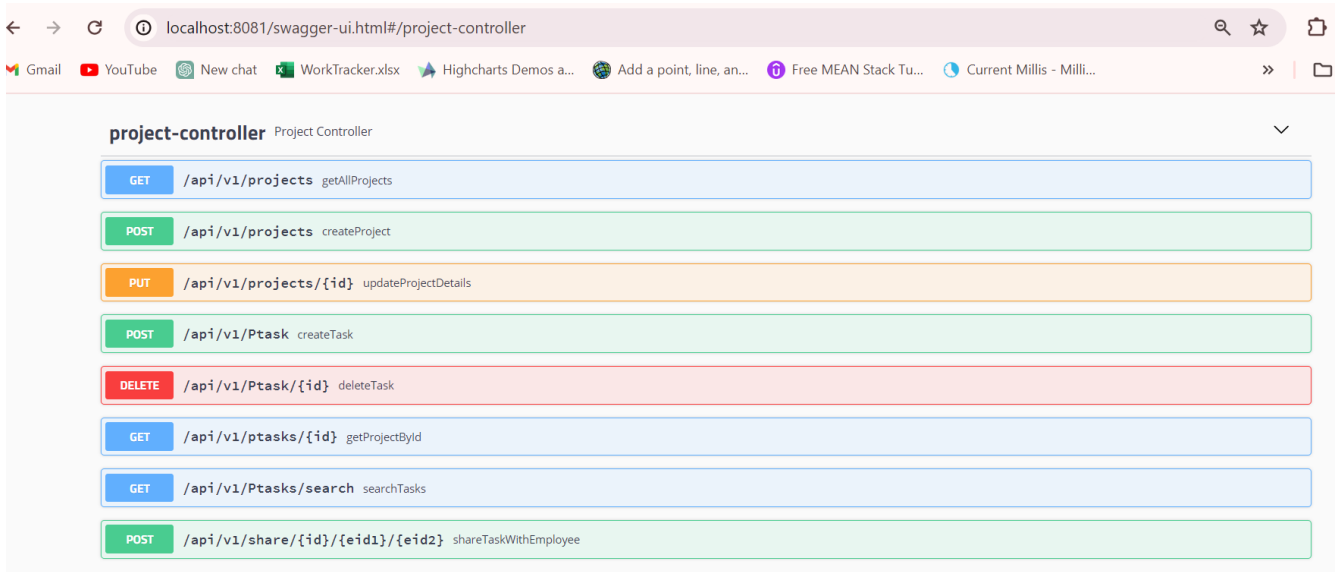
	id [PK] bigint	description character varying (255)	due_date timestamp without time zone	priority character varying (255)	status character varying (255)	title character varying (255)	eid bigint	project_name character varying (255)
1	2	make apis using swagger to be visible & saved	2024-04-24 05:30:00	High	Inprogress	Springboot	70983740	Meerkat
2	3	Make all the changes correctly & swagger	2024-04-25 05:30:00	Low	Todo	react	70983734	Meerkat
3	4	need fixes	2024-04-23 05:30:00	Medium	Completed	Angular	70983740	Meerkat
4	5	Need to add swagger to all APIs	2024-04-28 05:30:00	Low	Inprogress	Nodejs	70983734	Meerkat
5	6	Need to create terraform for ecs	2024-04-30 05:30:00	Medium	inprogress	Terraform	70983734	Meerkat

Documentation for APIs using tools like swagger, open api:

- Using the swagger doc can get access to all apis in controllers from project.

<http://localhost:8081/v2/api-docs>

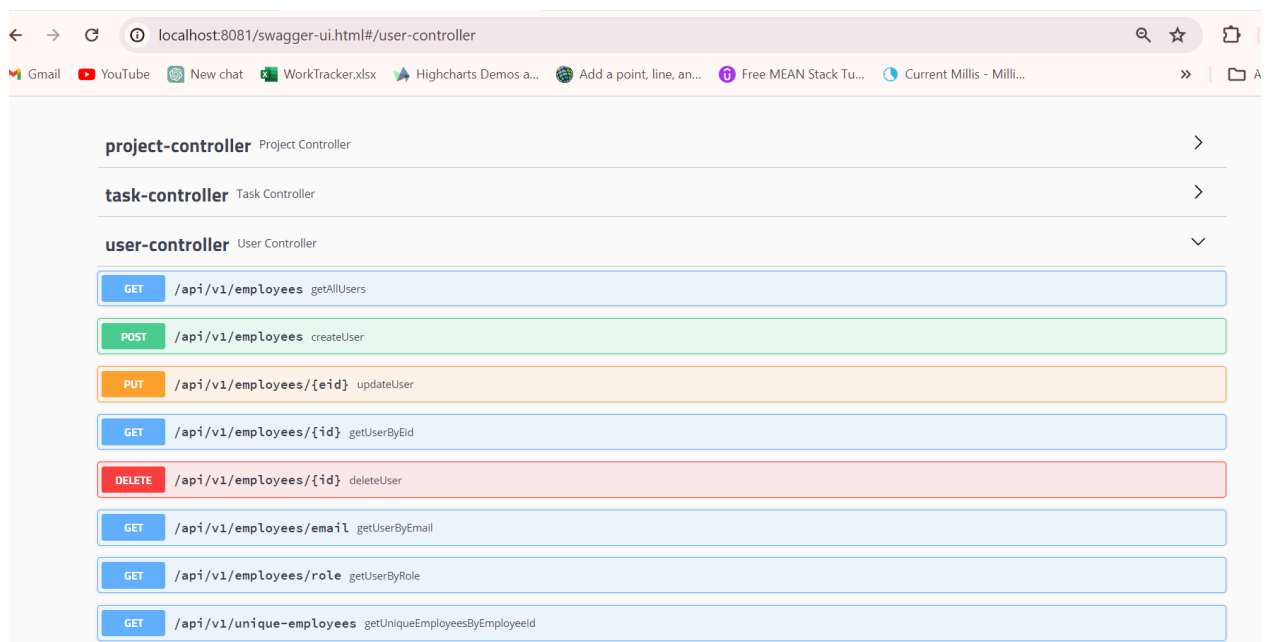
1. Project Controller (Project tasks) :



The screenshot shows the Swagger UI for the 'project-controller' API. The browser address bar displays 'localhost:8081/swagger-ui.html#/project-controller'. The page lists the following endpoints:

Method	Endpoint	Description
GET	/api/v1/projects	getAllProjects
POST	/api/v1/projects	createProject
PUT	/api/v1/projects/{id}	updateProjectDetails
POST	/api/v1/Ptask	createTask
DELETE	/api/v1/Ptask/{id}	deleteTask
GET	/api/v1/ptasks/{id}	getProjectById
GET	/api/v1/Ptasks/search	searchTasks
POST	/api/v1/share/{id}/{eid1}/{eid2}	shareTaskWithEmployee

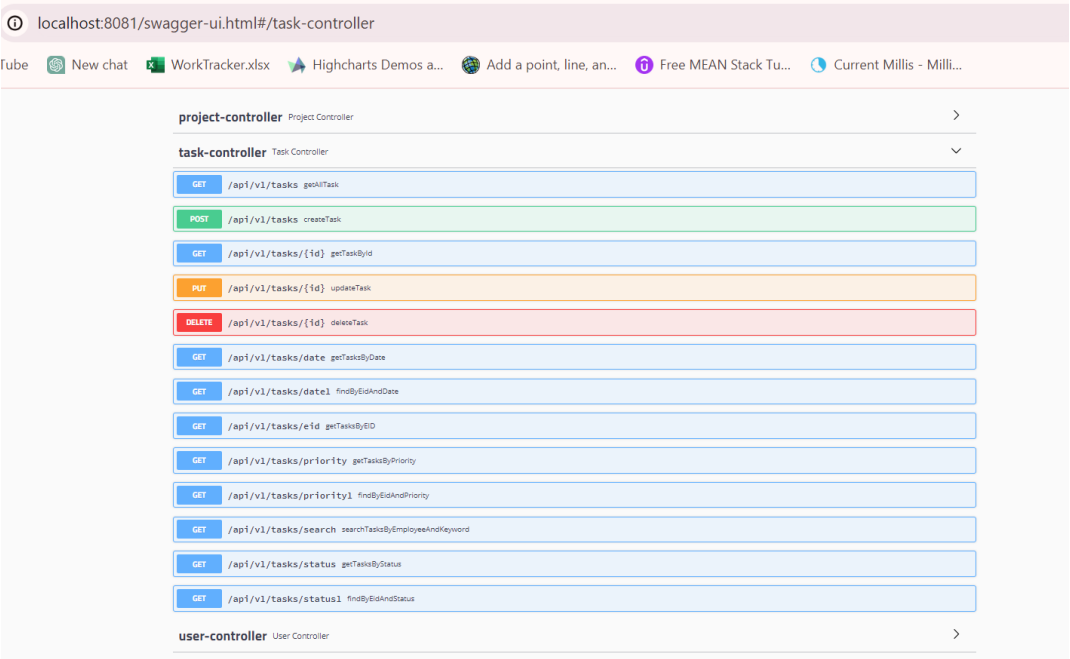
2. User Controller :



The screenshot shows the Swagger UI for the 'user-controller' API. The browser address bar displays 'localhost:8081/swagger-ui.html#/user-controller'. The page lists the following endpoints:

Method	Endpoint	Description
GET	/api/v1/employees	getAllUsers
POST	/api/v1/employees	createUser
PUT	/api/v1/employees/{eid}	updateUser
GET	/api/v1/employees/{id}	getUserById
DELETE	/api/v1/employees/{id}	deleteUser
GET	/api/v1/employees/email	getUserByEmail
GET	/api/v1/employees/role	getUserByRole
GET	/api/v1/unique-employees	getUniqueEmployeesByEmployeeid

3.Task Controller (Personal Tasks):



Test Cases Coverage:

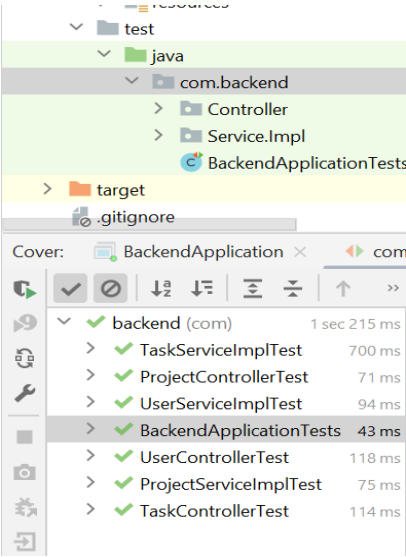
1.FRONTEND:

==== Coverage summary =====

Statements	: 61.4% (315/513)
Branches	: 30.9% (34/110)
Functions	: 56.83% (104/183)
Lines	: 60% (288/480)

====

2.BACKEND (JUnit testing) :



Element	Class, %	Method, %	Line, %
com.backend.Controller	100% (3/3)	100% (29/29)	91% (145/1...
UserController	100% (1/1)	100% (8/8)	95% (46/48)
TaskController	100% (1/1)	100% (13/13)	100% (39/39)
ProjectController	100% (1/1)	100% (8/8)	84% (60/71)

Element	Class, %	Method, %	Line, %
com.backend.Service.Impl	100% (3/3)	100% (26/26)	98% (67/68)
ServiceImpl	100% (1/1)	100% (5/5)	95% (20/21)
TaskServiceImpl	100% (1/1)	100% (13/13)	100% (19/19)
ProjectServiceImpl	100% (1/1)	100% (8/8)	100% (28/28)