WIPRO NGA Program – 25SUB4530_CP_JAVA

Capstone Project Presentation – 9th Feb & 10th Feb 2026

Project Title – Task/To-Do Management Application

Presented by – Devara Vasmitha
Dole Madhu Sri

www.rpsconsulting.in

Objective of the Project

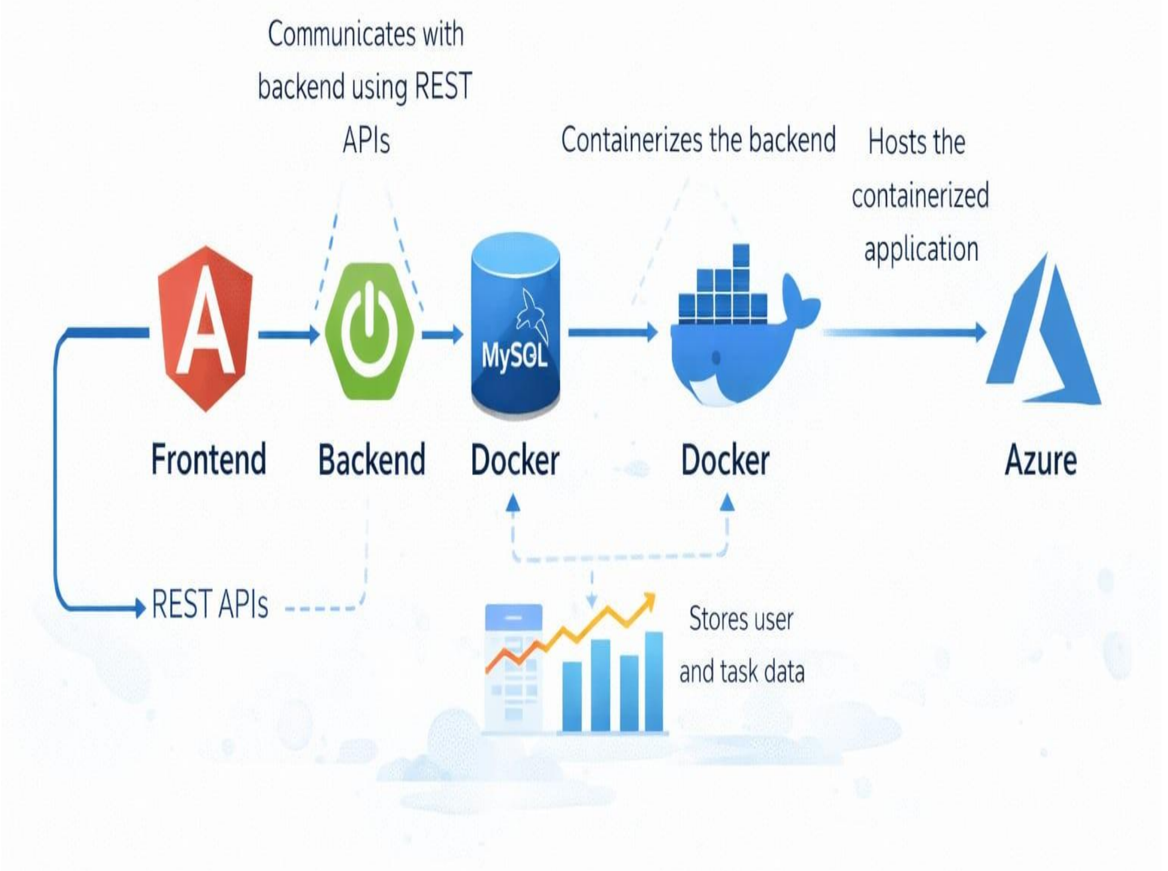
- Improve task organization and productivity
- Provide clear visibility of tasks
- Enable users to manage tasks efficiently
- Demonstrate full-stack development skills
- Implement DevOps concepts like Docker and Azure

Technology Stack

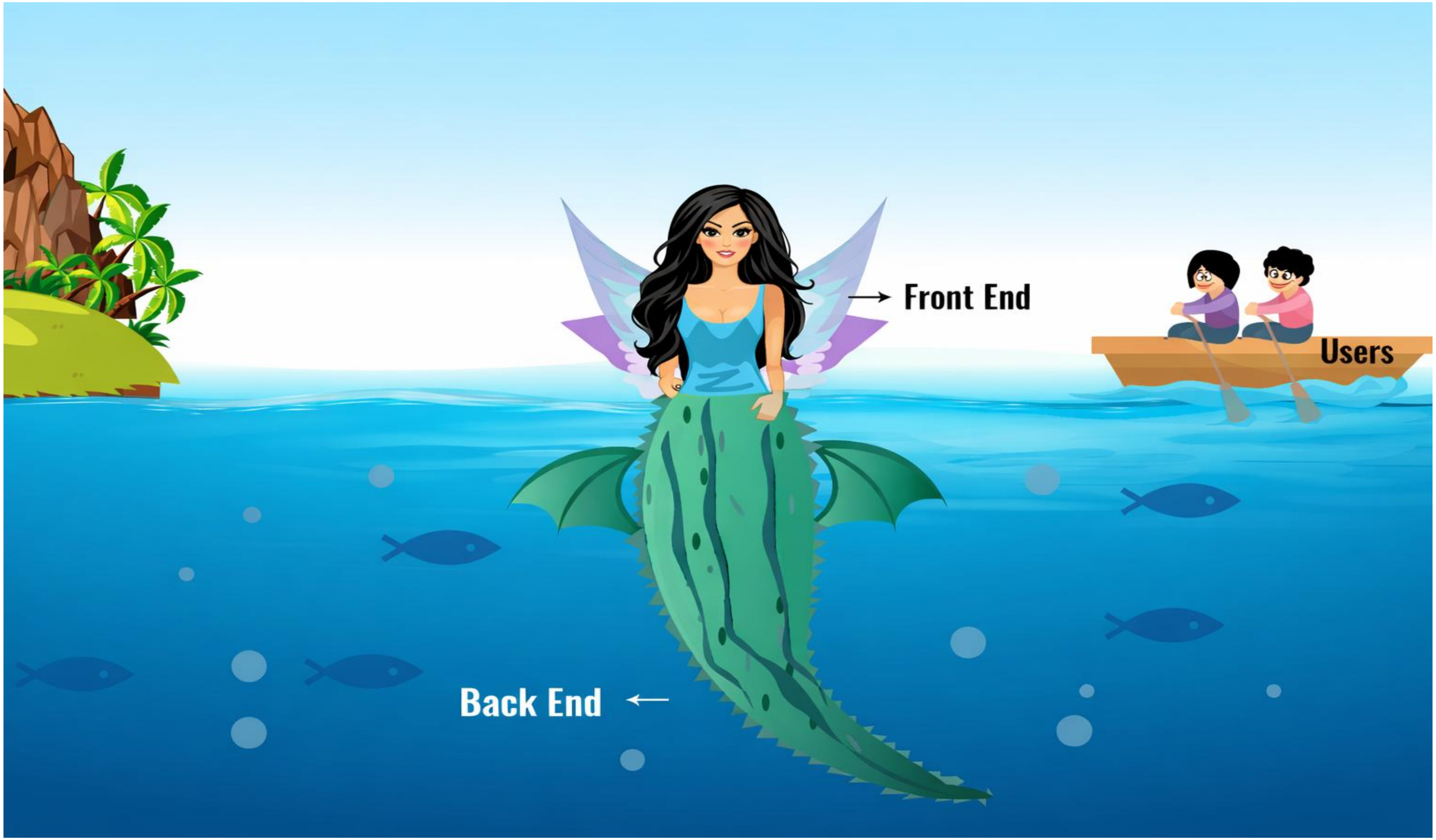
- **Frontend:**
  Angular
  HTML, CSS, TypeScript
- **Backend:**
  Spring Boot
  Spring Data JPA
- **Database:**
  MySQL
- **DevOps & Cloud:**
  Docker
  Azure Container Instances

System Architecture

- Angular frontend communicates with backend using REST APIs
- Spring Boot backend handles business logic
- Spring Data JPA manages database operations
- MySQL stores user and task data
- Docker containerizes the backend
- Azure hosts the containerized application



Communicates with backend using REST APIs
Containerizes the backend
Hosts the containerized application
MySQL
Frontend
Backend
Docker
Docker
Azure
REST APIs
Stores user and task data

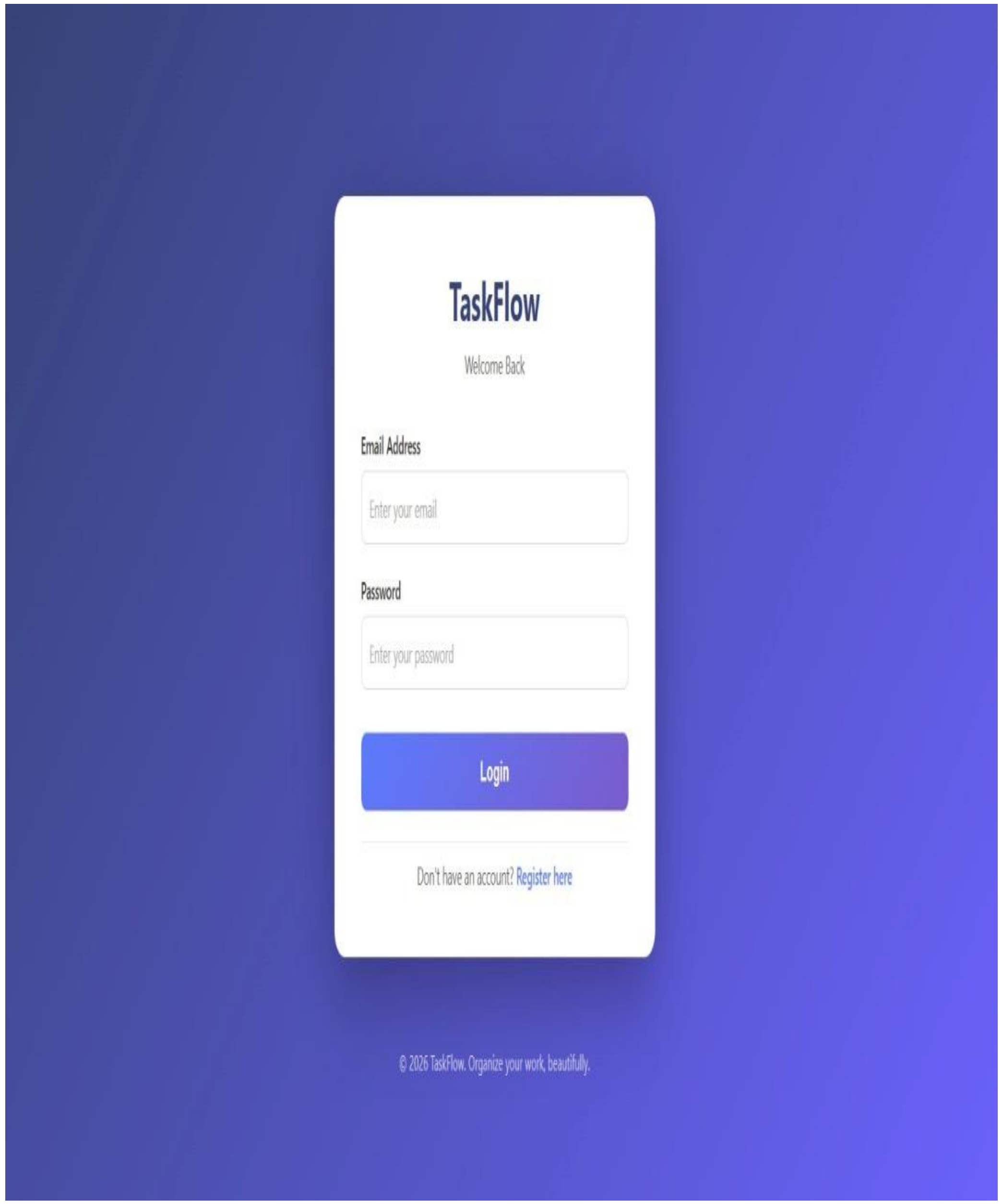

Front End
Users
Back End

Frontend Implementation

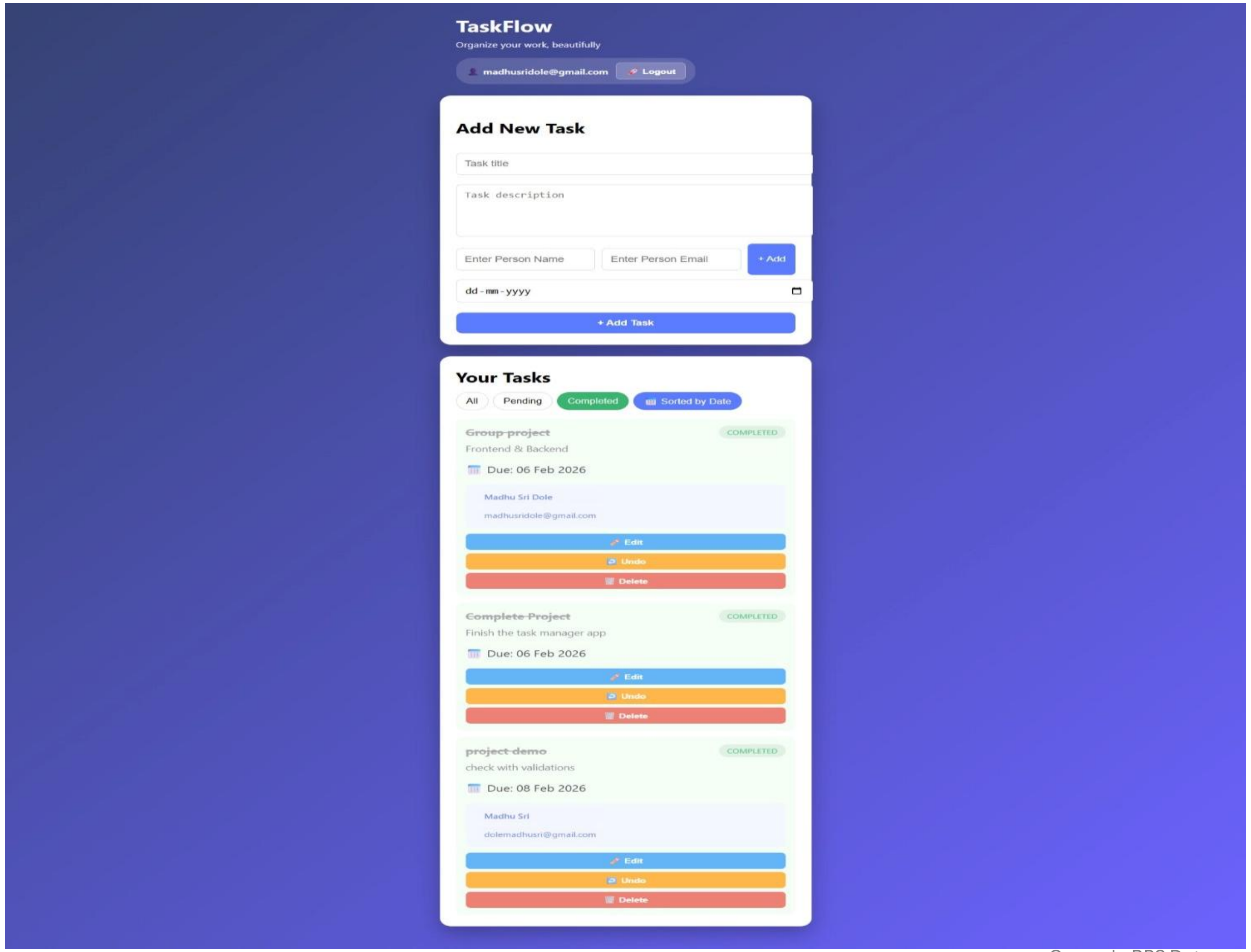
- Developed using Angular
- Responsive UI with clean layout
- Forms for task creation and update
- Filters for task status
- Includes filters to view tasks based on status
- Communicates with backend using HttpClient
- Ensues smooth user interaction and real-time data updates

Core Functionalities

- User login and authentication
- Create new tasks
- Edit existing tasks
- Delete tasks
- Mark tasks as **Completed / Pending**
- Filter tasks by status
- Sort tasks by due date


TaskFlow
Welcome Back
Email Address
Enter your email
Password
Enter your password
Login
Don't have an account? Register here
© 2026 TaskFlow. Organize your work, beautifully.

Outputs:



TaskFlow

Organize your work, beautifully

madhusridole@gmail.com

Add New Task

Task title

Task description

Enter Person Name

Enter Person Email

dd-mm-yyyy

Your Tasks

~~Group project~~ COMPLETED

Frontend & Backend

Due: 06 Feb 2026

Madhu Sri Dole

madhusridole@gmail.com

~~Complete Project~~ COMPLETED

Finish the task manager app

Due: 06 Feb 2026

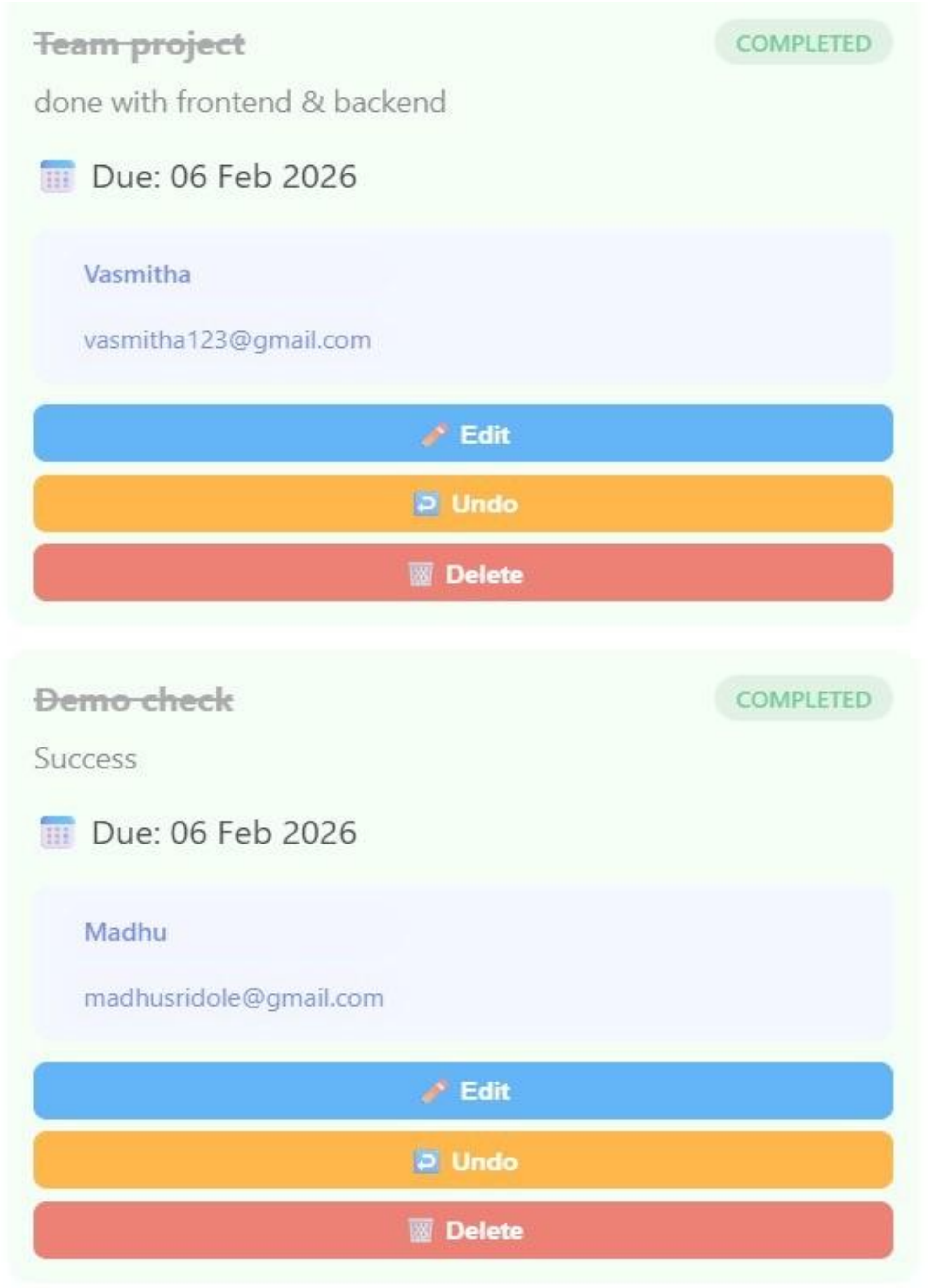
~~project demo~~ COMPLETED

check with validations

Due: 08 Feb 2026

Madhu Sri

dolemadhusri@gmail.com



~~Team project~~ COMPLETED

done with frontend & backend

Due: 06 Feb 2026

Vasmitha

vasmitha123@gmail.com

~~Demo check~~ COMPLETED

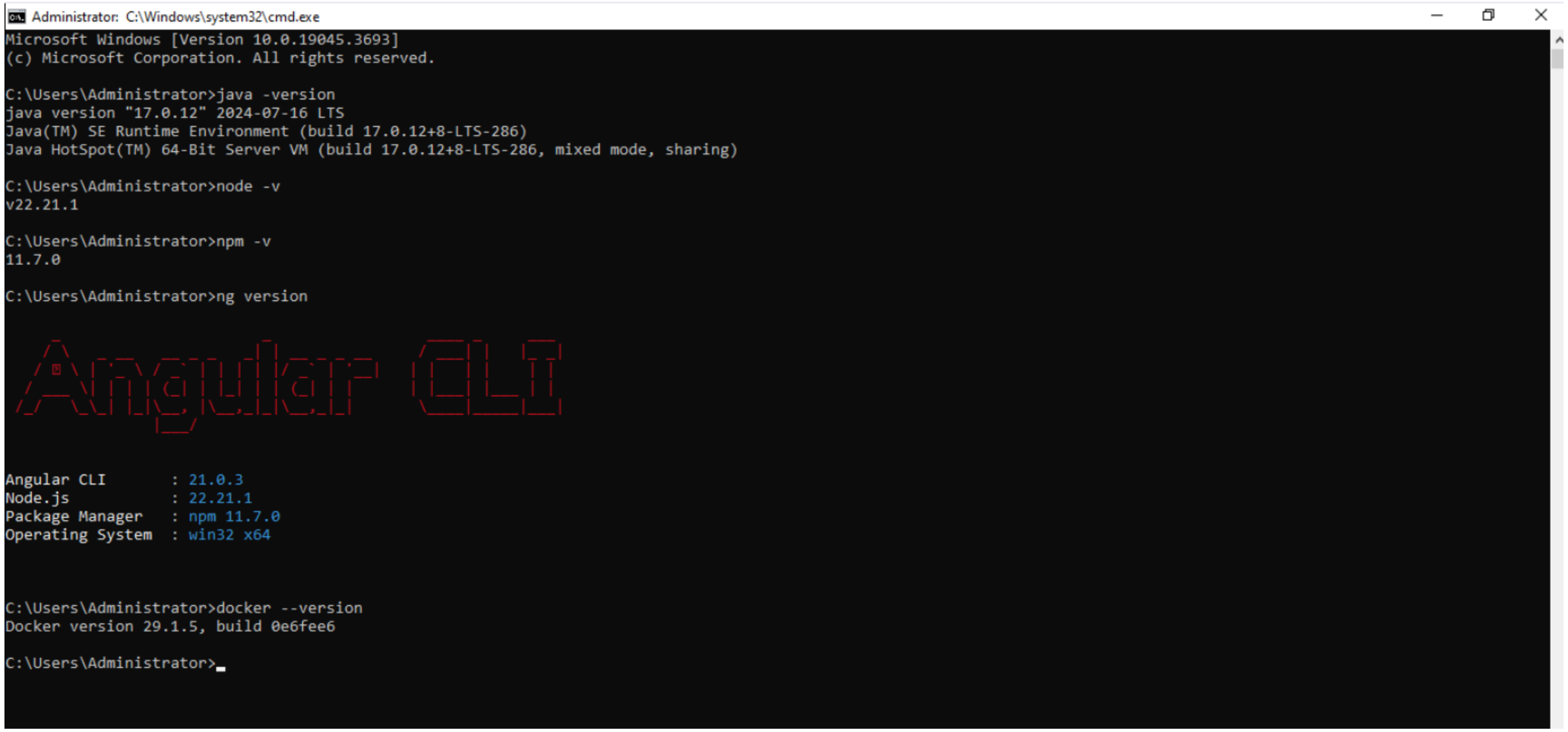
Success

Due: 06 Feb 2026

Madhu

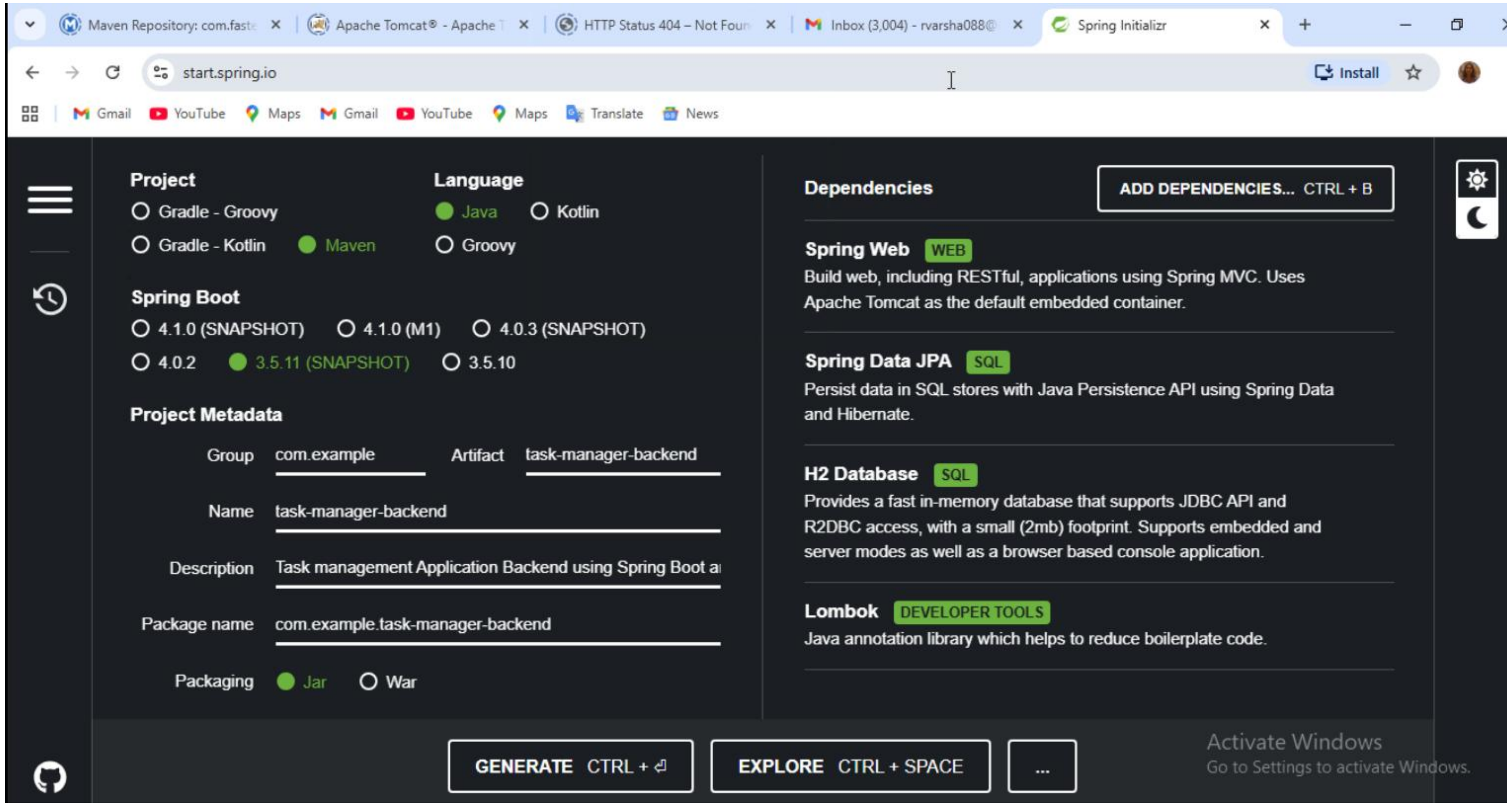
madhusridole@gmail.com

Backend and Frontend Versions


Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>java -version
java version "17.0.12" 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 17.0.12+8-LTS-286)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.12+8-LTS-286, mixed mode, sharing)

C:\Users\Administrator>node -v
v22.21.1

C:\Users\Administrator>npm -v
11.7.0

C:\Users\Administrator>ng version

     _                      _                 ____ _     ___
    / \   _ __   __ _ _   _| | __ _ _ __     / ___| |   |_ _|
   / △ \ | '_ \ / _` | | | | |/ _` | '__|   | |   | |    | |
  / ___ \| | | | (_| | |_| | | (_| | |      | |___| |___ | |
 /_/   \_\_| |_|\__, |\__,_|_|\__,_|_|       \____|_____|___|
                |___/


Angular CLI       : 21.0.3
Node.js           : 22.21.1
Package Manager   : npm 11.7.0
Operating System  : win32 x64


C:\Users\Administrator>docker --version
Docker version 29.1.5, build 0e6fee6

C:\Users\Administrator>_
```

Backend Implementation



start.spring.io

Spring Initializr

Project
- Gradle - Groovy
- Gradle - Kotlin
- [✓ Maven]

Language
- [✓ Java]
- Kotlin
- Groovy

Spring Boot
- 4.1.0 (SNAPSHOT)
- 4.1.0 (M1)
- 4.0.3 (SNAPSHOT)
- 4.0.2
- [✓ 3.5.11 (SNAPSHOT)]
- 3.5.10

Project Metadata

Group	com.example
Artifact	task-manager-backend
Name	task-manager-backend
Description	Task management Application Backend using Spring Boot a
Package name	com.example.task-manager-backend
Packaging	[✓ Jar] War

Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

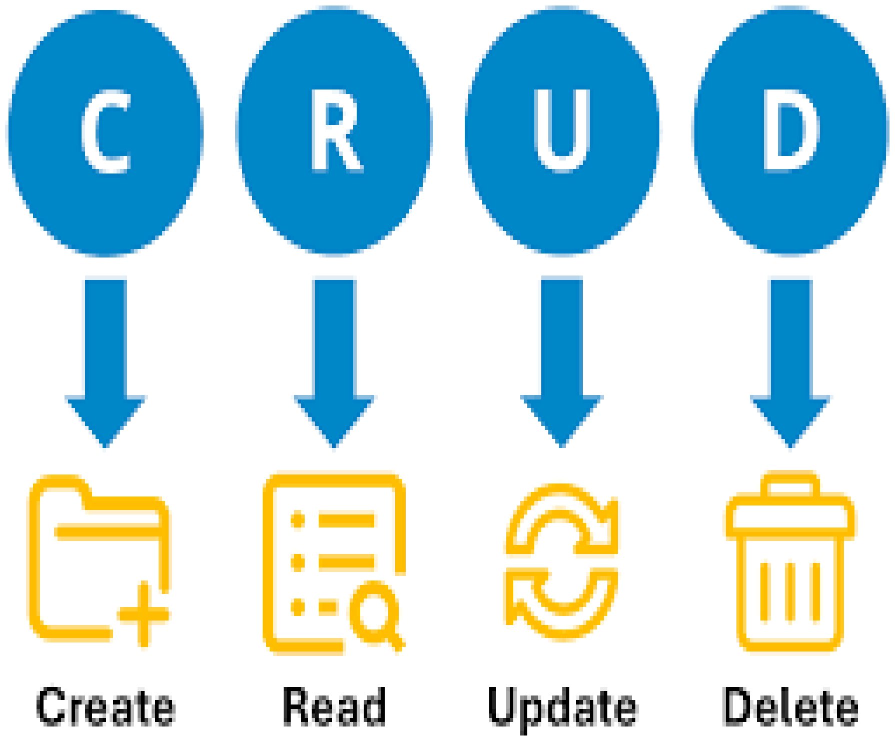
H2 Database SQL
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

Lombok DEVELOPER TOOLS
Java annotation library which helps to reduce boilerplate code.

ADD DEPENDENCIES... CTRL + B
GENERATE CTRL + ⏎
EXPLORE CTRL + SPACE
...
Activate Windows
Go to Settings to activate Windows.

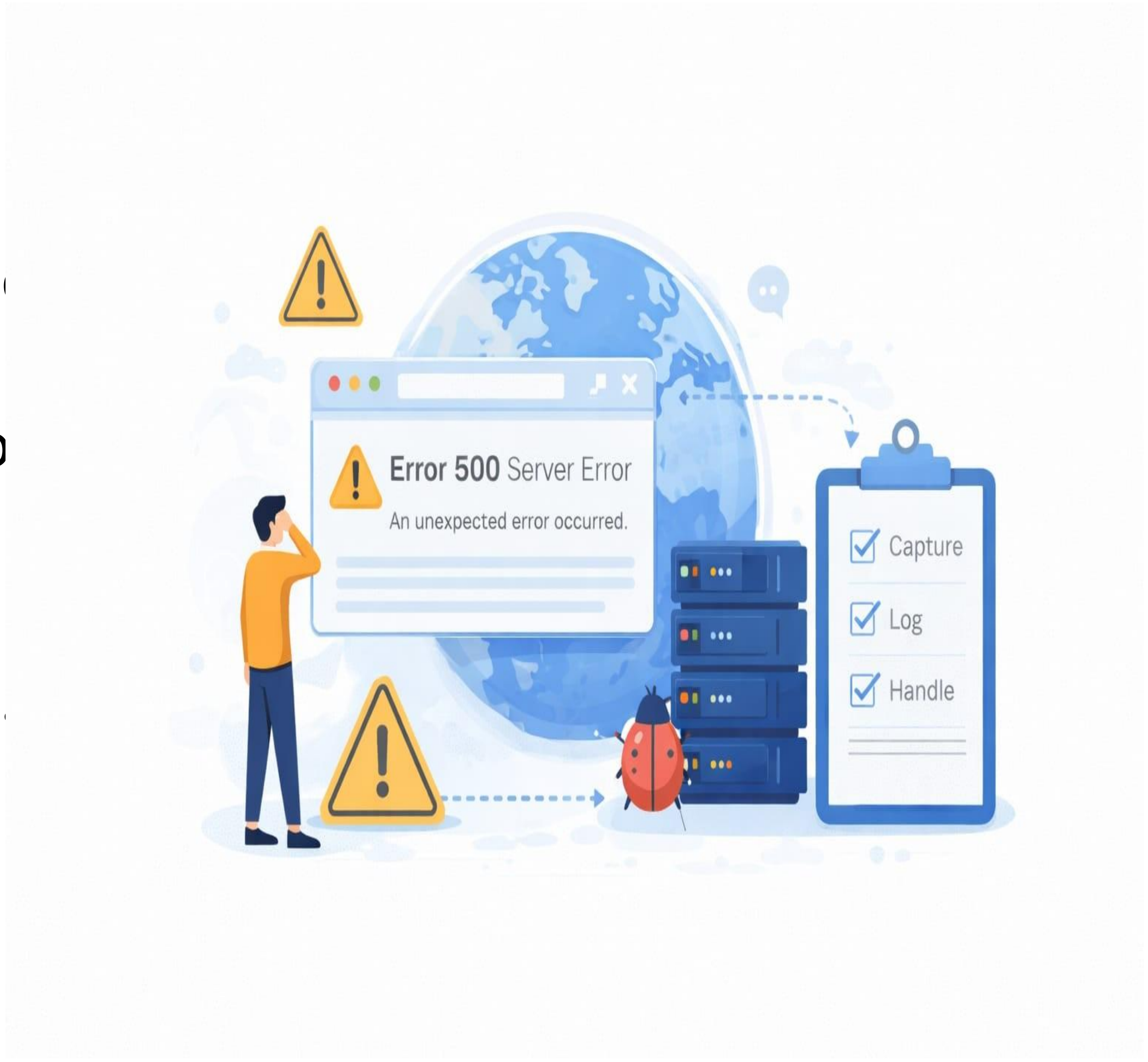
Backend Implementation

- Developed using Spring Boot
- REST APIs created for all CRUD operations
- Controllers handle HTTP requests
- Services contain business logic
- Repositories interact with database
- Uses Spring Data JPA for persistence



Global Exception

- Implemented global exception handling using @ControllerAdvic spring boot
- Handles runtime exceptions acro all REST APIs in the Task Management Application.
- Provides meaningful error messa
- Prevents application crash
- Improves backend reliability ,debugging and overall maintainability



Database Design

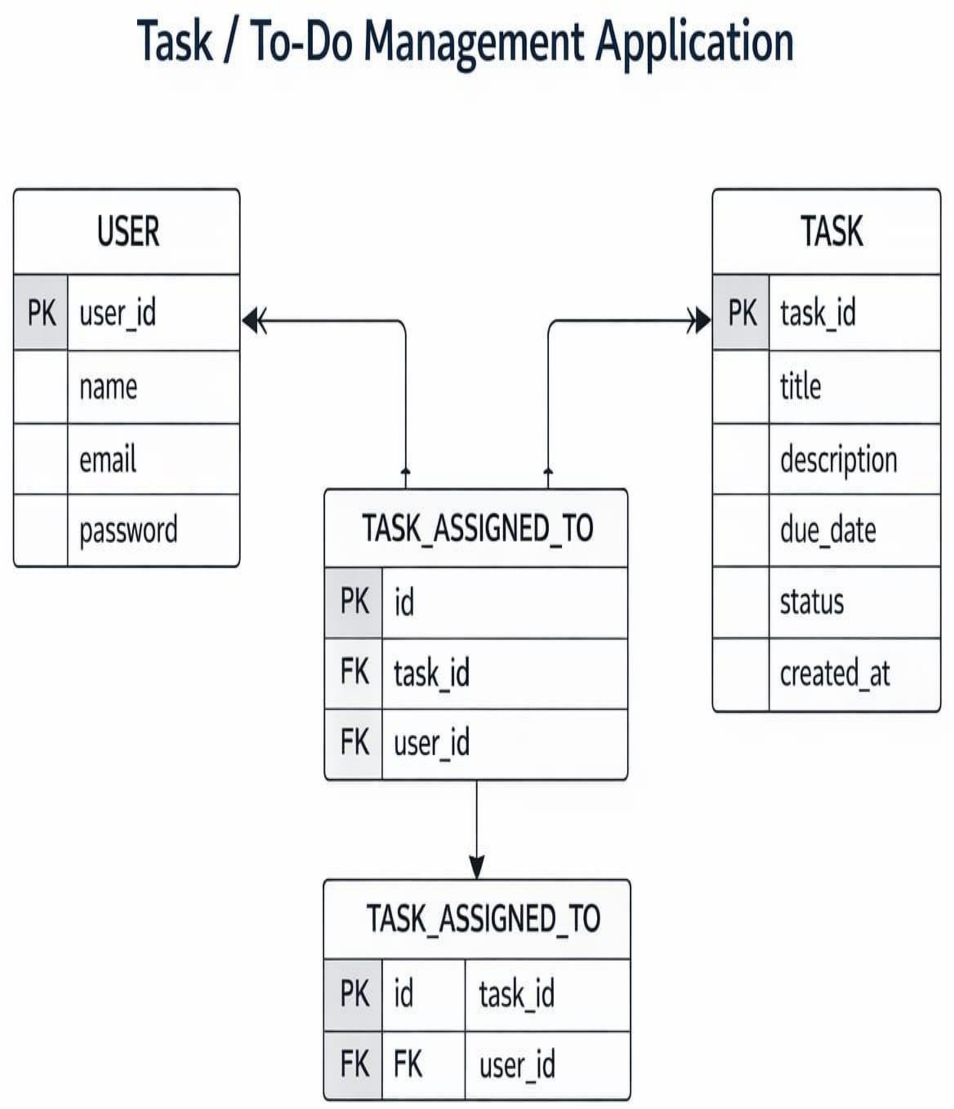
Tables Used:

User

- Task
- Task_Assigned_To

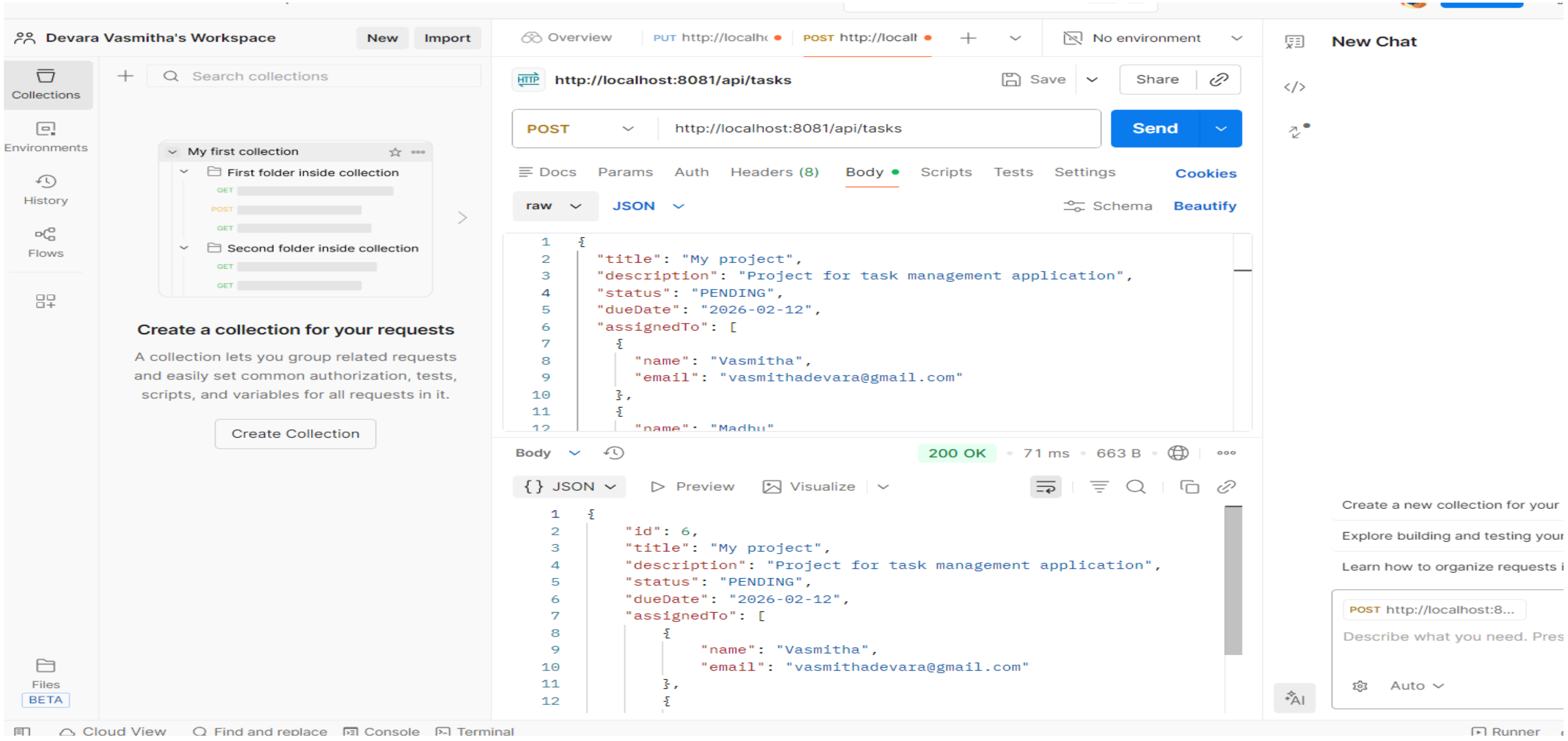
Features:

- One-to-many relationship (User → Tasks)
- Supports task assignment using Task_Assigned_To mapping table
- Stores task title, description, due date, and status
- Uses MySQL for persistent and relational storage



REST API EndPoints

- **POST /api/auth/login – User login**


Devara Vasmitha's Workspace
New
Import
Collections
Environments
History
Flows
Files
BETA
Search collections
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
Overview
PUT http://localh
POST http://localh
No environment
http://localhost:8081/api/tasks
Save
Share
POST
Send
Docs
Params
Auth
Headers (8)
Body
Scripts
Tests
Settings
Cookies
raw
JSON
Schema
Beautify
200 OK
71 ms
663 B
Preview
Visualize
New Chat
Create a new collection for your
Explore building and testing your
Learn how to organize requests i
POST http://localhost:8...
Describe what you need. Pres
Auto
Cloud View
Find and replace
Console
Terminal
Runner

POST http://localhost:8081/api/tasks

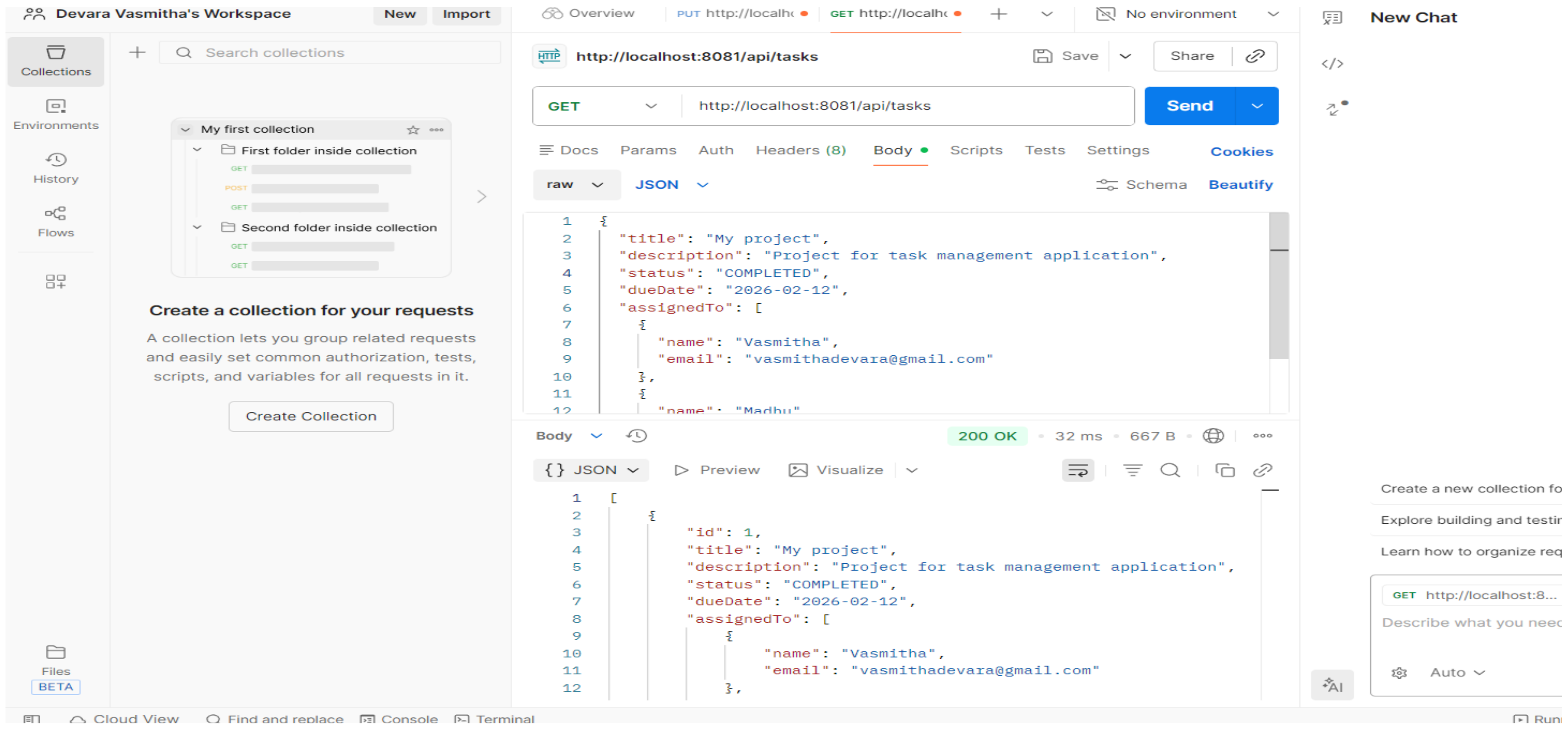
```
{
  "title": "My project",
  "description": "Project for task management application",
  "status": "PENDING",
  "dueDate": "2026-02-12",
  "assignedTo": [
    {
      "name": "Vasmitha",
      "email": "vasmithadevara@gmail.com"
    },
    {
      "name": "Madhu"
```

Body — 200 OK • 71 ms • 663 B

```
{
    "id": 6,
    "title": "My project",
    "description": "Project for task management application",
    "status": "PENDING",
    "dueDate": "2026-02-12",
    "assignedTo": [
        {
            "name": "Vasmitha",
            "email": "vasmithadevara@gmail.com"
        },
        {
```

REST API EndPoints

- **GET /api/task/user – Fetch tasks**



Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

http://localhost:8081/api/tasks

GET http://localhost:8081/api/tasks

```
{
  "title": "My project",
  "description": "Project for task management application",
  "status": "COMPLETED",
  "dueDate": "2026-02-12",
  "assignedTo": [
    {
      "name": "Vasmitha",
      "email": "vasmithadevara@gmail.com"
    },
    {
      "name": "Madhu"
```

Body 200 OK 32 ms 667 B

```
[
    {
        "id": 1,
        "title": "My project",
        "description": "Project for task management application",
        "status": "COMPLETED",
        "dueDate": "2026-02-12",
        "assignedTo": [
            {
                "name": "Vasmitha",
                "email": "vasmithadevara@gmail.com"
            },
```

Create a new collection fo

Explore building and testir

Learn how to organize req

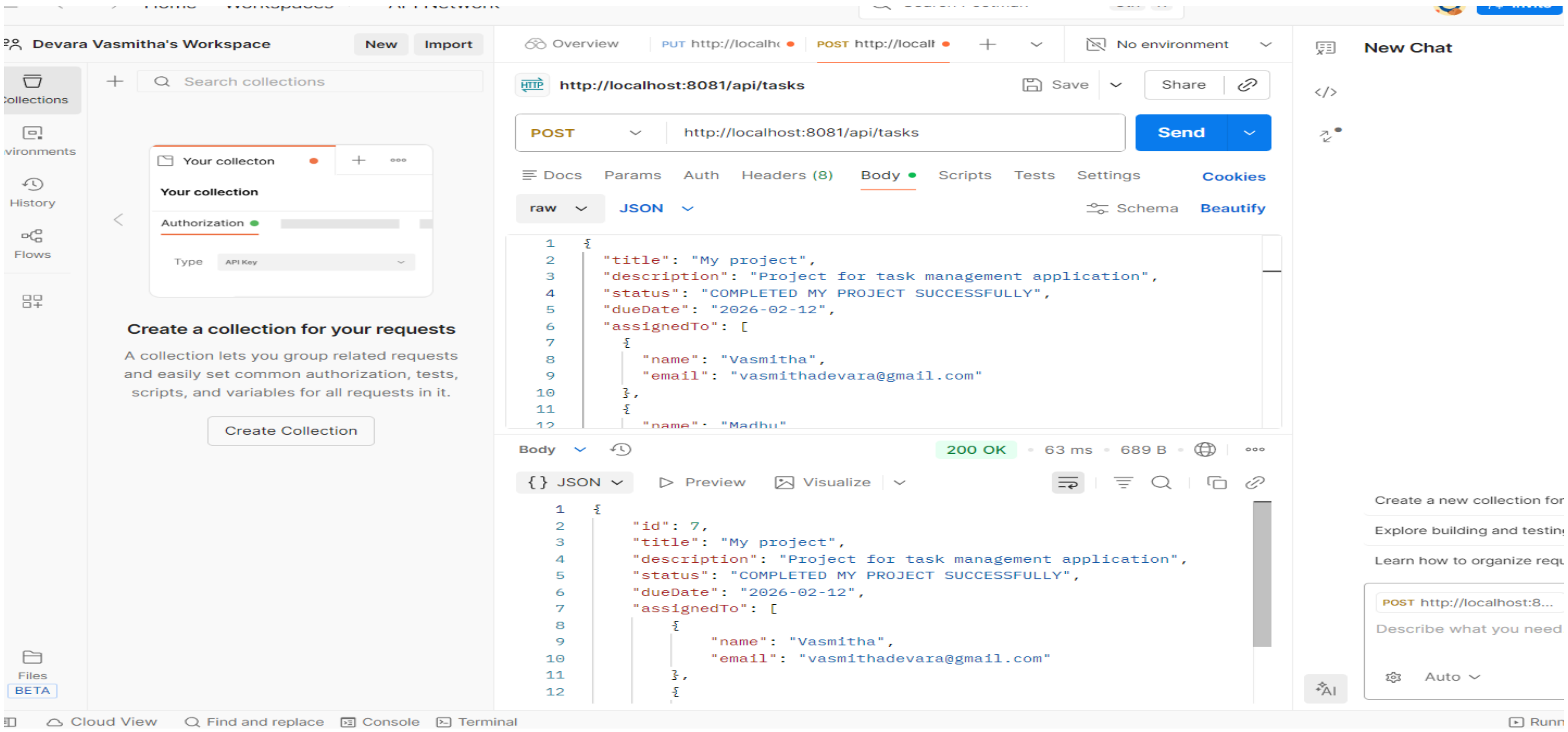
GET http://localhost:8...

Describe what you neec

RPS learn..evolve

REST API EndPoints

- **POST GET /api/task/user – Add task**



http://localhost:8081/api/tasks

POST http://localhost:8081/api/tasks

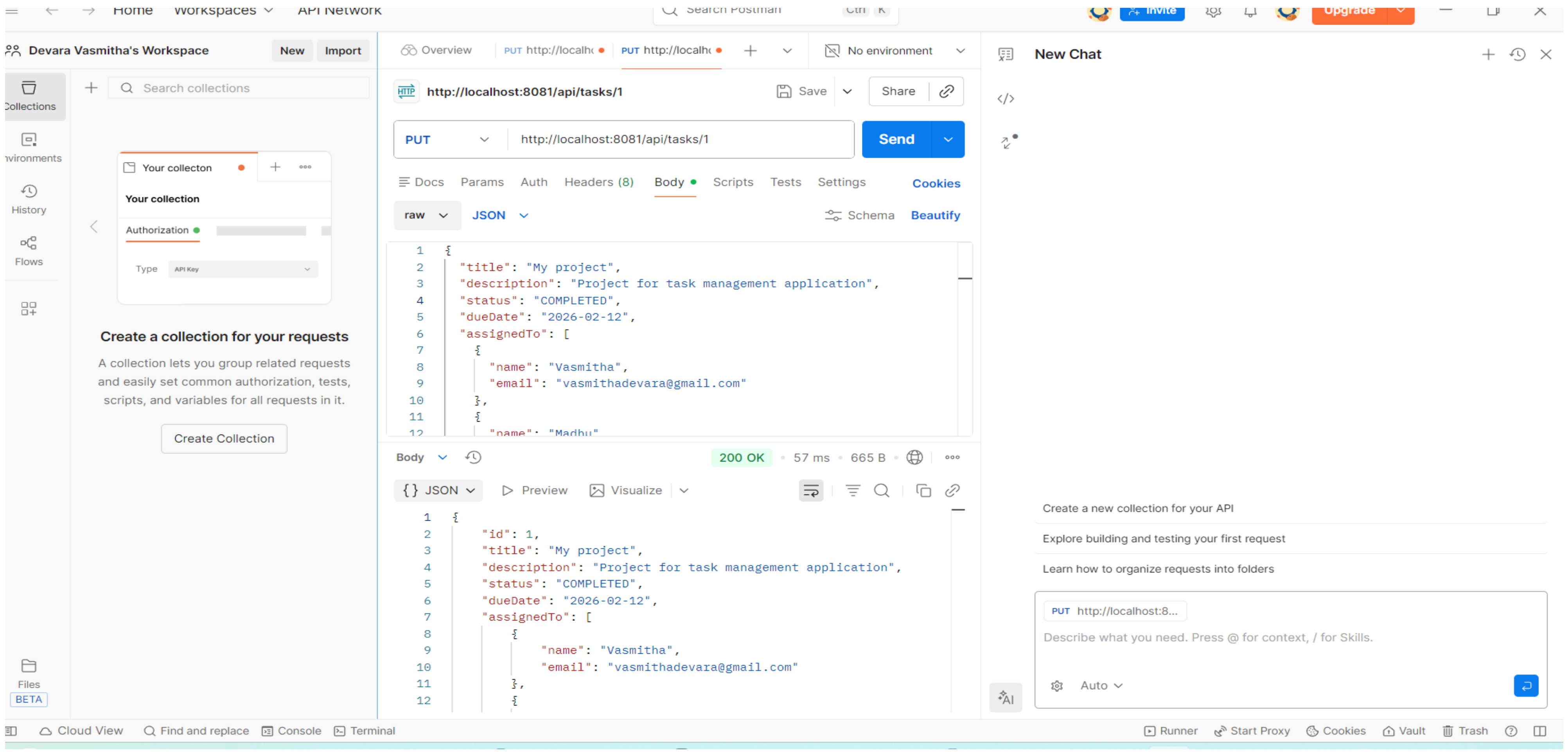
```
{
  "title": "My project",
  "description": "Project for task management application",
  "status": "COMPLETED MY PROJECT SUCCESSFULLY",
  "dueDate": "2026-02-12",
  "assignedTo": [
    {
      "name": "Vasmitha",
      "email": "vasmithadevara@gmail.com"
    },
    {
      "name": "Madhu"
```

Body

```
{
    "id": 7,
    "title": "My project",
    "description": "Project for task management application",
    "status": "COMPLETED MY PROJECT SUCCESSFULLY",
    "dueDate": "2026-02-12",
    "assignedTo": [
        {
            "name": "Vasmitha",
            "email": "vasmithadevara@gmail.com"
        },
        {
```

REST API EndPoints

- **PUT /api/task/user/{taskId} – Update task**


Devara Vasmitha's Workspace

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

http://localhost:8081/api/tasks/1

PUT http://localhost:8081/api/tasks/1

```
{
  "title": "My project",
  "description": "Project for task management application",
  "status": "COMPLETED",
  "dueDate": "2026-02-12",
  "assignedTo": [
    {
      "name": "Vasmitha",
      "email": "vasmithadevara@gmail.com"
    },
    {
      "name": "Madhu"
```

Body 200 OK 57 ms 665 B

```
{
    "id": 1,
    "title": "My project",
    "description": "Project for task management application",
    "status": "COMPLETED",
    "dueDate": "2026-02-12",
    "assignedTo": [
        {
            "name": "Vasmitha",
            "email": "vasmithadevara@gmail.com"
        },
        {
```

New Chat

Create a new collection for your API

Explore building and testing your first request

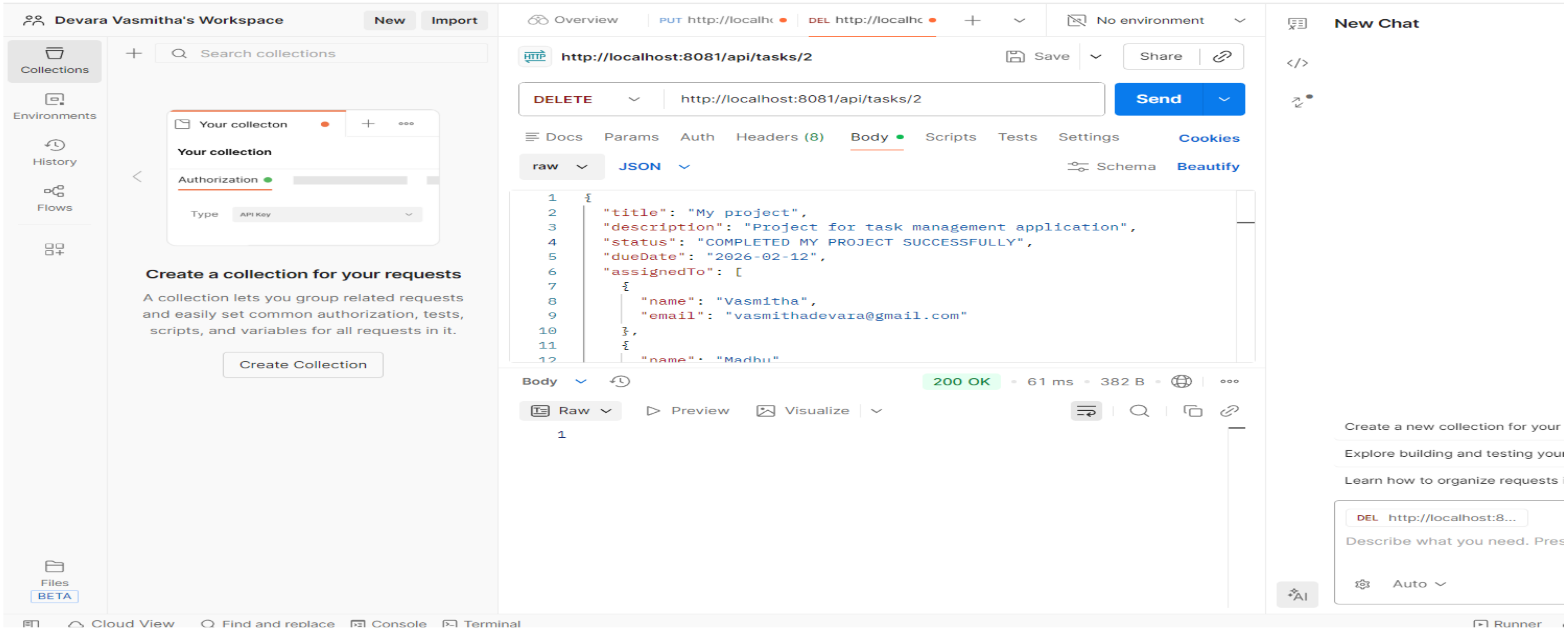
Learn how to organize requests into folders

PUT http://localhost:8...

Describe what you need. Press @ for context, / for Skills.

REST API End Points

- **DELETE /api/tasks/{taskId}– Delete task**



Devara Vasmitha's Workspace

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Your collection — Authorization — Type: API Key

PUT http://localhc | DEL http://localhc | No environment

http://localhost:8081/api/tasks/2

DELETE http://localhost:8081/api/tasks/2

```
{
  "title": "My project",
  "description": "Project for task management application",
  "status": "COMPLETED MY PROJECT SUCCESSFULLY",
  "dueDate": "2026-02-12",
  "assignedTo": [
    {
      "name": "Vasmitha",
      "email": "vasmithadevara@gmail.com"
    },
    {
      "name": "Madhu"
```

Response: 200 OK · 61 ms · 382 B

```
1
```

New Chat

Create a new collection for your

Explore building and testing your

Learn how to organize requests

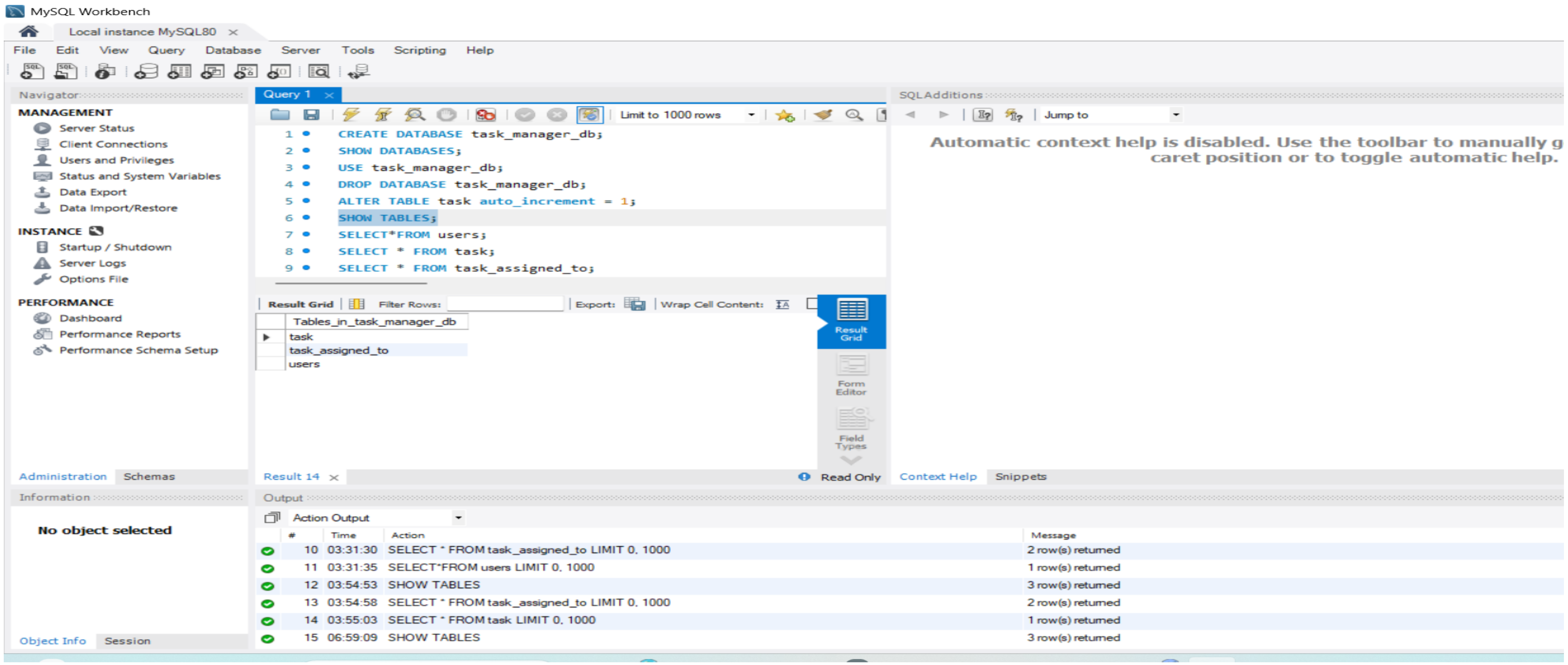
DEL http://localhost:8...

Describe what you need. Pres

learn..evolve

General - RPS Data

Backend Implementation



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT
- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE
- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE
- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1

Limit to 1000 rows

```
CREATE DATABASE task_manager_db;
SHOW DATABASES;
USE task_manager_db;
DROP DATABASE task_manager_db;
ALTER TABLE task auto_increment = 1;
SHOW TABLES;
SELECT*FROM users;
SELECT * FROM task;
SELECT * FROM task_assigned_to;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Tables_in_task_manager_db
task
task_assigned_to
users

Result 14 — Read Only

SQLAdditions

Jump to

Automatic context help is disabled. Use the toolbar to manually g
caret position or to toggle automatic help.

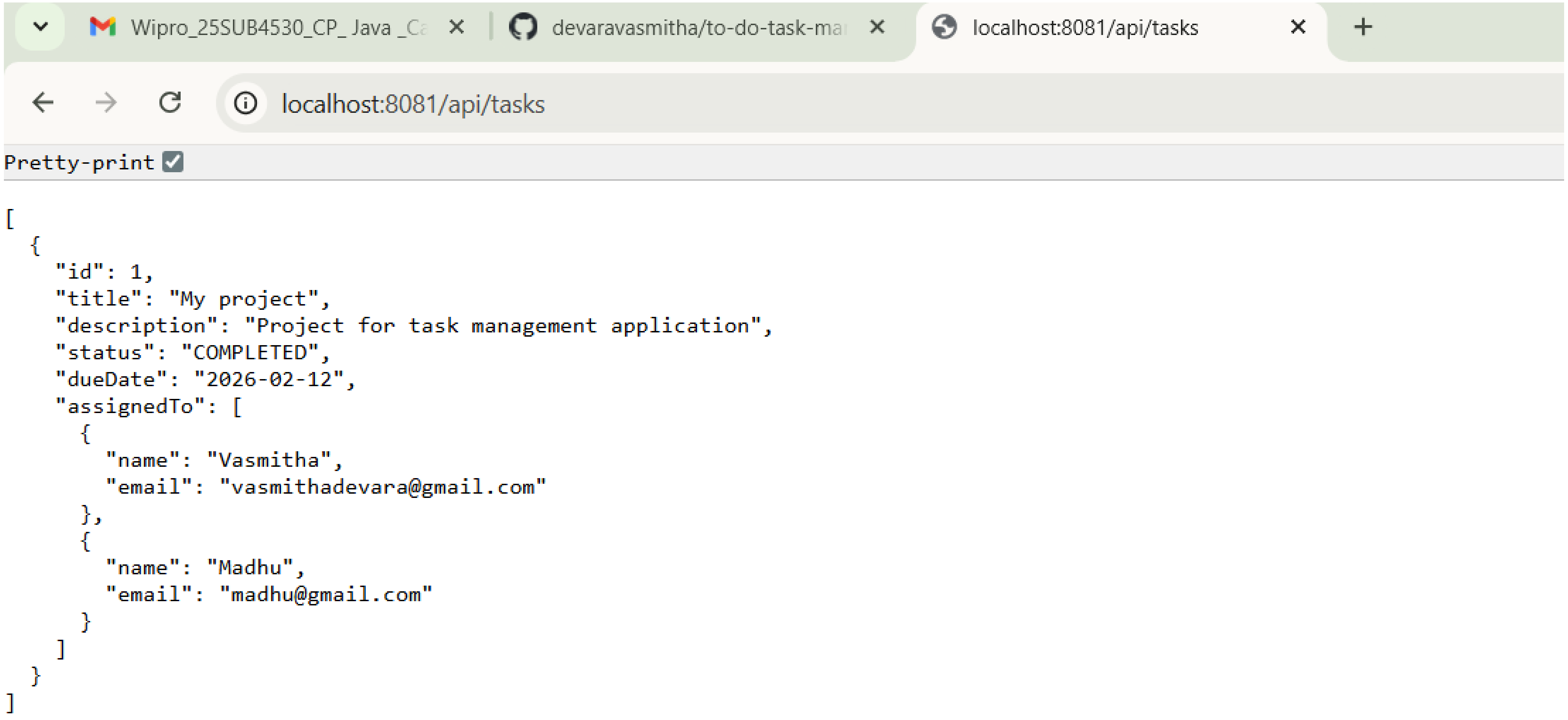
Context Help Snippets

Output

Action Output

```
 #   Time      Action                                           Message
10   03:31:30  SELECT * FROM task_assigned_to LIMIT 0, 1000     2 row(s) returned
11   03:31:35  SELECT*FROM users LIMIT 0, 1000                  1 row(s) returned
12   03:54:53  SHOW TABLES                                      3 row(s) returned
13   03:54:58  SELECT * FROM task_assigned_to LIMIT 0, 1000     2 row(s) returned
14   03:55:03  SELECT * FROM task LIMIT 0, 1000                 1 row(s) returned
15   06:59:09  SHOW TABLES                                      3 row(s) returned
```

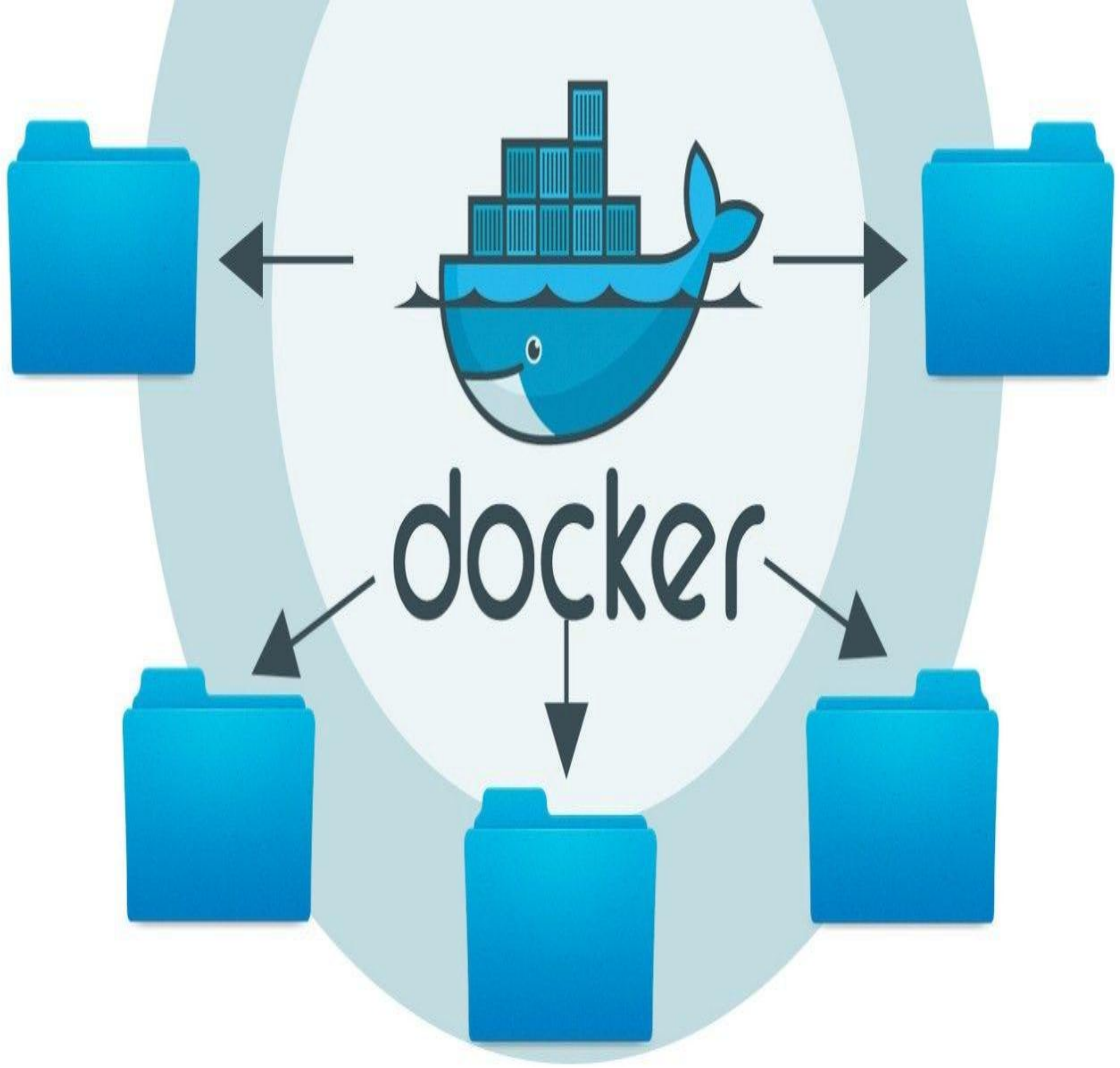
Output:



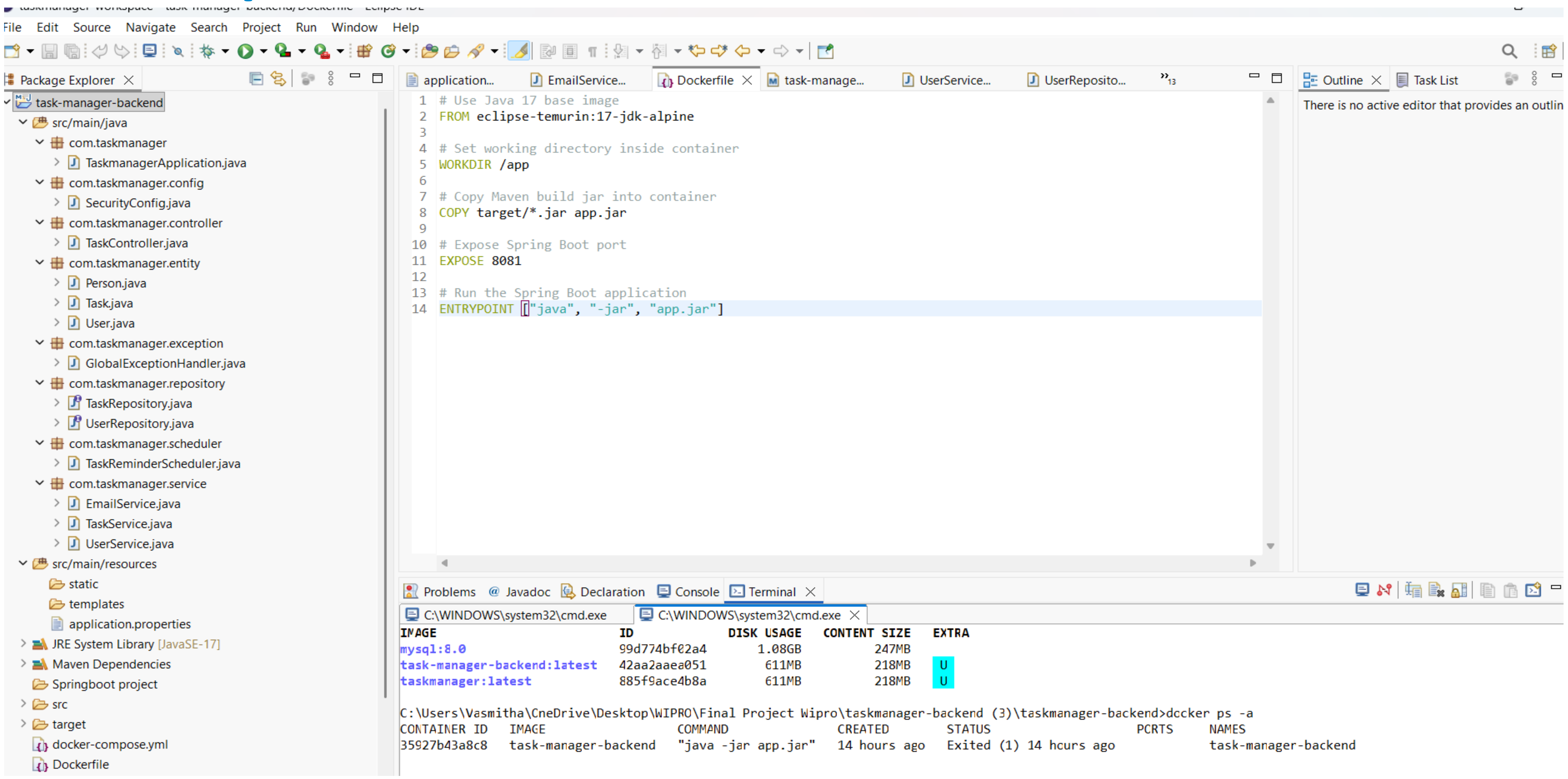
```
[
  {
    "id": 1,
    "title": "My project",
    "description": "Project for task management application",
    "status": "COMPLETED",
    "dueDate": "2026-02-12",
    "assignedTo": [
      {
        "name": "Vasmitha",
        "email": "vasmithadevara@gmail.com"
      },
      {
        "name": "Madhu",
        "email": "madhu@gmail.com"
      }
    ]
  }
]
```

Docker Implementation

- Created Dockerfile for backend
- Docker image built successfully
- Backend application packaged as JAR
- Docker container created from image
- Enables portability and consistency



Docker Implementation

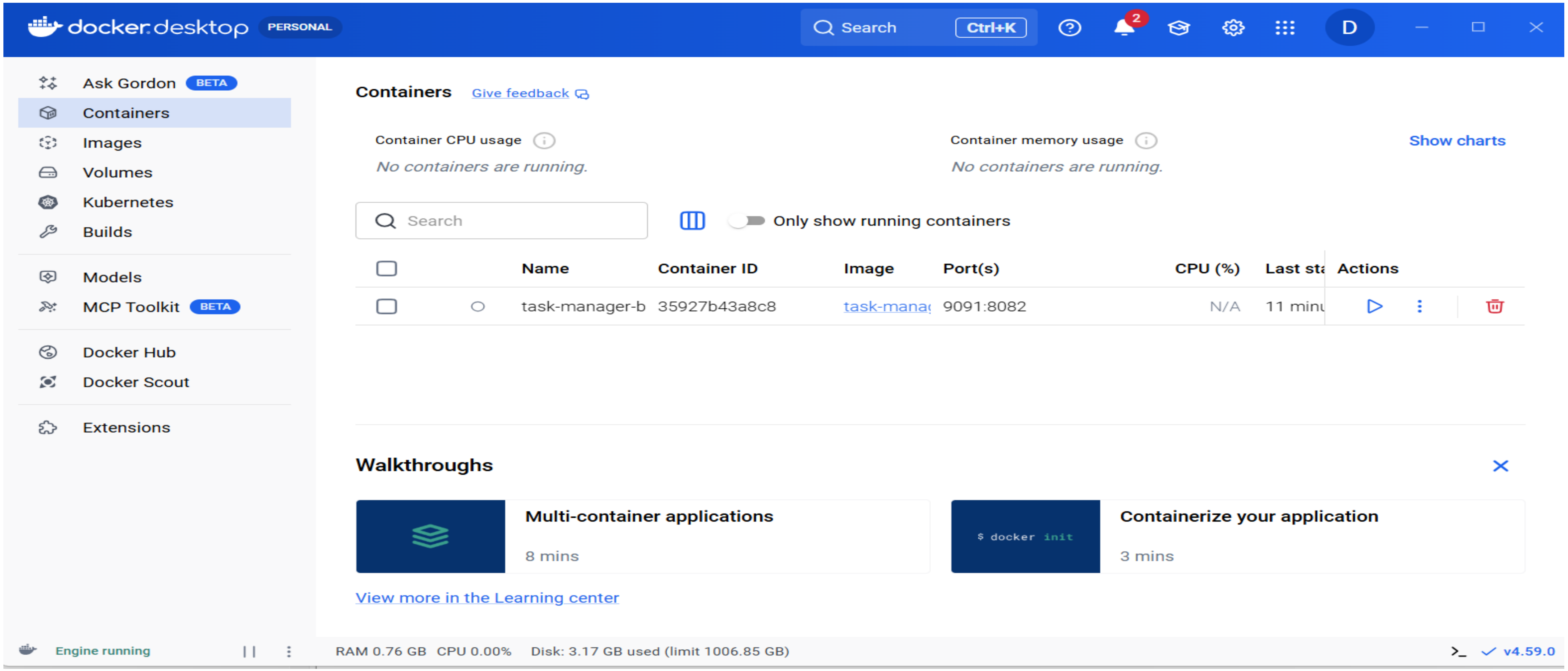


```
task-manager-backend
  src/main/java
    com.taskmanager
      TaskmanagerApplication.java
    com.taskmanager.config
      SecurityConfig.java
    com.taskmanager.controller
      TaskController.java
    com.taskmanager.entity
      Person.java
      Task.java
      User.java
    com.taskmanager.exception
      GlobalExceptionHandler.java
    com.taskmanager.repository
      TaskRepository.java
      UserRepository.java
    com.taskmanager.scheduler
      TaskReminderScheduler.java
    com.taskmanager.service
      EmailService.java
      TaskService.java
      UserService.java
  src/main/resources
    static
    templates
    application.properties
  JRE System Library [JavaSE-17]
  Maven Dependencies
  Springboot project
  src
  target
  docker-compose.yml
  Dockerfile
```

```dockerfile
# Use Java 17 base image
FROM eclipse-temurin:17-jdk-alpine

# Set working directory inside container
WORKDIR /app

# Copy Maven build jar into container
COPY target/*.jar app.jar

# Expose Spring Boot port
EXPOSE 8081

# Run the Spring Boot application
ENTRYPOINT ["java", "-jar", "app.jar"]
```

```
IMAGE                        ID             DISK USAGE   CONTENT SIZE   EXTRA
mysql:8.0                    99d774bf02a4       1.08GB          247MB
task-manager-backend:latest  42aa2aaea051        611MB          218MB    U
taskmanager:latest           885f9ace4b8a        611MB          218MB    U

C:\Users\Vasmitha\OneDrive\Desktop\WIPRO\Final Project Wipro\taskmanager-backend (3)\taskmanager-backend>docker ps -a
CONTAINER ID   IMAGE                  COMMAND                CREATED        STATUS                     PORTS     NAMES
35927b43a8c8   task-manager-backend   "java -jar app.jar"    14 hours ago   Exited (1) 14 hours ago              task-manager-backend
```

Docker Implementation



Containers Give feedback

Container CPU usage
No containers are running.

Container memory usage
No containers are running.

Show charts

Search

Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last sta	Actions
task-manager-b	35927b43a8c8	task-mana	9091:8082	N/A	11 minu	

Walkthroughs

Multi-container applications
8 mins

```
$ docker init
```

Containerize your application
3 mins

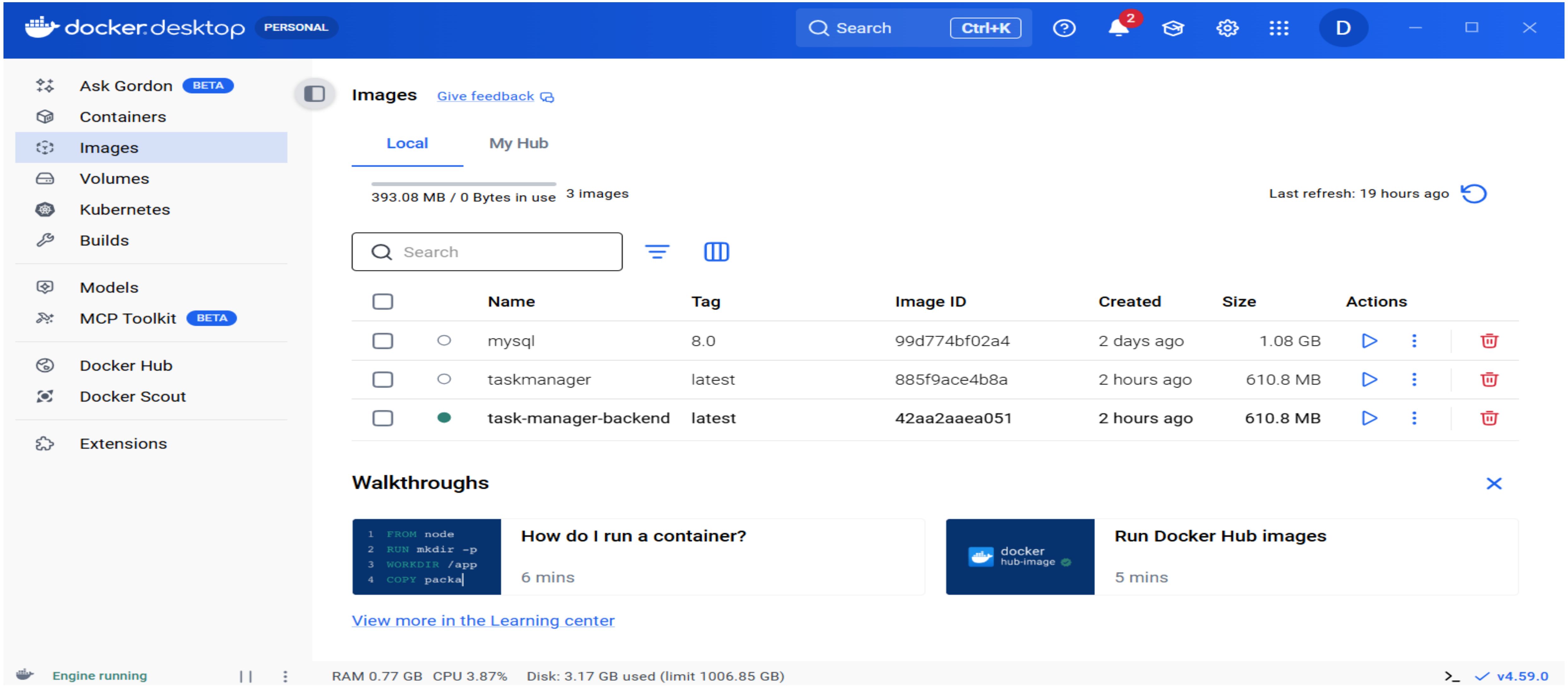
View more in the Learning center

Engine running

RAM 0.76 GB CPU 0.00% Disk: 3.17 GB used (limit 1006.85 GB)

v4.59.0

Docker Implementation



393.08 MB / 0 Bytes in use 3 images

Last refresh: 19 hours ago

		Name	Tag	Image ID	Created	Size	Actions
☐	○	mysql	8.0	99d774bf02a4	2 days ago	1.08 GB	
☐	○	taskmanager	latest	885f9ace4b8a	2 hours ago	610.8 MB	
☐	●	task-manager-backend	latest	42aa2aaea051	2 hours ago	610.8 MB	

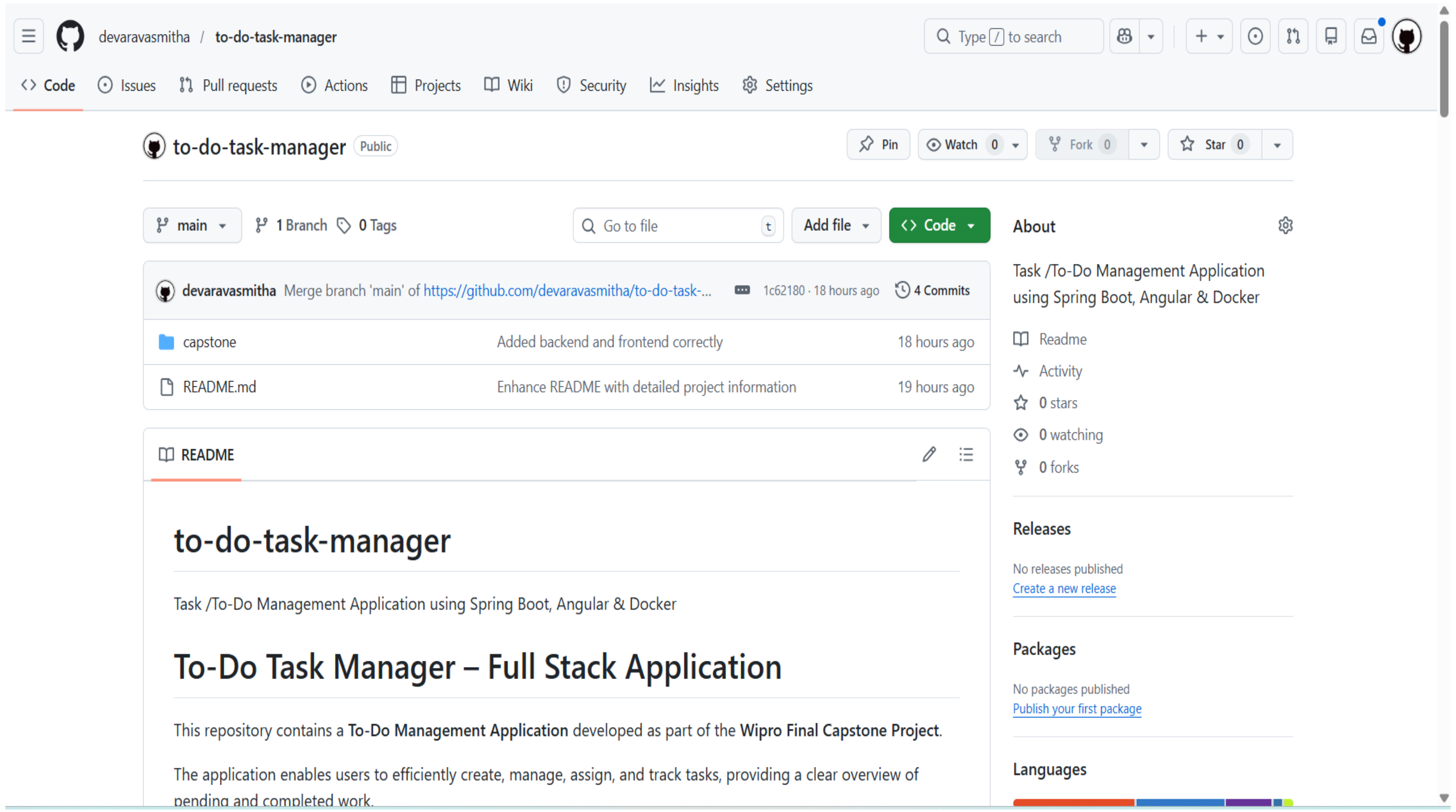
```
FROM node
RUN mkdir -p
WORKDIR /app
COPY packa
```

RAM 0.77 GB CPU 3.87% Disk: 3.17 GB used (limit 1006.85 GB) v4.59.0

Azure Deployement

- **Docker image pushed to Azure**
- **Azure Container Instance created**
- **Backend deployed on Azure**
- **Application accessible via public endpoint**

Source Code Management



devaravasmitha Merge branch 'main' of https://github.com/devaravasmitha/to-do-task-...	1c62180 · 18 hours ago	4 Commits
capstone	Added backend and frontend correctly	18 hours ago
README.md	Enhance README with detailed project information	19 hours ago

README

to-do-task-manager

Task /To-Do Management Application using Spring Boot, Angular & Docker

To-Do Task Manager – Full Stack Application

This repository contains a **To-Do Management Application** developed as part of the **Wipro Final Capstone Project**.

The application enables users to efficiently create, manage, assign, and track tasks, providing a clear overview of pending and completed work.

About

Task /To-Do Management Application using Spring Boot, Angular & Docker

Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- GitHub used for version control
- Proper project structure maintained
- Meaningful commit messages
- README file included with setup instructions

Learning Outcomes

- Full-stack application development
- REST API design using Spring Boot
- Angular frontend development
- Docker containerization
- Azure cloud deployment
- Debugging and problem-solving

Conclusion

The Task Management Application provides an efficient and user-friendly solution for organizing daily tasks by allowing users to create, update, delete, and track tasks based on their completion status and due dates. By integrating an Angular-based frontend with a Spring Boot backend, the system ensures smooth interaction, secure data handling, and reliable persistence using Spring Data JPA. The application improves productivity by offering clear visibility of pending and completed tasks through filtering and status updates. Overall, this project demonstrates a practical full-stack implementation that follows real-world development practices and serves as a strong foundation for future enhancements such as notifications, advanced analytics, and cloud scalability.

THANK YOU