Final capstone project

Capstone Group Project

Assignment Title: **Full-Stack Web Application Development & Deployment**

1. Objective	***The objective of this capstone project is for student teams to collaboratively plan, design, implement, and deploy a functional full-stack web application. This project will synthesize the skills and knowledge acquired throughout the course, emphasizing real-world development practices, teamwork, and professional presentation.***
2. Required Technical Elements & Functionality	✓ ***User Authentication*** ✓ ***Database Integration*** ✓ ***External API Integration*** ✓ ***Web Application Structure*** ✓ ***Deployment & DevOps*** • **Dockerization** • **Cloud Deployment(AWS EC2 Instance)** • **DockerHub**

Final capstone project

Capstone Group Project

Assignment Title: **Full-Stack Web Application Development & Deployment**

3. Deliverables

1. GitHub Repository

https://github.com/chongwongus/TaskFlow-student-project-management

2. Live Deployed Application

http://13.220.160.13:3000/

3. DockerHub Images

https://hub.docker.com/repository/docker/janninegdm888/taskflow-client/general

https://hub.docker.com/repository/docker/janninegdm888/taskflow-server/general

4. Presentation / Video Submission

University of Washington Tacoma
School of Engineering and Technology
24-25 Graduate Certificate in Software Development Engineering program (GC-SDE)

TCSS 506 A Sp 25:

Practical Full Stack Development

Preston Harms

AWS Project Manager

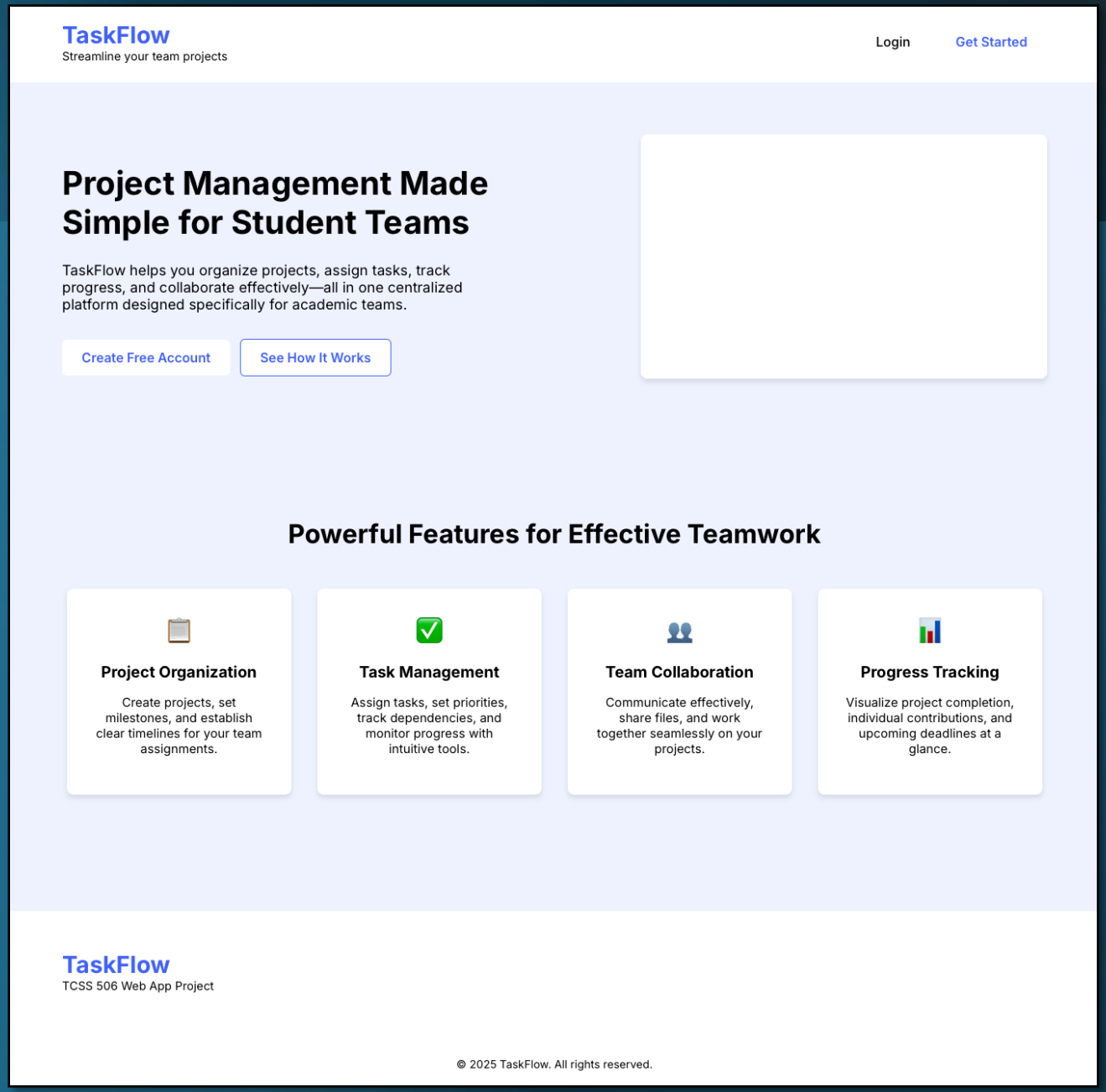
Richard Le

Git and GitHub Version Control Project Manager

Jannine G. D. MacGormain

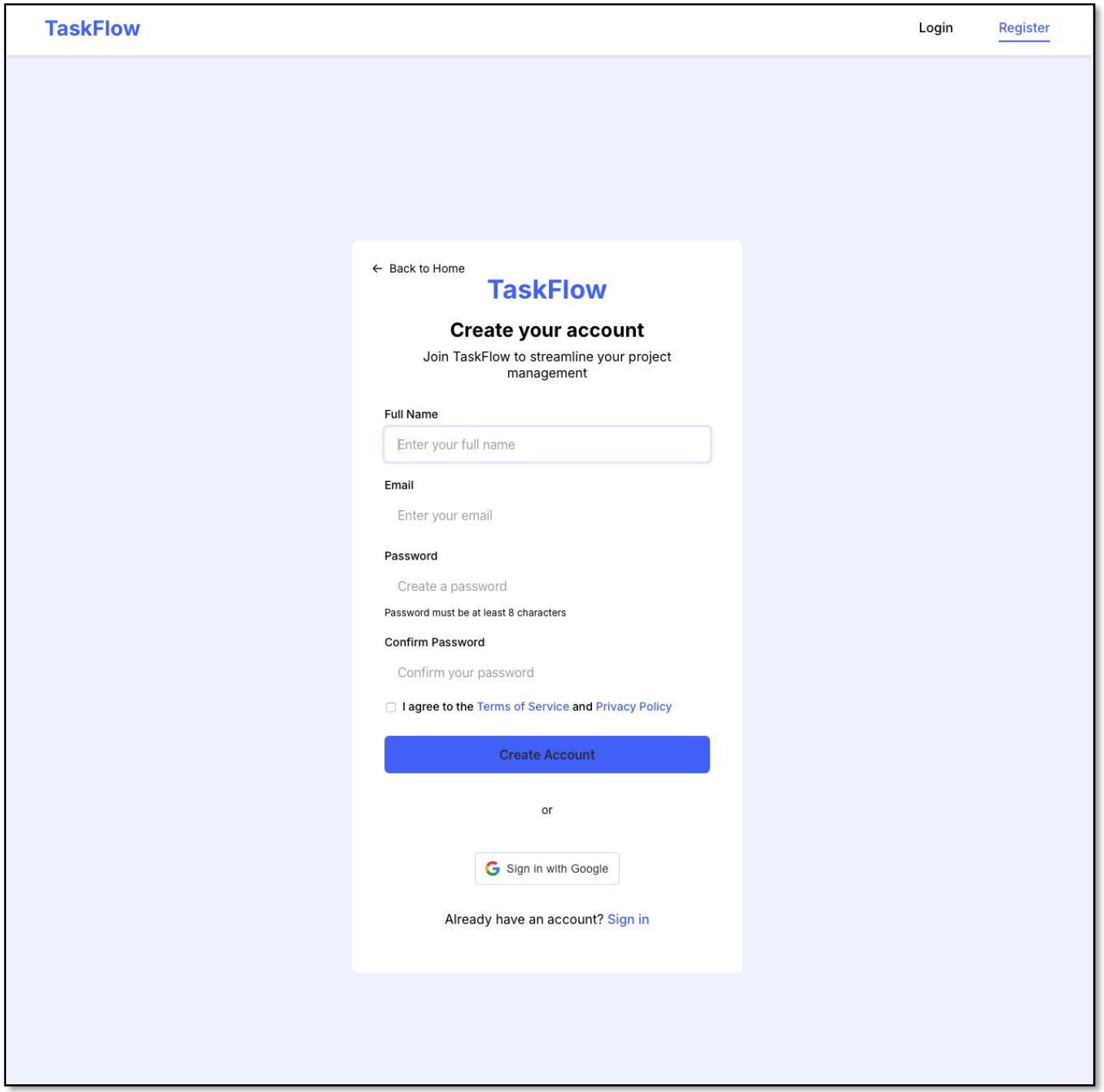
YouTrack Project Manager

TaskFlow: A Project Management Platform



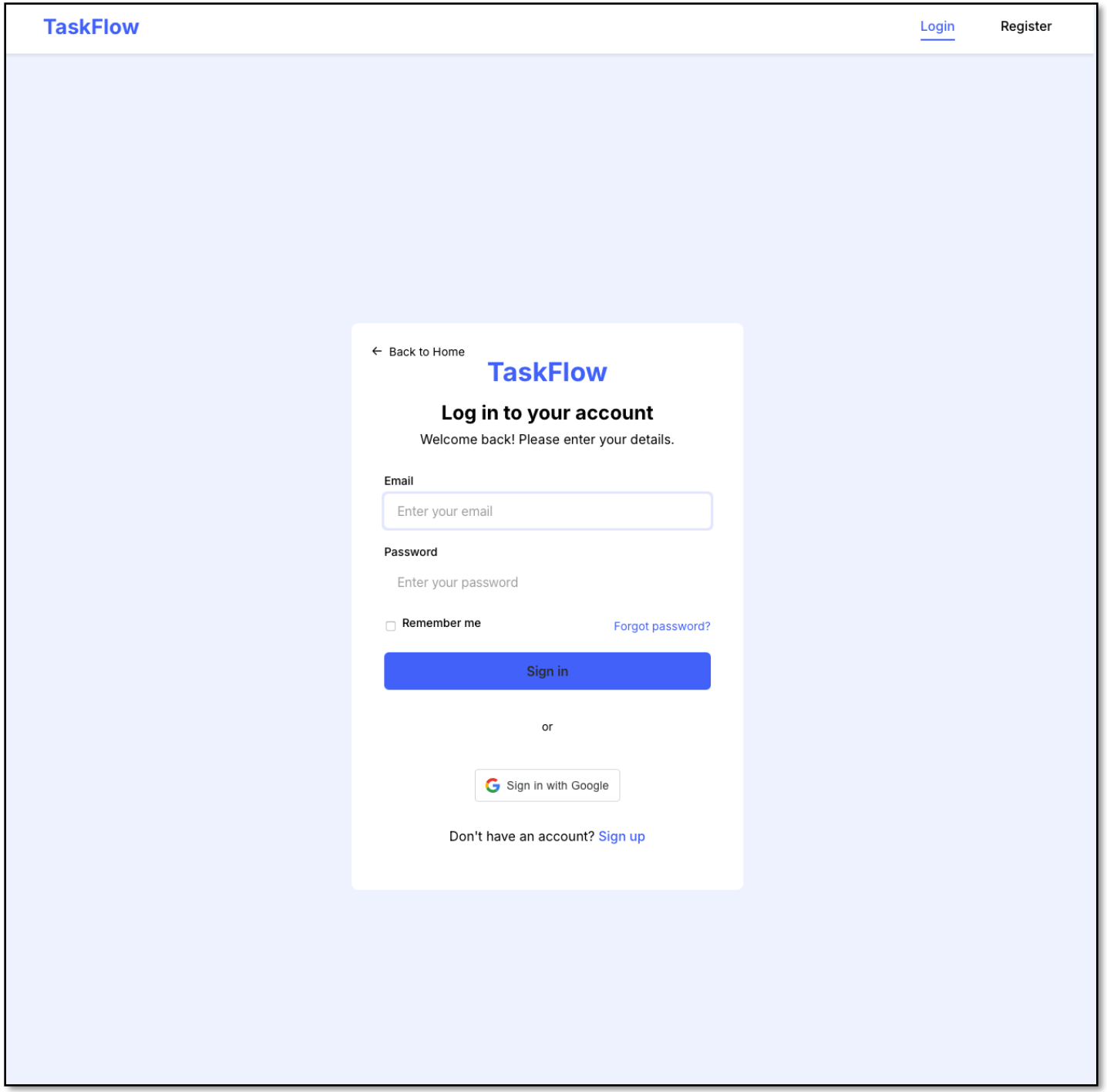
TaskFlow

Register Page



TaskFlow

Login Page

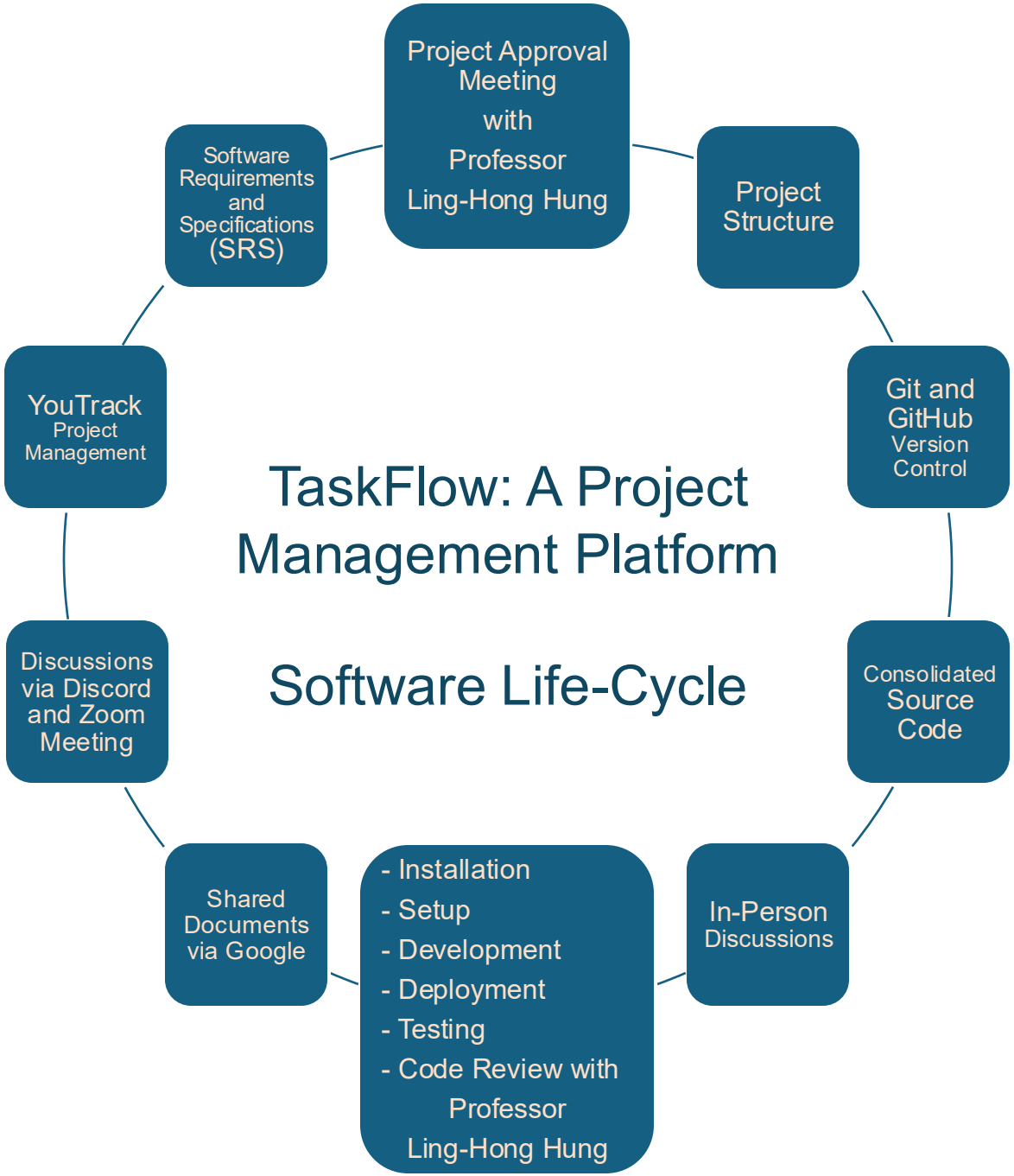


The following slides are for illustration purposes only.

The **TaskFlow p**roject will be demonstrated live shortly.

TaskFlow: A Project Management Platform

Software Life-Cycle



Course Project Overview

TaskFlow is a standalone web application, developed as part of the **TCSS 506** Web Development Course Group Project.

It aims to be a comprehensive project management platform designed specifically for student teams to coordinate group projects, track assignments and enhance collaboration. The application is intended for project group members who are responsible for implementing and verifying the functionality of the web application.

The project emphasizes collaborative planning, design, and deployment of the web application while applying design methodologies to improve the project's modularity.

TaskFlow will utilize MongoDB for database management, React with Typescript for frontend development and Node.js for backend services. The application will be deployed on Amazon Web Services (AWS) and will incorporate external APIs for data integration. Additionally, **TaskFlow** will be containerized using Docker to facilitate efficient deployment.

Software Requirements

- The software will run on macOS, Windows, and Linux operating systems with a minimum requirement of 4 GB RAM.
- Node.js 14.x or higher (preferably the latest stable version) is required.
- npm 6.x or higher (preferably the latest stable version) is required.
- Python 3.x or higher (preferably the latest stable version) is required.
- MongoDB for data management (preferably the latest stable version) is required.

Hardware Requirements

- Monitor (laptop or desktop)
- Keyboard
- Mouse

Pending Decisions

- The choice of additional third-party libraries for enhanced graphics and other functionalities.

- Outstanding decisions regarding the implementation of multiple page content are still pending, which will impact future development.

Special Thanks to:

Professor Ling-Hong Hung for the valuable learnings, guidance, dedication, and wisdom shared during the TCSS 506 course.

References:

- Course Project Guidelines and Outline prepared by Professor Ling-Hong Hung.
- TCSS 506 Modules prepared by Professor Ling-Hong Hung.
- Team discussions via Discord and collaborative task progress tracking through the GitHub repository and YouTrack project management.
- TCSS 505 Modules prepared by Professor Robert Cordingly.
- [SRS Templates] TCSS 504 Software Requirements and Specifications (SRS) Assignment prepared by Professor Tom Capaul.
- TCSS 504 Modules prepared by Professor Tom Capaul.
- Student Mentoring with Professor Charlie LeWarne.
- Ubuntu Linux Manual Pages
- AWS Documentation: ***https://docs.aws.amazon.com***
- Dockerdocs: ***https://docs.docker.com/manuals/***
- MongoDB Documentation: ***https://www.mongodb.com/docs/***
- API Library: ***https://console.cloud.google.com/apis/library?pli=1&inv=1&invt=AbyN3Q***

TaskFlow: A Project Management Platform

Preston Harms	Richard Le	Jannine G. D. MacGormain
AWS Project Manager	Git and GitHub Version Control Project Manager	YouTrack Project Manager
Main Job	**Main Job**	**Main Job**
Amazon Web Services(AWS) Manager	Git and GitHub Version Control Manager	YouTrack Course Project Management Manager
Other Jobs	**Other Jobs**	**Other Jobs**
▪ Implemented the Foundational Source Code	▪ Implemented the Foundational Source Code	▪ Documentation Writer ▪ Source Code Reviewer
Key Contributions	**Key Contributions**	**Key Contributions**
▪ SRS – Proofreader and Editor. ▪ Conducted Project Development and Deployment ▪ Proofreader and Editor Project Presentation Slides.	▪ SRS – Proofreader and Editor. ▪ Conducted Project Development and Deployment ▪ Proofreader and Editor Project Presentation Slides.	▪ Created Software Requirements and Specifications (SRS): Initial and Final Draft. ▪ Project Development and Deployment Reviewer ▪ Created Project Presentation Slides.

Project Structure

```
project-root/
│
├── frontend/           # React TypeScript frontend
│   ├── public/         # Static files
│   └── src/            # Source files
│       ├── components/ # React components
│       ├── pages/      # Page components
│       ├── services/   # API services
│       └── context/    # React context (auth, etc.)
│
├── backend/            # Node.js Express backend
│   ├── controllers/    # Request handlers
│   ├── models/         # Mongoose models
│   ├── routes/         # API routes
│   ├── middleware/     # Express middleware
│   ├── services/       # External service integrations
│   └── config/         # Configuration files
│
├── docker/             # Docker configuration files
│   ├── Dockerfile
│   └── docker-compose.yml
│
└── docs/               # Documentation
    └── planning.md     # Project planning document
```

Installation and Setup

Google Developer account

Node.js V14+

MongoDB

GitHub account

npm v6+

Docker

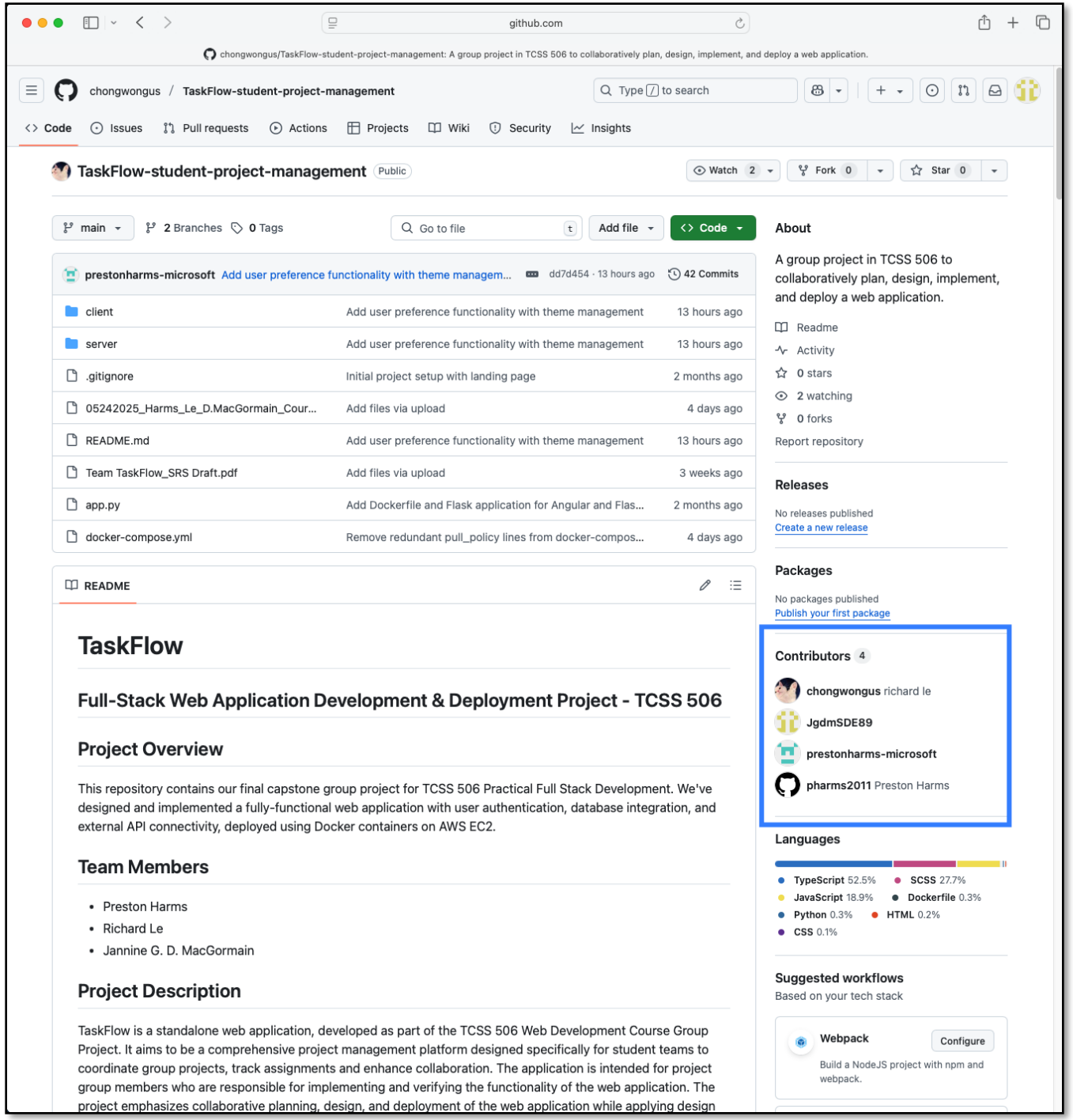
Required Accounts for the **TaskFlow** Project

GitHub	MongoDB Atlas	Google Developer	Docker	Amazon Web Services (AWS)
*For **version control** and collaboration on the codebase*	*For cloud-based **database management** and access*	*To set up Google **APIs** for authentication and data integration*	*For managing **Docker images** and **containers** for deployment.*	*For deploying the **application** on Amazon Web Services*

TaskFlow Technology Stack

Frontend	***React*, *TypeScript*, *SCSS*, *React Router***
Backend	***Node.js*, *Express*, *Mongoose***
Database	***MongoDB Atlas***
Authentication	***JWT*, *Google Auth***
External APIs	***GitHub API***
Deployment	***Docker*, *AWS EC2***
Version Control	***Git*, *GitHub***
Sprints (via Task Cards)	***YouTrack (by JetBrains)***

GitHub


github.com
chongwongus/TaskFlow-student-project-management: A group project in TCSS 506 to collaboratively plan, design, implement, and deploy a web application.
chongwongus / TaskFlow-student-project-management
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
TaskFlow-student-project-management
Public
Watch 2
Fork 0
Star 0
main
2 Branches
0 Tags
Go to file
t
Add file
Code
prestonharms-microsoft Add user preference functionality with theme managem...
dd7d454 · 13 hours ago
42 Commits
client
Add user preference functionality with theme management
13 hours ago
server
Add user preference functionality with theme management
13 hours ago
.gitignore
Initial project setup with landing page
2 months ago
05242025_Harms_Le_D.MacGormain_Cour...
Add files via upload
4 days ago
README.md
Add user preference functionality with theme management
13 hours ago
Team TaskFlow_SRS Draft.pdf
Add files via upload
3 weeks ago
app.py
Add Dockerfile and Flask application for Angular and Flas...
2 months ago
docker-compose.yml
Remove redundant pull_policy lines from docker-compos...
4 days ago
README
TaskFlow
Full-Stack Web Application Development & Deployment Project - TCSS 506
Project Overview
This repository contains our final capstone group project for TCSS 506 Practical Full Stack Development. We've designed and implemented a fully-functional web application with user authentication, database integration, and external API connectivity, deployed using Docker containers on AWS EC2.
Team Members
Preston Harms
Richard Le
Jannine G. D. MacGormain
Project Description
TaskFlow is a standalone web application, developed as part of the TCSS 506 Web Development Course Group Project. It aims to be a comprehensive project management platform designed specifically for student teams to coordinate group projects, track assignments and enhance collaboration. The application is intended for project group members who are responsible for implementing and verifying the functionality of the web application. The project emphasizes collaborative planning, design, and deployment of the web application while applying design
About
A group project in TCSS 506 to collaboratively plan, design, implement, and deploy a web application.
Readme
Activity
0 stars
2 watching
0 forks
Report repository
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package
Contributors 4
chongwongus richard le
JgdmSDE89
prestonharms-microsoft
pharms2011 Preston Harms
Languages
TypeScript 52.5%
SCSS 27.7%
JavaScript 18.9%
Dockerfile 0.3%
Python 0.3%
HTML 0.2%
CSS 0.1%
Suggested workflows
Based on your tech stack
Webpack
Configure
Build a NodeJS project with npm and webpack.

GitHub

OAuth Setup

TaskFlow integrates with GitHub to connect repositories and sync issues with tasks. To enable GitHub integration:

1. Create a GitHub OAuth App

1. Go to GitHub Developer Settings
2. Click **"OAuth Apps"** in the left sidebar
3. Click **"New OAuth App"**
4. Fill in the application details:
   - **Application name:** `TaskFlow` (or your preferred name)
   - **Homepage URL:** `http://localhost:3000` (for local development)
   - **Application description:** `Project management platform for student teams`
   - **Authorization callback URL:** `http://localhost:3000/github/callback`
5. Click **"Register application"**
6. Copy the **Client ID** and **Client Secret** (you'll need these for your environment variables)

2. Configure Environment Variables

Add the GitHub OAuth credentials to your backend `.env` file:

```
# GitHub OAuth Configuration
GITHUB_CLIENT_ID=your_github_client_id_here
GITHUB_CLIENT_SECRET=your_github_client_secret_here

# Frontend URL for OAuth redirects
CLIENT_URL=http://localhost:3000
```

3. Production Setup

For production deployment, update your GitHub OAuth App:

1. Edit your GitHub OAuth App
2. Update the **Homepage URL** to your production domain
3. Update the **Authorization callback URL** to: `https://yourdomain.com/github/callback`
4. Update your production `.env` with:

```
CLIENT_URL=https://yourdomain.com
```

GitHub Integration

Once you have GitHub OAuth configured:

1. **Connect Your GitHub Account**:
   - Navigate to any project you own
   - Click "Connect GitHub Repository"
   - You'll be redirected to GitHub for authorization
   - After authorizing, you'll be redirected back to TaskFlow
2. **Link a Repository**:
   - Select a repository from your GitHub account
   - Click "Connect" to link it to your project
   - The repository will appear in your project details
3. **Future Features** (coming soon):
   - Sync GitHub issues with TaskFlow tasks
   - View commit history and repository stats
   - Create GitHub issues from TaskFlow tasks

Local Development

```
(venv) parallels@ubuntu-linux-2404:~/Desktop$ git clone https://github.com/chongwongus/TaskFlow-student-project-management.git
Cloning into 'TaskFlow-student-project-management'...
remote: Enumerating objects: 215, done.
remote: Counting objects: 100% (215/215), done.
remote: Compressing objects: 100% (147/147), done.
remote: Total 215 (delta 79), reused 184 (delta 55), pack-reused 0 (from 0)
Receiving objects: 100% (215/215), 242.84 KiB | 1.32 MiB/s, done.
Resolving deltas: 100% (79/79), done.
```

```
(venv) parallels@ubuntu-linux-2404:~/Desktop$ cd TaskFlow-student-project-management
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ ls
app.py  client  Dockerfile  README.md  server
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ cd client
```

Frontend

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ sudo apt-get install npm
[sudo] password for parallels:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-headers-6.8.0-58 linux-headers-6.8.0-58-generic linux-image-6.8.0-58-generic linux-modules-6.8.0-58-generic
  linux-modules-extra-6.8.0-58-generic linux-tools-6.8.0-58 linux-tools-6.8.0-58-generic
```

```
Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
 systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ npm install
npm WARN deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprecated. See the compatibility table on MDN: https:/
/developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort#browser_compatibility
npm WARN deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
```

Frontend

```
npm WARN deprecated @babel/plugin-proposal-private-property-in-object@7.21.11: This proposal has been merged to the ECMAScript standard and thus this plugin is no longe
r maintained. Please use @babel/plugin-transform-private-property-in-object instead.

added 1369 packages, and audited 1370 packages in 7s

277 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ ls
node_modules  package.json  package-lock.json  public  src  tsconfig.json
```

Backend

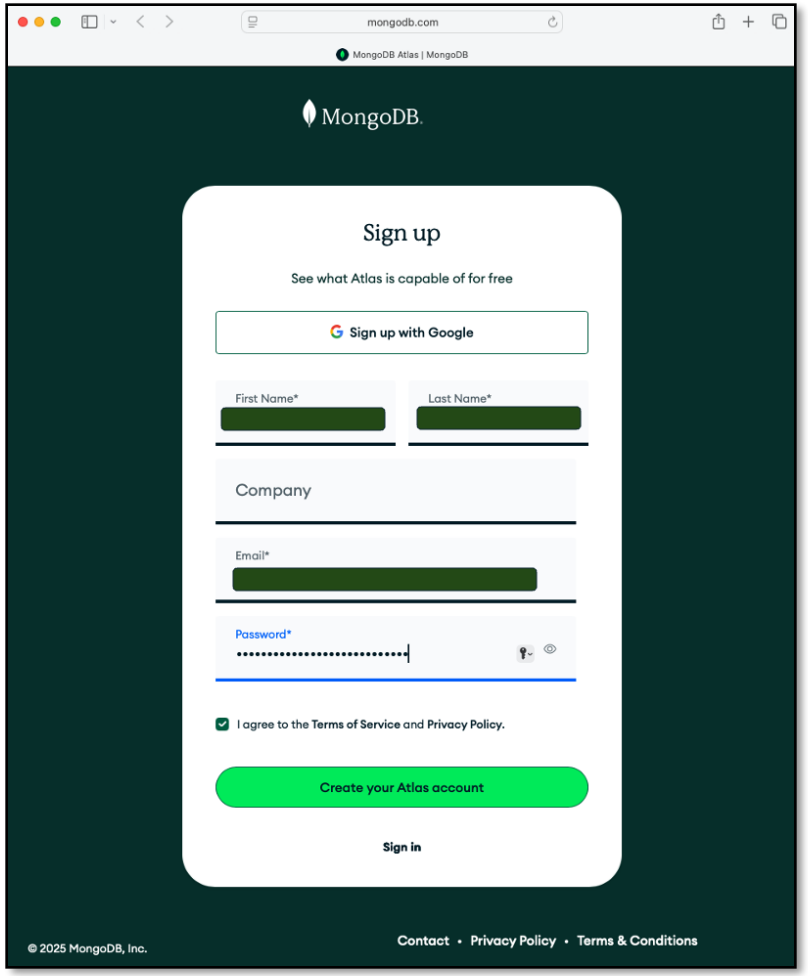
```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ cd ../server
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/server$ npm install

added 179 packages, and audited 180 packages in 710ms

25 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

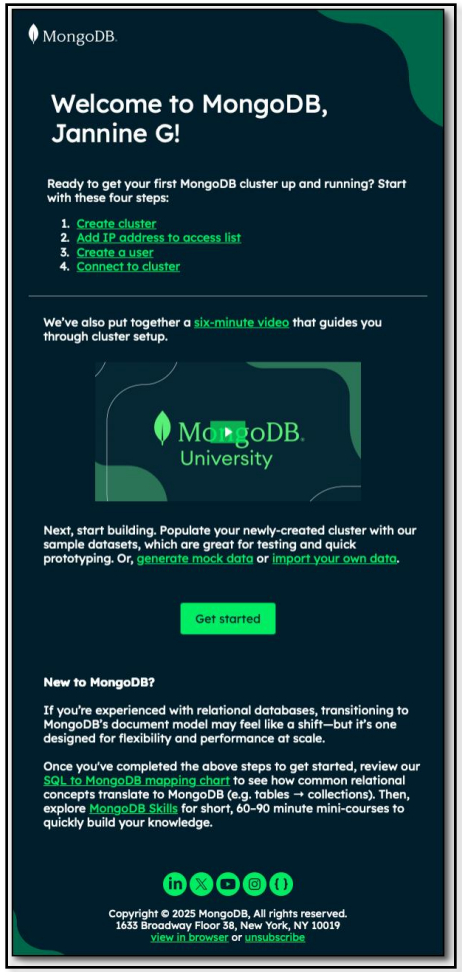
Environment Setup

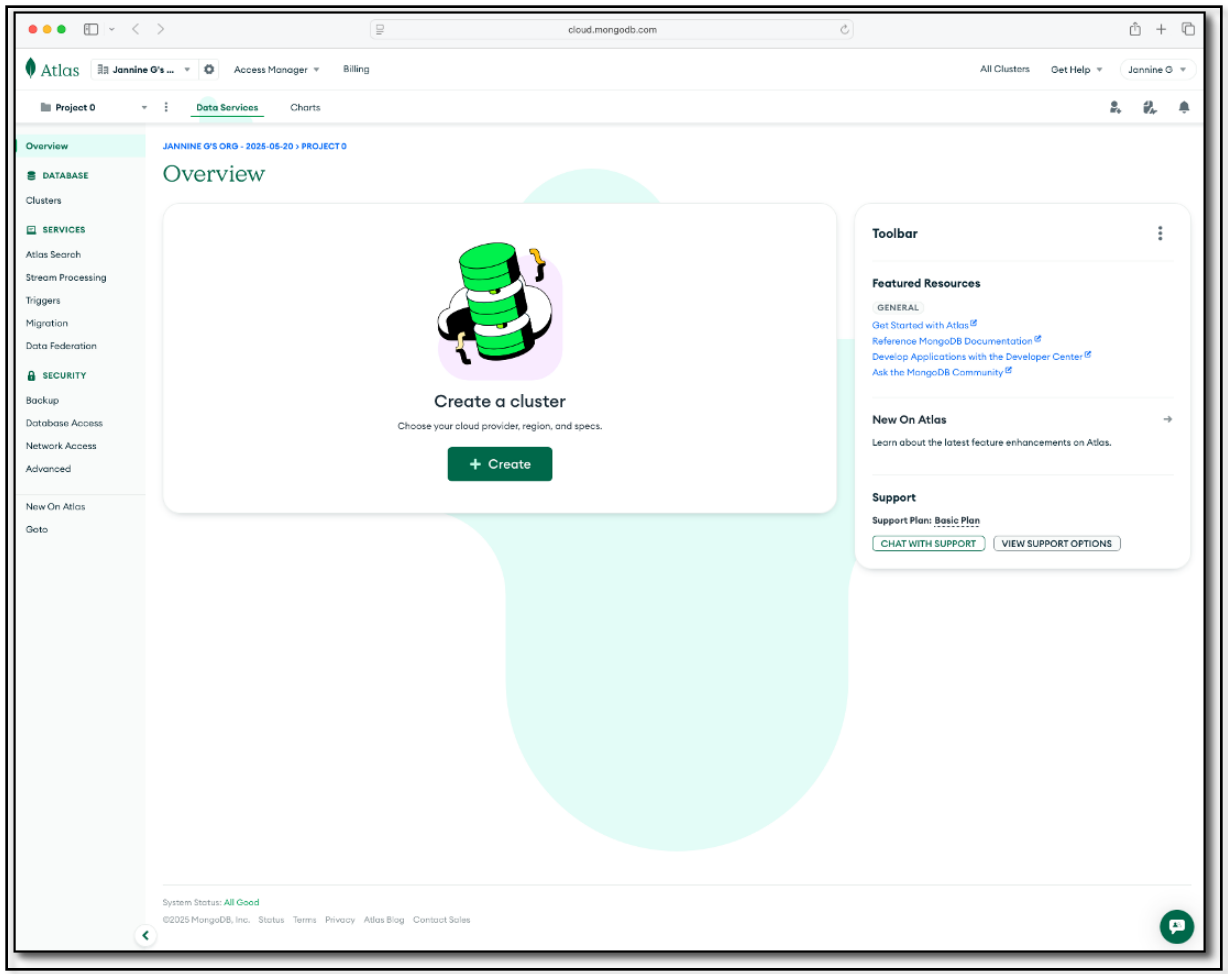
MongoDB Atlas

Google OAuth

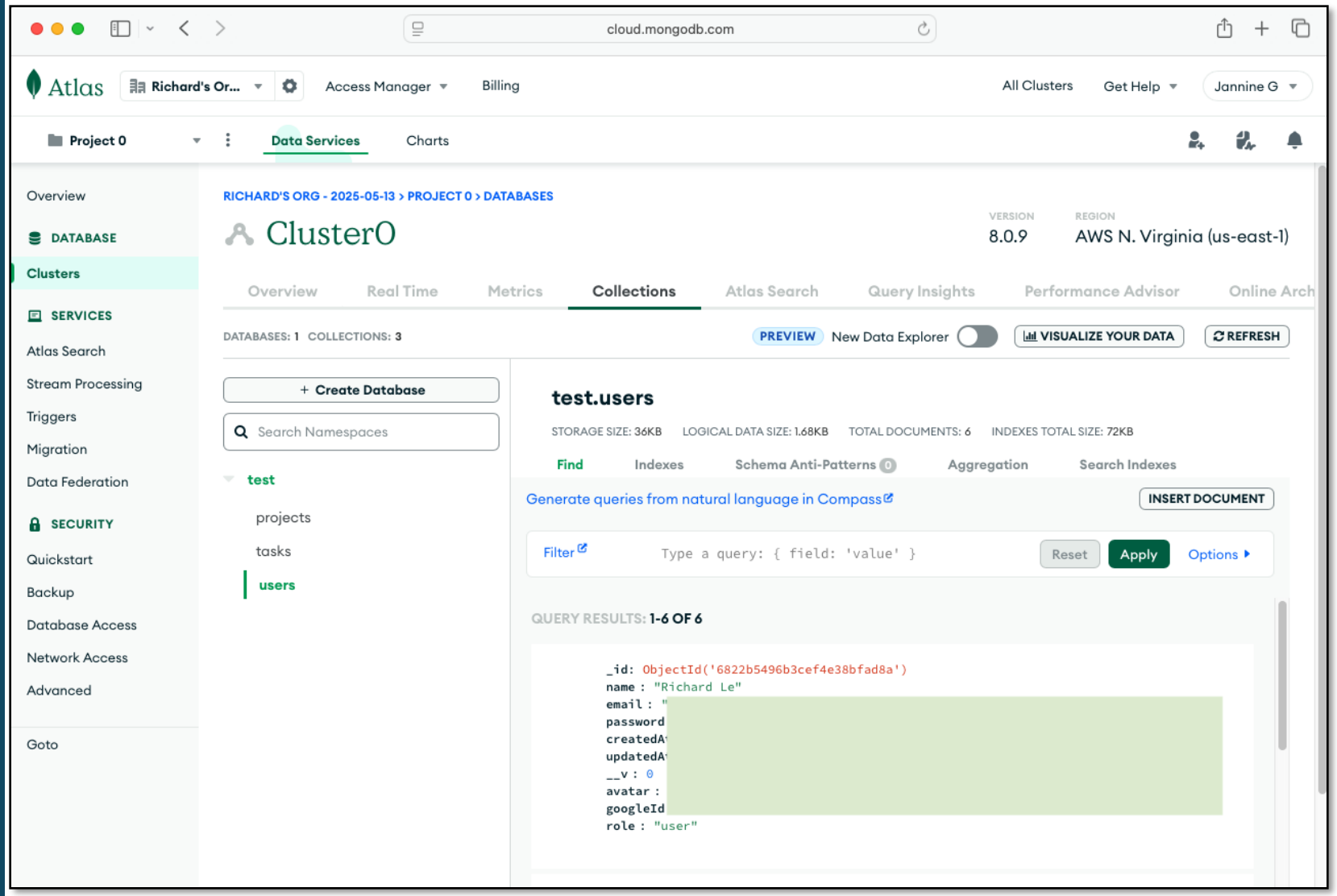
MongoDB Atlas



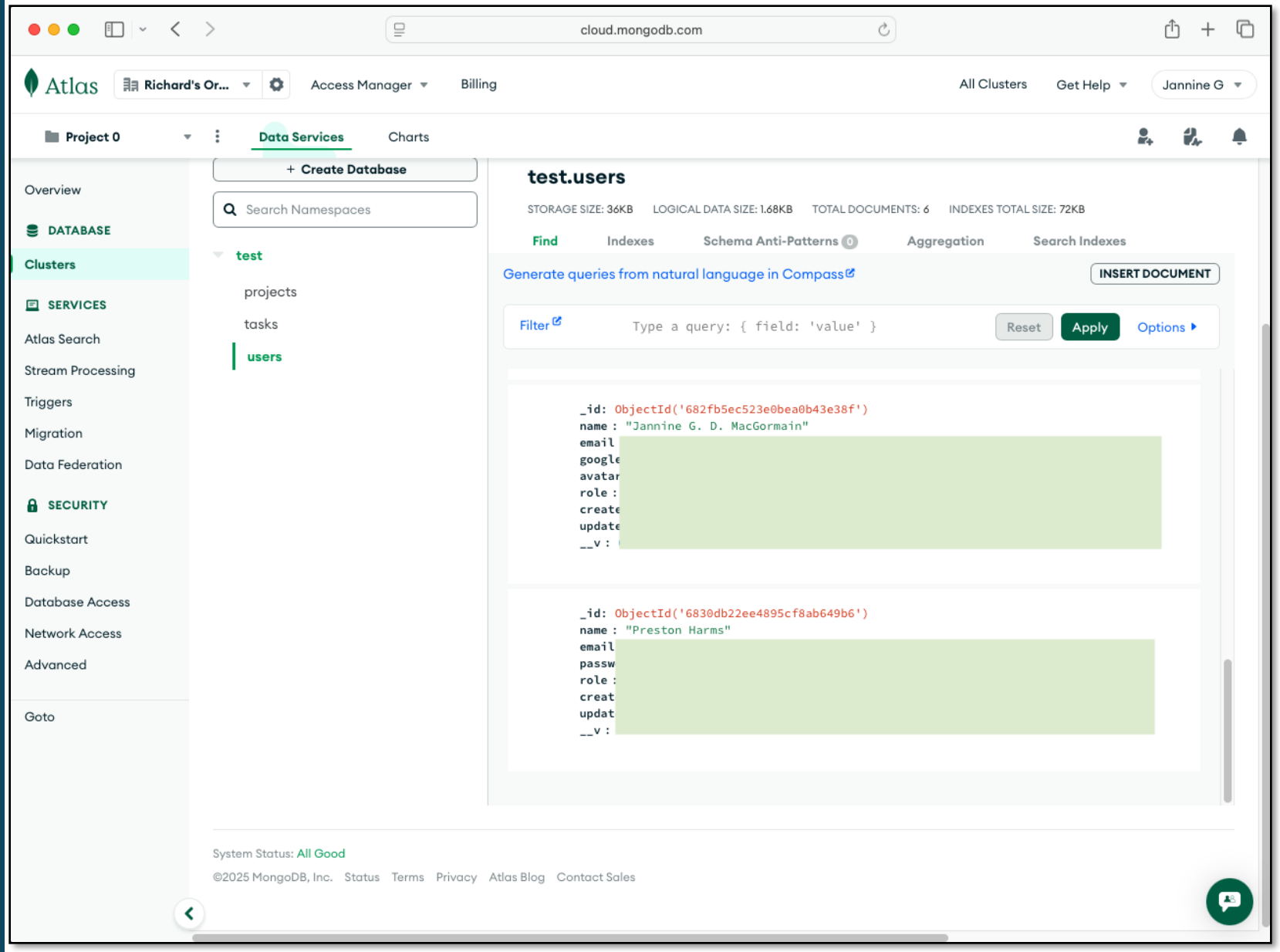




Database


cloud.mongodb.com
Atlas
Richard's Or...
Access Manager
Billing
All Clusters
Get Help
Jannine G
Project 0
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
RICHARD'S ORG - 2025-05-13 > PROJECT 0 > DATABASES
Cluster0
VERSION
8.0.9
REGION
AWS N. Virginia (us-east-1)
Overview
Real Time
Metrics
Collections
Atlas Search
Query Insights
Performance Advisor
Online Arch
DATABASES: 1 COLLECTIONS: 3
PREVIEW
New Data Explorer
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
test
projects
tasks
users
test.users
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 1.68KB
TOTAL DOCUMENTS: 6
INDEXES TOTAL SIZE: 72KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
QUERY RESULTS: 1-6 OF 6
_id: ObjectId('6822b5496b3cef4e38bfad8a')
name : "Richard Le"
email :
password
createdA
updatedA
__v : 0
avatar :
googleId
role : "user"

Database


cloud.mongodb.com
Atlas
Richard's Or...
Access Manager
Billing
All Clusters
Get Help
Jannine G
Project 0
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
+ Create Database
Search Namespaces
test
projects
tasks
users
test.users
STORAGE SIZE: 36KB
LOGICAL DATA SIZE: 1.68KB
TOTAL DOCUMENTS: 6
INDEXES TOTAL SIZE: 72KB
Find
Indexes
Schema Anti-Patterns 0
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
_id: ObjectId('682fb5ec523e0bea0b43e38f')
name : "Jannine G. D. MacGormain"
email
google
avatar
role :
create
update
__v :
_id: ObjectId('6830db22ee4895cf8ab649b6')
name : "Preston Harms"
email
passw
role :
creat
updat
__v :
System Status: All Good
©2025 MongoDB, Inc.
Status
Terms
Privacy
Atlas Blog
Contact Sales

Google Developer

developers.google.com

Google Identity | Authentication | Authorization | Cross-platform | Cross-Account Protection (RISC) | Search | Language

Google Account Authorization

Google Account Authorization | App verification to use Google Authorization APIs | Google Account Linking | Resources

Filter

Google account authorization
- Overview
- Cross-client identity
- OAuth 2.0 scopes
- OAuth 2.0 policies

Authorization considerations by app type
- for Server-side Web Apps
- for JavaScript Web Apps
- for Mobile & Desktop Apps
- for TV & Device Apps
- for Service Accounts

Authorization code samples
- Windows
- iOS
- Android

Home > Products > Google Identity > Google Account Authorization

Was this helpful? | Send feedback

Using OAuth 2.0 to Access Google APIs

Note: Use of Google's implementation of OAuth 2.0 is governed by the OAuth 2.0 Policies.

Google APIs use the OAuth 2.0 protocol for authentication and authorization. Google supports common OAuth 2.0 scenarios such as those for web server, client-side, installed, and limited-input device applications.

To begin, obtain OAuth 2.0 client credentials from the Google API Console. Then your client application requests an access token from the Google Authorization Server, extracts a token from the response, and sends the token to the Google API that you want to access. For an interactive demonstration of using OAuth 2.0 with Google (including the option to use your own client credentials), experiment with the OAuth 2.0 Playground.

This page gives an overview of the OAuth 2.0 authorization scenarios that Google supports, and provides links to more detailed content. For details about using OAuth 2.0 for authentication, see OpenID Connect.

Note: Given the security implications of getting the implementation correct, we strongly encourage you to use OAuth 2.0 libraries when interacting with Google's OAuth 2.0 endpoints. It is a best practice to use well-debugged code provided by others, and it will help you protect yourself and your users. For more information, see Client libraries.

Basic steps

All applications follow a basic pattern when accessing a Google API using OAuth 2.0. At a high level, you follow five steps:

1. Obtain OAuth 2.0 credentials from the Google API Console.

Visit the Google API Console to obtain OAuth 2.0 credentials such as a client ID and client secret that are known to both Google and your application. The set of values varies based on what type of application you are building. For example, a JavaScript application does not require a secret, but a web server application does.

You must create an OAuth client appropriate for the platform on which your app will run, for example:

- For server-side or JavaScript web apps use the "web" client type. Do not use this client type for any other application, such as native or mobile apps.
- For Android apps, use the "Android" client type.
- For iOS and macOS apps, use the "iOS" client type.
- For Universal Windows Platform apps, use the "Universal Windows Platform" client type.
- For limited input devices, such as TV or embedded devices, use the "TVs and Limited Input devices" client type.
- For server-to-server interactions, use service accounts.

Page info

On this page

Basic steps
1. Obtain OAuth 2.0 credentials from the Google API Console.
2. Obtain an access token from the Google Authorization Server.
3. Examine scopes of access granted by the user.
4. Send the access token to an API.
5. Refresh the access token, if necessary.

Scenarios
- Web server applications
- Installed applications
- Client-side (JavaScript) applications
- Applications on limited-input devices
- Service accounts

Token size

Refresh token expiration
- Dealing with session control policies for Google Cloud Platform (GCP) organizations

Client libraries

Key Takeaways

AI-GENERATED

- Google APIs use OAuth 2.0 for secure access to user data, requiring applications to obtain credentials and access tokens.
- Different application types have specific OAuth 2.0 flows, including web server, installed, client-side, and service accounts.
- Refresh tokens enable long-term access but can expire due to various factors, such as user revocation or inactivity.
- Service accounts allow server-to-server interactions without user consent using JWTs for authentication.
- Client libraries are available to simplify OAuth 2.0 implementation and ensure security best practices.

console.cloud.google.com

Enabled APIs & services – APIs & Services – Google Cloud console

Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more | Dismiss | Start free

Google Cloud | Select a project | Search (/) for resources, docs, products, and more | Search

APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Page usage agreements

Enabled APIs & services

To view this page, select a project. | Create project

Google Cloud

Welcome [redacted]

Create and manage your Google Cloud instances, disks, networks, and other resources in one place.

J [redacted] | Switch account

Country

United States

Terms of Service

☑ I agree to the Google Cloud Platform Terms of Service, and the terms of service of any applicable services and APIs.

Agree and continue

Authentication

To securely share credentials within the team:

1. **Never commit .env files** containing actual credentials to the repository
2. Use the provided `.env.example` files as templates
3. Share actual credentials through secure channels:
   - Password manager (LastPass, 1Password, etc.)
   - Encrypted messaging (Signal, WhatsApp)
   - Secure team communication tools (Slack direct messages)
4. Each team member should maintain their own `.env` files with the shared credentials
5. If you need access to credentials, contact one of the team **TaskFlow** members

Server and Client *Overview*

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/server$ micro .env
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/server$ cat .env
# # Server Configuration
```

Must NEVER BE COMMITTED to GitHub or other public repositories for security reasons

Server and Client *Overview*

```
parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ cd client
parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ cat .env
# # Google OAuth
```

Must NEVER BE COMMITTED to GitHub or other public repositories for security reasons

Server and Client *Overview*

```
parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/server$ npm run dev

> taskflow-server@1.0.0 dev
> nodemon server.js

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Server running in development mode on port 5000
MongoDB Connected: ac-xuhdepx-shard-00-02.5kfnod8.mongodb.net
--- Google Login Request Received ---
Request Headers: {
```

Server and Client *Overview*

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management/client$ npm start

> taskflow-client@0.1.0 start
> react-scripts start
```

parallels@ubuntu-linux-2404: ~/Desktop/TaskFlow-student-project-management/client

Firefox Just now
Firefox
"TaskFlow — Mozilla Firefox" is ready

```
Compiled successfully!

You can now view taskflow-client in the browser.

  Local:            http://localhost:3000


Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Files successfully emitted, waiting for typecheck results...
Issues checking in progress...
No issues found.
```

API *Endpoints*

Authentication

- `POST /api/auth/register` - User registration
- `POST /api/auth/login` - User login
- `POST /api/auth/google` - Google OAuth login
- `GET /api/auth/me` - Get current user

Projects

- `GET /api/projects` - Get all projects for user
- `POST /api/projects` - Create a new project
- `GET /api/projects/:id` - Get a specific project
- `PUT /api/projects/:id` - Update a project
- `DELETE /api/projects/:id` - Delete a project
- `POST /api/projects/:id/members` - Add a member to a project

Tasks

- `GET /api/tasks/project/:projectId` - Get all tasks for a project
- `POST /api/tasks` - Create a new task
- `GET /api/tasks/:id` - Get a specific task
- `PUT /api/tasks/:id` - Update a task
- `DELETE /api/tasks/:id` - Delete a task

GitHub Integration

- `GET /api/github/auth-url` - Get GitHub OAuth authorization URL
- `POST /api/github/token` - Exchange OAuth code for access token
- `GET /api/github/status` - Check GitHub connection status
- `DELETE /api/github/disconnect` - Disconnect GitHub account
- `GET /api/github/repositories` - Get user's GitHub repositories
- `GET /api/github/repository/:owner/:repo` - Get repository details
- `GET /api/github/repository/:owner/:repo/issues` - Get repository issues
- `POST /api/projects/:id/github` - Connect GitHub repository to project

External APIs

```
Checking if user exists with email:
User not found. Creating new user...
New user created: {
  name: 'Jannine G. D. MacGormain',
  email
  googl
  avata
  role:
  _id:
  creat
  updat
  __v:
  id: '
}
Sending token response...
--- Google Login Request Completed Successfully ---
POST /api/auth/google 200 307.256 ms - 500
GET /api/projects 200 177.294 ms - 36
GET /api/projects 200 184.545 ms - 36
POST /api/projects 201 184.030 ms - 324
GET /api/projects 200 264.297 ms - 561
GET /api/projects 200 272.845 ms - 561
GET /api/projects/682fb620523e0bea0b43e396 200 274.357 ms - 560
GET /api/tasks/project/682fb620523e0bea0b43e396 200 282.500 ms - 36
GET /api/projects/682fb620523e0bea0b43e396 200 907.891 ms - 560
GET /api/tasks/project/682fb620523e0bea0b43e396 304 284.692 ms - -
POST /api/projects/682fb620523e0bea0b43e396/members 200 379.474 ms - 478
GET /api/projects/682fb620523e0bea0b43e396 200 270.954 ms - 905
POST /api/projects/682fb620523e0bea0b43e396/members 404 269.496 ms - 44
GET /api/auth/me 200 193.942 ms - 410
GET /api/auth/me 200 194.156 ms - 410
GET /api/projects 200 272.398 ms - 715
GET /api/projects 200 281.614 ms - 715
GET /api/auth/me 304 192.898 ms - -
GET /api/auth/me 200 196.454 ms - 410
GET /api/projects 304 275.943 ms - -
GET /api/projects 200 281.667 ms - 715
GET /api/projects/682fb620523e0bea0b43e396 200 290.512 ms - 905
GET /api/projects/682fb620523e0bea0b43e396 304 969.501 ms - -
PUT /api/projects/682fb620523e0bea0b43e396 200 296.885 ms - 507
GET /api/projects 200 265.541 ms - 744
GET /api/projects 200 269.798 ms - 744
```

Running with Docker

```
(venv) parallels@ubuntu-linux-2404:~/Desktop$ sudo usermod -aG docker $USER
(venv) parallels@ubuntu-linux-2404:~/Desktop$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

(venv) parallels@ubuntu-linux-2404:~/Desktop$ sudo apt-get update
Get:1 https://download.docker.com/linux/ubuntu noble InRelease [48.8 kB]
```

```
ubuntu@ip-172-31-5-251:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

ubuntu@ip-172-31-5-251:~$ docker pull janninegdm888/taskflow-server:latest
latest: Pulling from janninegdm888/taskflow-server
98da3b680390: Pull complete
96432eac497b: Pull complete
bab3dec6af72: Pull complete
2f0d35c51a75: Pull complete
378bc28767f6: Pull complete
4ae235f71081: Pull complete
79f57c17da10: Pull complete
06fd204c9f69: Pull complete
033f647901ca: Pull complete
Digest: sha256:0ea9040e3edbeaf000aa53a8dd0df7cb17cecadf9a49cf811177ff3bd1697bab
Status: Downloaded newer image for janninegdm888/taskflow-server:latest
docker.io/janninegdm888/taskflow-server:latest
ubuntu@ip-172-31-5-251:~$ docker pull janninegdm888/taskflow-client:latest
latest: Pulling from janninegdm888/taskflow-client
6e771e15690e: Pull complete
c60e446e49a0: Pull complete
e6557c42ebea: Pull complete
```

Running with Docker

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ sudo docker build -t taskflow-client ./client
[+] Building 22.8s (16/16) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                              0.0s
 => => transferring dockerfile: 487B                                                                              0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                   1.6s
 => [internal] load metadata for docker.io/library/node:18-slim                                                   1.6s
```

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ sudo docker build -t taskflow-server ./server
[+] Building 2.7s (10/10) FINISHED                                                                     docker:default
 => [internal] load build definition from Dockerfile                                                              0.0s
 => => transferring dockerfile: 349B                                                                              0.0s
 => [internal] load metadata for docker.io/library/node:18-slim                                                   0.4s
 => [internal] load .dockerignore                                                                                 0.0s
 => => transferring context: 2B                                                                                   0.0s
```

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ sudo docker-compose up --build
[+] Running 3/3
 ✔ Network taskflow-student-project-management_default     Created                                                0.0s
 ✔ Container taskflow-student-project-management-server-1  Created                                                0.0s
 ✔ Container taskflow-student-project-management-client-1  Created                                                0.0s
Attaching to client-1, server-1
client-1  | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
```

Running with Docker

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker login

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: SPCH-WBWQ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…
WARNING! Your password will be stored unencrypted in /home/parallels/snap/docker/3065/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
```

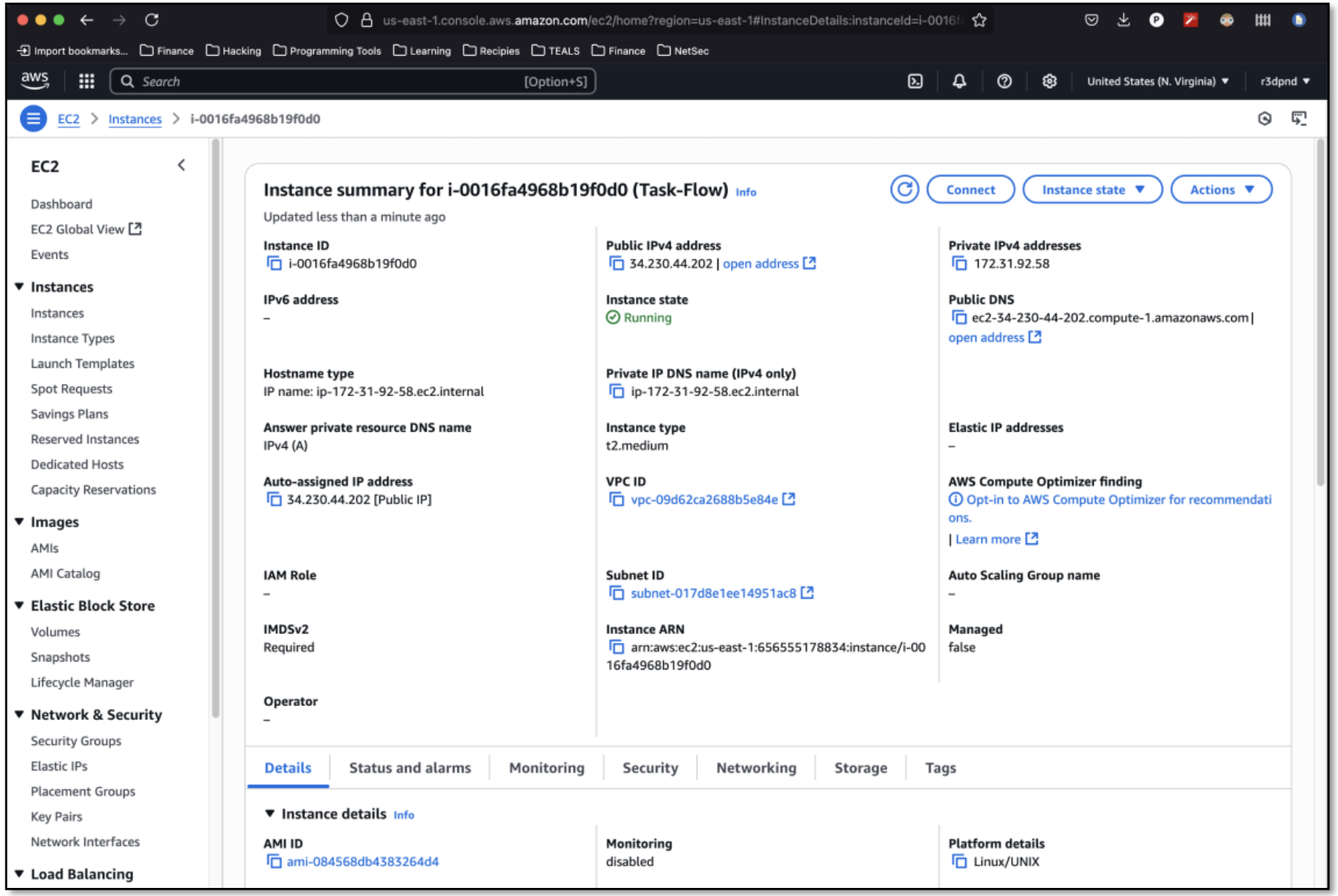
Running with Docker

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker tag taskflow-server:latest  janninegdm888/taskflow-server:latest
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker tag taskflow-client:latest  janninegdm888/taskflow-client:latest
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker push janninegdm888/taskflow-server:latest
The push refers to repository [docker.io/janninegdm888/taskflow-server]
d4901561726d: Pushed
e60a543329b2: Pushed
6fa54343b2dc: Pushed
4c016d4a98b0: Pushed
e5b861e7e74d: Pushed
811e90eb154f: Pushed
970dac0b679a: Pushed
d96d4b0ff1b8: Pushed
41d20f587704: Pushed
latest: digest: sha256:0ea9040e3edbeaf000aa53a8dd0df7cb17cecadf9a49cf811177ff3bd1697bab size: 2205
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker push janninegdm888/taskflow-client:latest
The push refers to repository [docker.io/janninegdm888/taskflow-client]
d93d0c06369c: Pushed
96a0d9b1e80c: Mounted from library/nginx
97ca58bb468d: Mounted from library/nginx
4a651ac2c218: Mounted from library/nginx
0fe7bdfe639d: Mounted from library/nginx
c32437183174: Mounted from library/nginx
9aad78ecf380: Mounted from library/nginx
3459dc8f926a: Mounted from library/nginx
a16e98724c05: Mounted from library/nginx
latest: digest: sha256:9bda83c2ba8f4a84d3bd1b43372c3f0703cf605622e642013eebb77a628857d0 size: 2199
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$
```

Running with Docker

```
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker tag taskflow-server:latest  janninegdm888/taskflow-server:latest
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker tag taskflow-client:latest  janninegdm888/taskflow-client:latest
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker push janninegdm888/taskflow-server:latest
The push refers to repository [docker.io/janninegdm888/taskflow-server]
d4901561726d: Pushed
e60a543329b2: Pushed
6fa54343b2dc: Pushed
4c016d4a98b0: Pushed
e5b861e7e74d: Pushed
811e90eb154f: Pushed
970dac0b679a: Pushed
d96d4b0ff1b8: Pushed
41d20f587704: Pushed
latest: digest: sha256:0ea9040e3edbeaf000aa53a8dd0df7cb17cecadf9a49cf811177ff3bd1697bab size: 2205
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$ docker push janninegdm888/taskflow-client:latest
The push refers to repository [docker.io/janninegdm888/taskflow-client]
d93d0c06369c: Pushed
96a0d9b1e80c: Mounted from library/nginx
97ca58bb468d: Mounted from library/nginx
4a651ac2c218: Mounted from library/nginx
0fe7bdfe639d: Mounted from library/nginx
c32437183174: Mounted from library/nginx
9aad78ecf380: Mounted from library/nginx
3459dc8f926a: Mounted from library/nginx
a16e98724c05: Mounted from library/nginx
latest: digest: sha256:9bda83c2ba8f4a84d3bd1b43372c3f0703cf605622e642013eebb77a628857d0 size: 2199
(venv) parallels@ubuntu-linux-2404:~/Desktop/TaskFlow-student-project-management$
```

Amazon Web Services (AWS) *EC2*

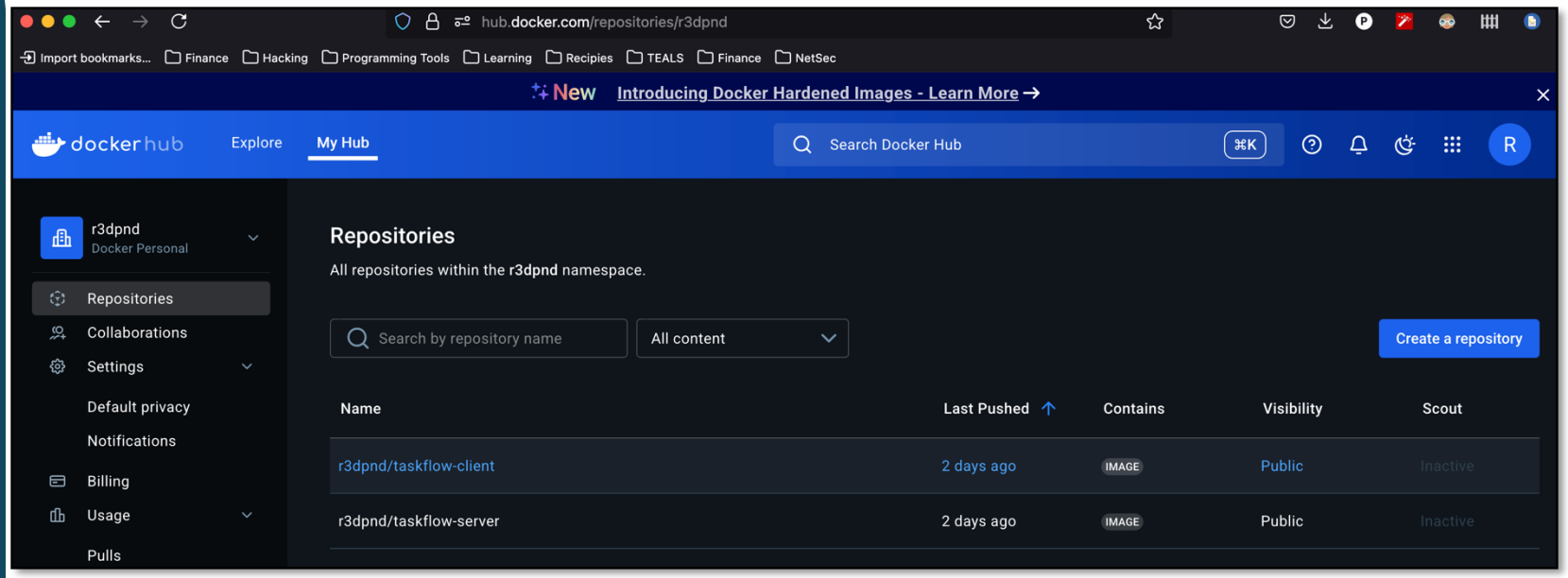


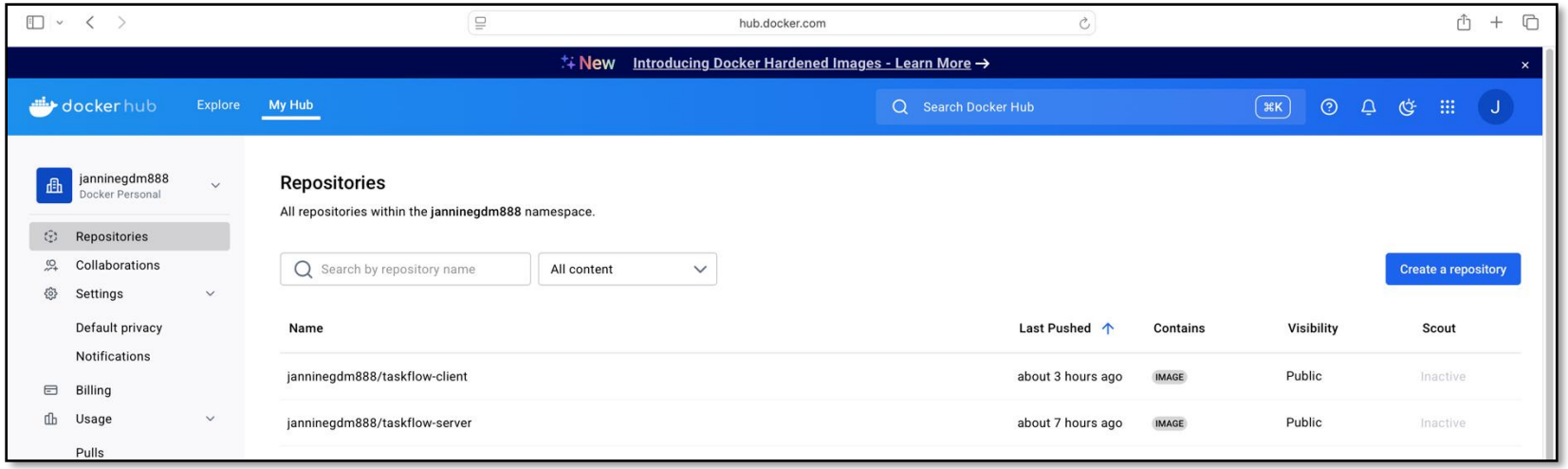
EC2

Deployment

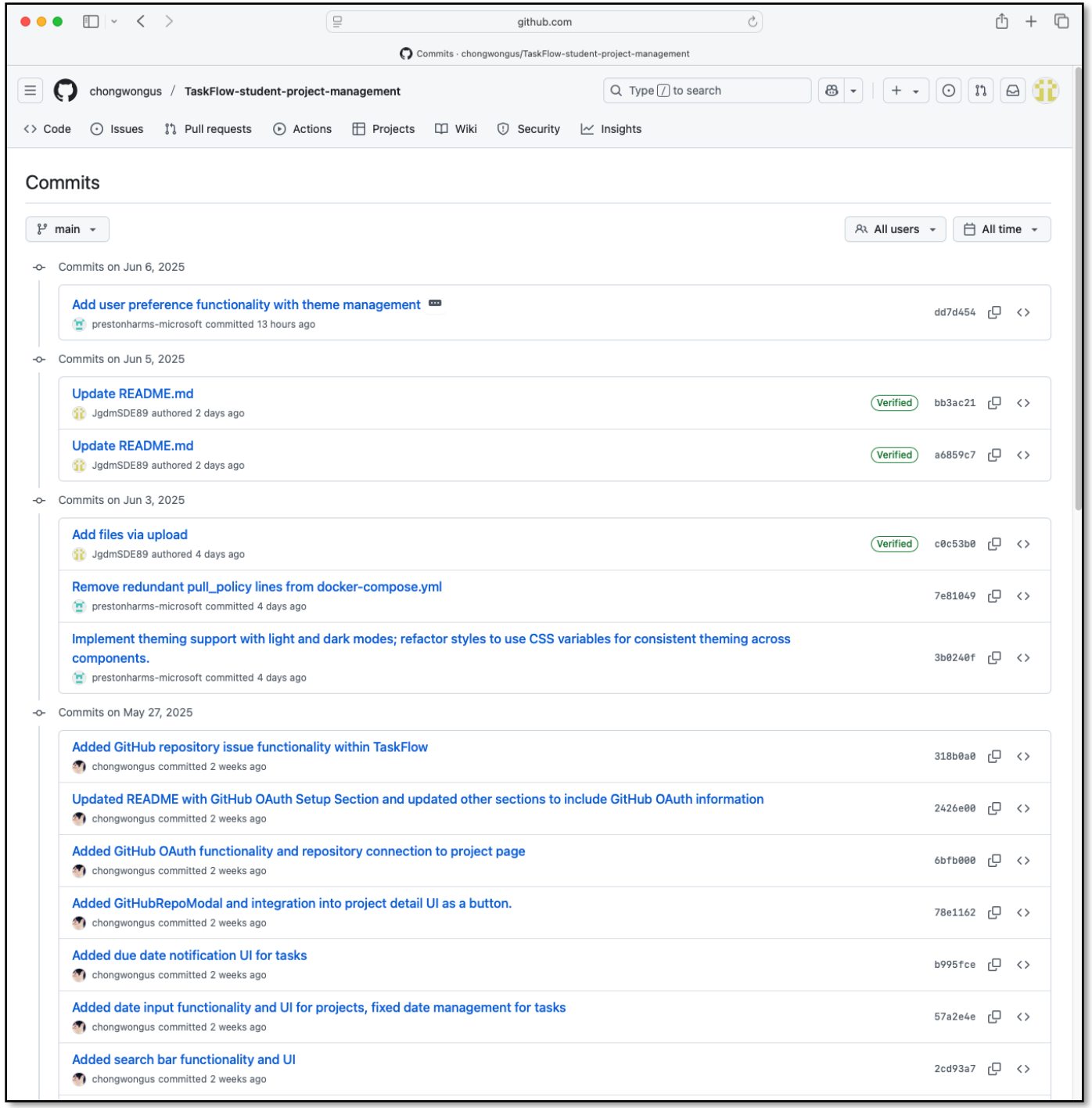
```
ubuntu@ip-172-31-5-251:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

ubuntu@ip-172-31-5-251:~$ docker pull janninegdm888/taskflow-server:latest
latest: Pulling from janninegdm888/taskflow-server
98da3b680390: Pull complete
96432eac497b: Pull complete
bab3dec6af72: Pull complete
2f0d35c51a75: Pull complete
378bc28767f6: Pull complete
4ae235f71081: Pull complete
79f57c17da10: Pull complete
06fd204c9f69: Pull complete
033f647901ca: Pull complete
Digest: sha256:0ea9040e3edbeaf000aa53a8dd0df7cb17cecadf9a49cf811177ff3bd1697bab
Status: Downloaded newer image for janninegdm888/taskflow-server:latest
docker.io/janninegdm888/taskflow-server:latest
ubuntu@ip-172-31-5-251:~$ docker pull janninegdm888/taskflow-client:latest
latest: Pulling from janninegdm888/taskflow-client
6e771e15690e: Pull complete
c60e446e49a0: Pull complete
e6557c42ebea: Pull complete
```

Docker

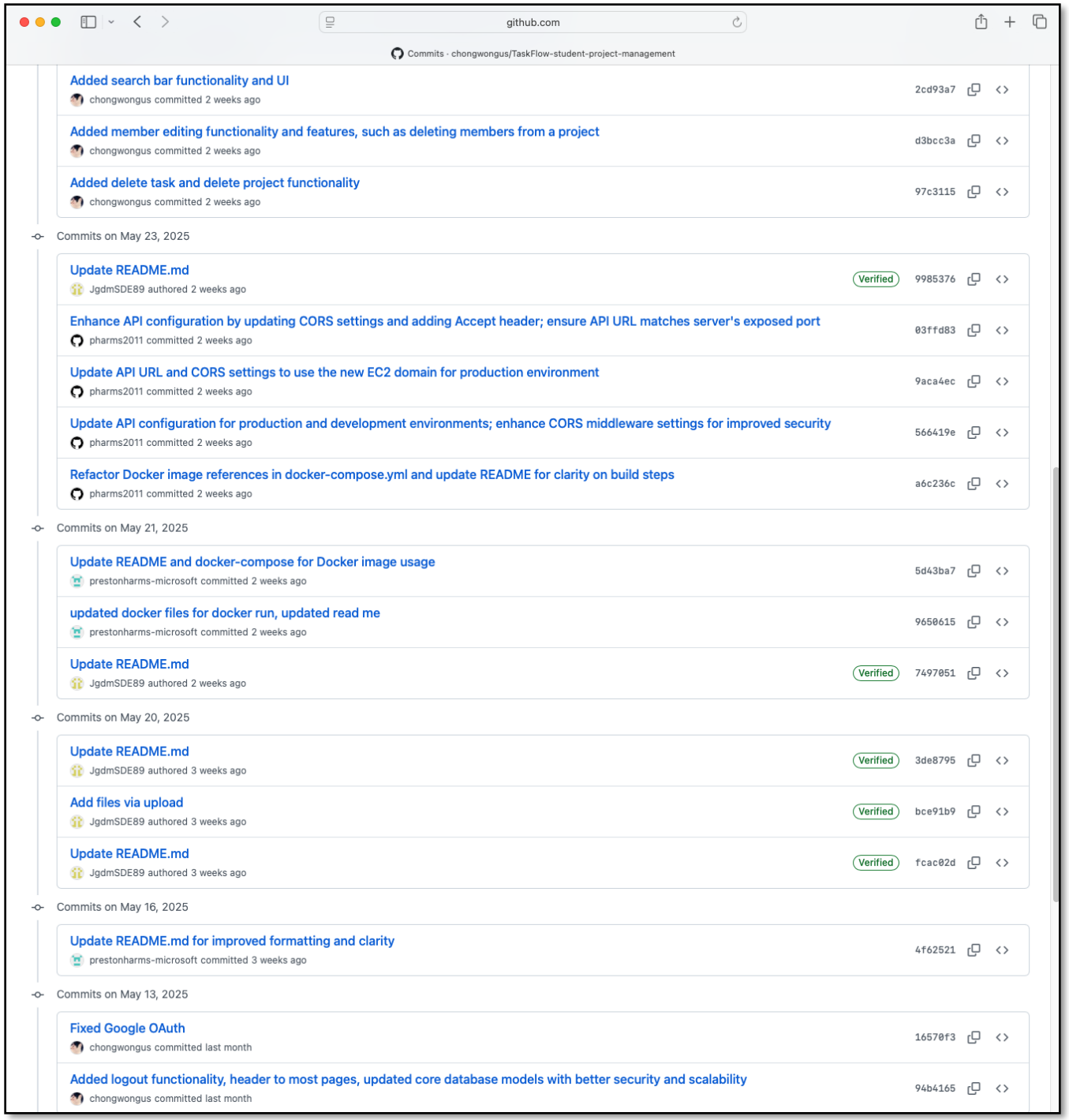

hub.docker.com/repositories/r3dpnd
Import bookmarks... Finance Hacking Programming Tools Learning Recipies TEALS Finance NetSec
New Introducing Docker Hardened Images - Learn More
docker hub
Explore
My Hub
Search Docker Hub
⌘K
r3dpnd
Docker Personal
Repositories
Collaborations
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Repositories
All repositories within the r3dpnd namespace.
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
r3dpnd/taskflow-client
2 days ago
IMAGE
Public
Inactive
r3dpnd/taskflow-server
2 days ago
IMAGE
Public
Inactive


hub.docker.com
New Introducing Docker Hardened Images - Learn More
docker hub
Explore
My Hub
Search Docker Hub
⌘K
janninegdm888
Docker Personal
Repositories
Collaborations
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Repositories
All repositories within the janninegdm888 namespace.
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
janninegdm888/taskflow-client
about 3 hours ago
IMAGE
Public
Inactive
janninegdm888/taskflow-server
about 7 hours ago
IMAGE
Public
Inactive

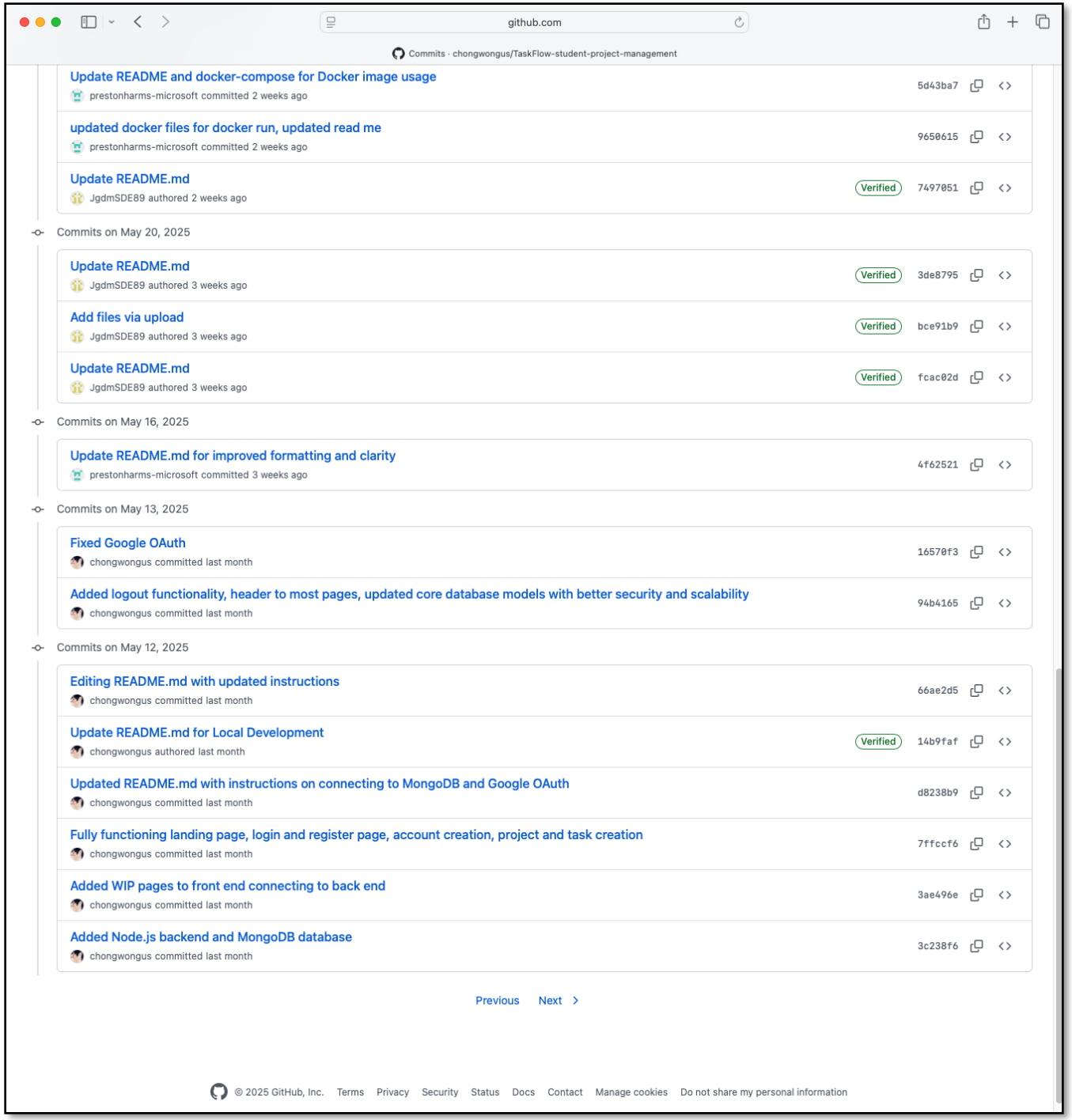
Version Control



Version Control



Version Control



YouTrack Project Management Report

Sprint	Task Cards	Duration
Sprint 1 *(TaskFlow Project Setup Basics)*	5 Task Cards	April 27 – May 10, 2025
Sprint 2 *(Development and Design Phase)*	7 Task Cards	May 11 – May 24, 2025
Sprint 3 *(Final Development and Design Phase for Project Submission)*	9 Task Cards	May 25 – June 07, 2025

Sprints (*via Task Cards*)

jgdm-sde2425.youtrack.cloud

YouTrack

Projects / TeamTaskFlow(TCSS 506)

New Project

- Issues
- Dashboards
- Agile Boards
- Reports
- Projects
- Knowledge Base
- Timesheets
- Gantt Charts

TeamTaskFlow(TCSS 506)

New Issue

Add widget | Import from dashboard

TeamTaskFlow(TCSS 506) Team ³

PROJECT OWNER

JG Jannine G. D. MacGormain
[redacted]@yahoo.com

TEAM

LE ler23
ler23@uw.edu

PH pharms
pharms@uw.edu

project: {TeamTaskFlow(TCSS 506)} State: Unres...

- N TT5-5 Setup YouTrack TeamTaskFlow(TCSS 506) project.
- N TT5-4 Project Approval Meeting.
- N TT5-3 Add Dockerfile and Flask application.
- N TT5-2 TaskFlow SRS Draft.
- N TT5-1 Setup GitHub repository for the TaskFlow project's version control.

TT5

Issues
Articles

Agile Boards

TeamTaskFlow(TCSS 506) Project Development
TeamTaskFlow(TCSS 506) Project Management

Administration

General Info
Access
Team

Sprints (*via Task Cards*)

jgdm-sde2425.youtrack.cloud

YouTrack

Issues
Dashboards
Agile Boards
Reports
Projects
Knowledge Base
Timesheets
Gantt Charts

Agile Boards / TeamTaskFlow(TCSS 506) Project Development

Filter cards on the board | New card

Sprint 1 - TaskFlow projec... | 27 Apr — 10 May | Start sprint | 6 days left

S XL TV

Open 0 | In Progress 0 | To Verify 0 | Done 0

Uncategorized Cards — 5 cards

TT5-1 Setup GitHub repository for the TaskFlow project's version control.
Normal Task ? Not estimated

TT5-2 TaskFlow SRS Draft.
Normal Task ? Not estima

TT5-3 Add Dockerfile and Flask application.
Normal Task ? Not estima

TT5-4 Project Approval Meeting.
Normal Task ? Not estima

TT5-5 Setup YouTrack TeamTaskFlow(TCSS 506 project.
Normal Task ? Not estima

Edit Sprint

Sprints on this board are linked to the Sprints custom field. Changes to sprints are applied to the values that are stored in this field in all of the projects that use this field. Changes are only applied when the owner of the board has permission to update the set of values for this field.

Sprint Name: Sprint 1 - TaskFlow project setup basics

Goal: Setup GitHub repository for the TaskFlow project's version control, TaskFlow SRS Draft, Add Dockerfile and Flask application, Project Approval Meeting,

Schedule: 27 Apr — 10 May

Apply changes | Cancel | Archive | Delete

Sprints (*via Task Cards*)

YouTrack

Agile Boards / TeamTaskFlow(TCSS 506) Project Development

Filter cards on the board | New card

Issues
Dashboards
Agile Boards
Reports
Projects
Knowledge Base
Timesheets
Gantt Charts

Sprint 1 - TaskFlow projec... | 27 Apr — 10 May | Start sprint

S — XL TV

Open 0	In Progress 0	To Verify 0	Done 1

Uncategorized Cards

4 cards

~~TT5-3~~ Add Dockerfile and Flask application.
PH Normal Task ? 1

~~TT5-4~~ Project Approval Meeting.
JG Normal Task ? Not estimated

~~TT5-2~~ TaskFlow SRS Draft.
JG Normal Task ? Not estimated

~~TT5-5~~ Setup YouTrack TeamTaskFlow(TCSS 506) project.
JG Normal Task ? Not estimated

New Sprint

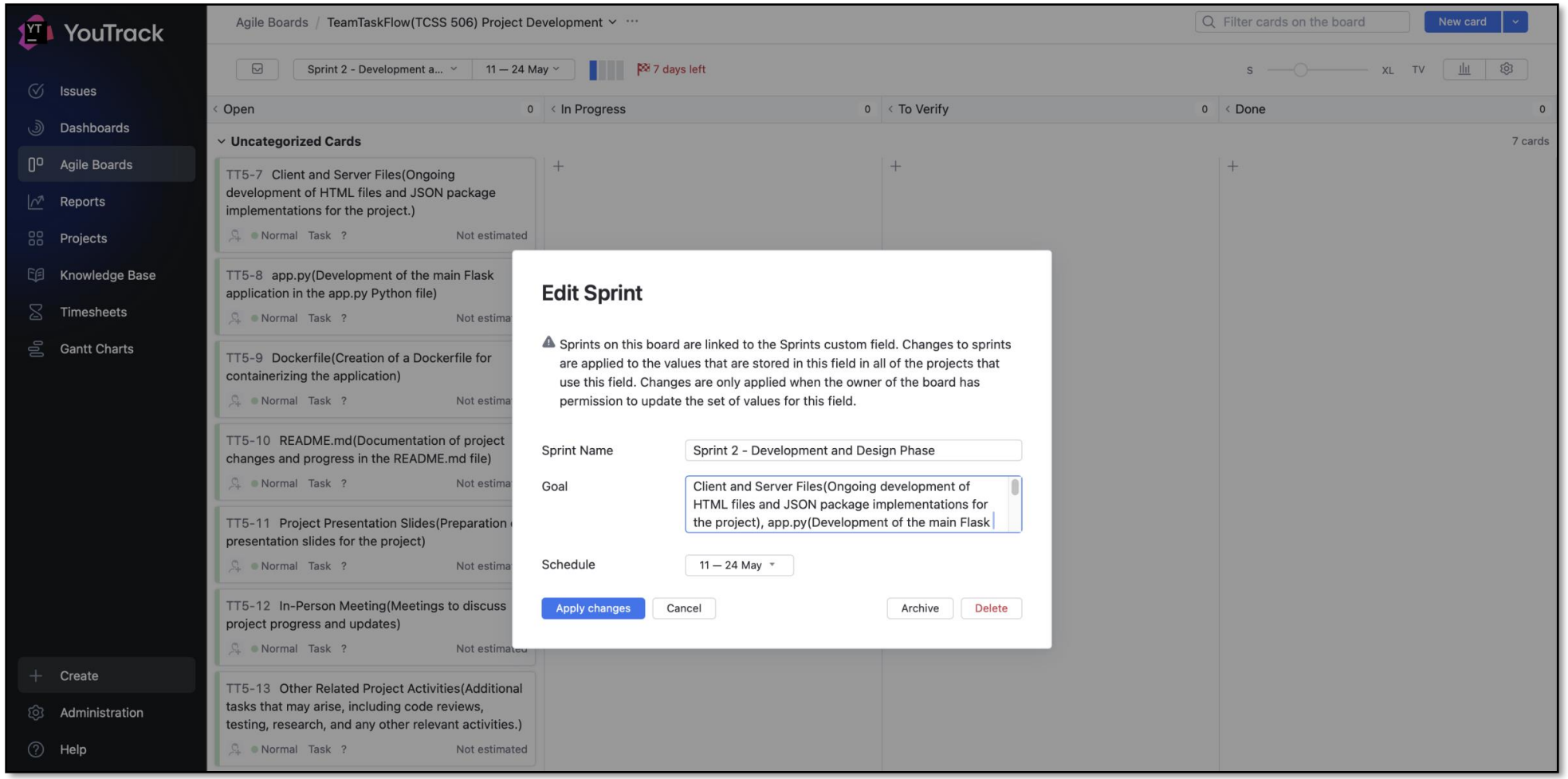
Sprint Name: Sprint 2 - Development and Design Phase

Goal: Client and Server Files(Ongoing development of HTML files and JSON package implementations for the project), app.py(Development of the main Flask

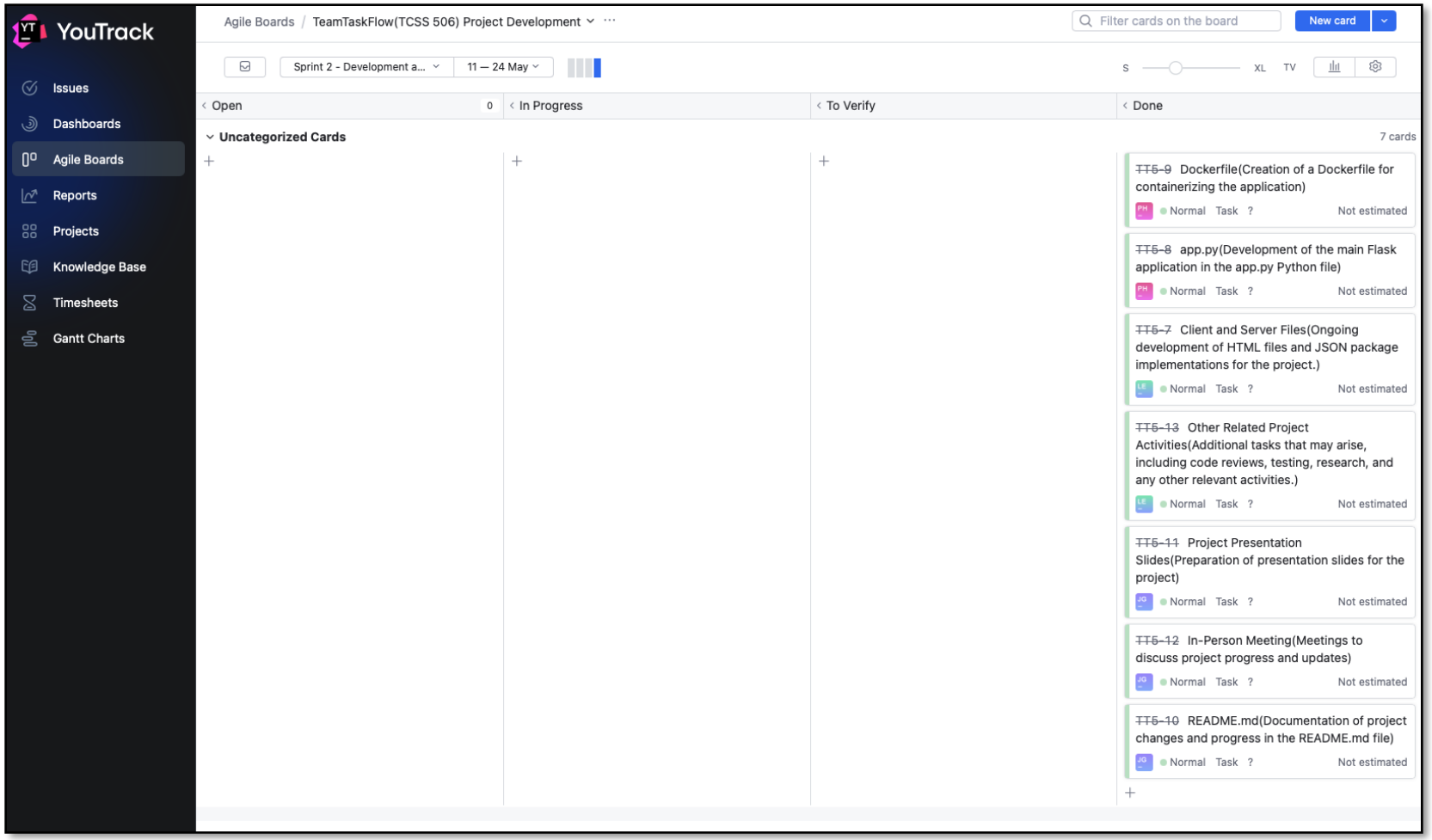
Schedule: 11 — 24 May

Create sprint | Cancel

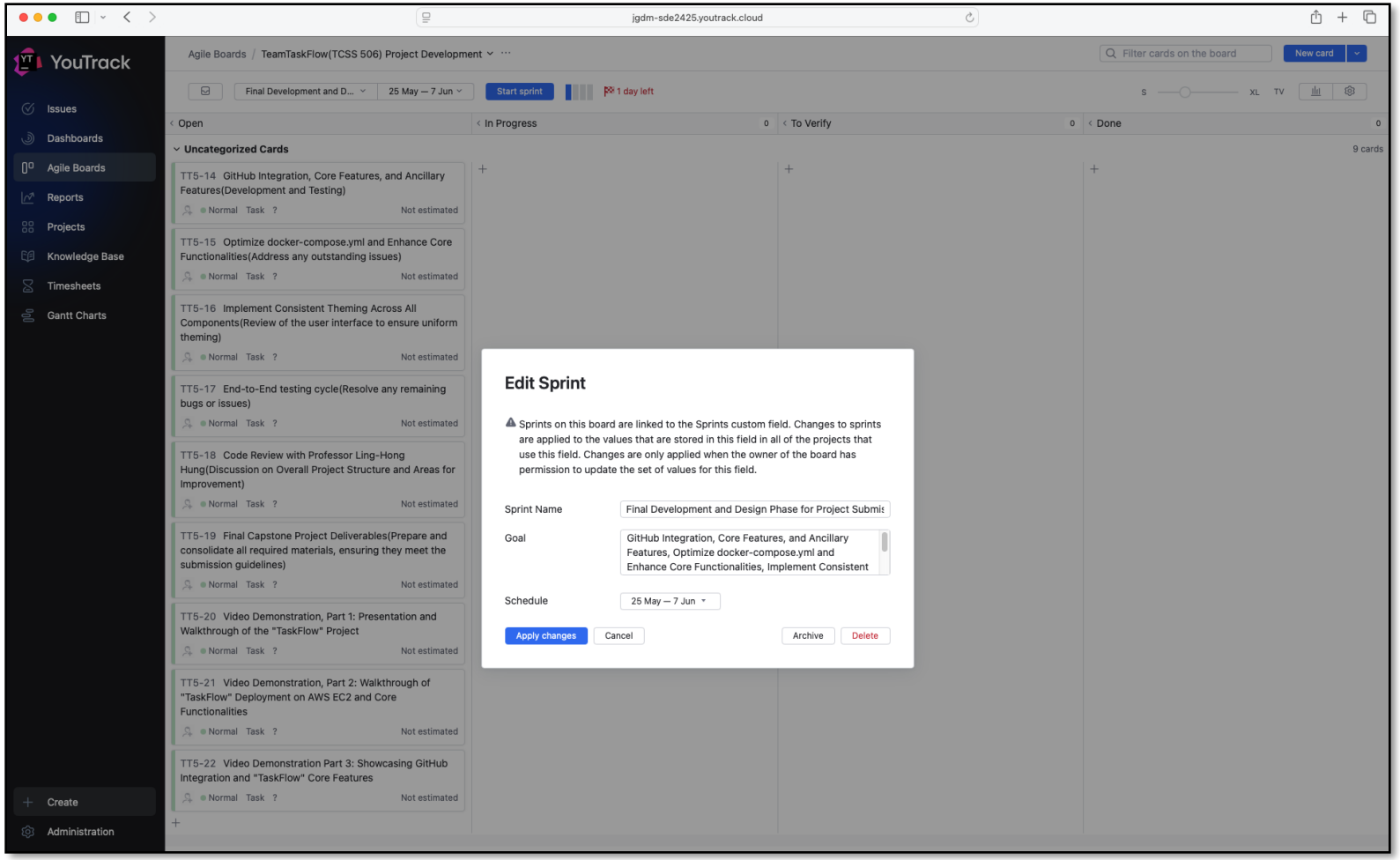
Sprints *(via Task Cards)*



Sprints (*via Task Cards*)



Sprints (*via Task Cards*)



Sprints (*via Task Cards*)

jgdm-sde2425.youtrack.cloud

YouTrack

- Issues
- Dashboards
- Agile Boards
- Reports
- Projects
- Knowledge Base
- Timesheets
- Gantt Charts

Agile Boards / TeamTaskFlow(TCSS 506) Project Development

Filter cards on the board | New card

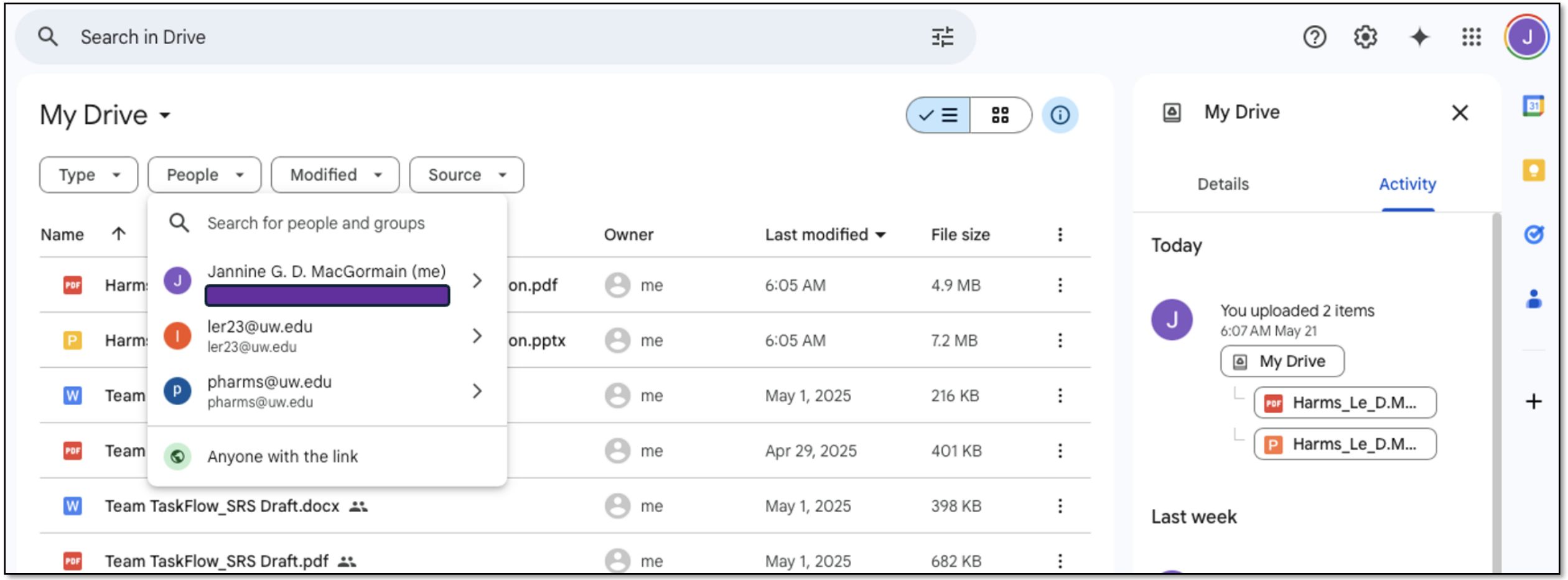
Final Development and D... | 25 May — 7 Jun | Start sprint | 1 day left

S — XL TV

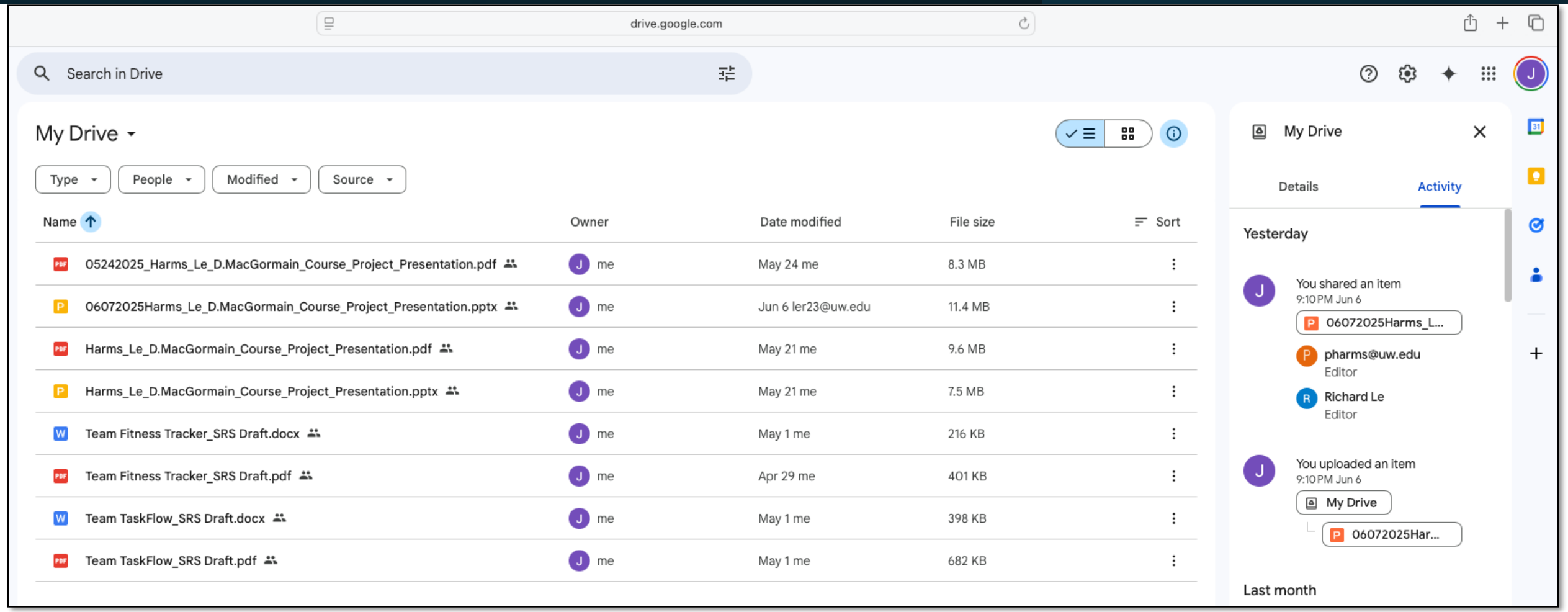
Uncategorized Cards — 9 cards

Open 0	In Progress 0	To Verify 0	Done 0
	TT5-17 End-to-End testing cycle(Resolve any remaining bugs or issues) LE Normal Task ? Not estimated	TT5-20 Video Demonstration, Part 1: Presentation and Walkthrough of the "TaskFlow" Project JG Normal Task ? Not estimated	~~TT5-14~~ GitHub Integration, Core Features, and Ancillary Features(Development and Testing) LE Normal Task ? Not estimated
	TT5-18 Code Review with Professor Ling-Hong Hung(Discussion on Overall Project Structure and Areas for Improvement) JG Normal Task ? Not estimated	TT5-21 Video Demonstration, Part 2: Walkthrough of "TaskFlow" Deployment on AWS EC2 and Core Functionalities PH Normal Task ? Not estimated	~~TT5-15~~ Optimize docker-compose.yml and Enhance Core Functionalities(Address any outstanding issues) PH Normal Task ? Not estimated
	TT5-19 Final Capstone Project Deliverables(Prepare and consolidate all required materials, ensuring they meet the submission guidelines) JG Normal Task ? Not estimated	TT5-22 Video Demonstration Part 3: Showcasing GitHub Integration and "TaskFlow" Core Features LE Normal Task ? Not estimated	~~TT5-16~~ Implement Consistent Theming Across All Components(Review of the user interface to ensure uniform theming) PH Normal Task ? Not estimated

Shared Documents via Google



Shared Documents via Google



Basic *Commands*

For Python Setup

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt install python3-pip
- sudo apt install python3.12-venv
- python3 -m venv venv
- source venv/bin/activate
- pip install Flask

For Docker Installation

- sudo apt-get install -y apt-transport-https ca-certificates curl software-properties-common
- sudo mkdir -p /etc/apt/trusted.gpg.d
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee /etc/apt/trusted.gpg.d/docker.gpg > /dev/null
- sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
- sudo apt-get install -y docker-ce
- sudo usermod -aG docker $USER
- sudo docker run hello-world

For Git and GitHub Setup

- sudo apt-get install git
- git config --global user.name "Your user name"
- git config --global user.email "Your user email"
- git init
- git add <file_name>
- git commit -m "Your commit message"
- git push origin <branch_name>
- git pull
- git status

Checking Versions

- flask –version
- docker –version
- git –version

Testing

Run tests with:

```
# Backend tests
cd backend
npm test

# Frontend tests
cd frontend
npm test
```

Troubleshooting

MongoDB Connection Issues

- Verify your IP address is in the MongoDB Atlas whitelist
- Check that your username and password are correct
- Ensure the connection string format is correct

Google OAuth Issues

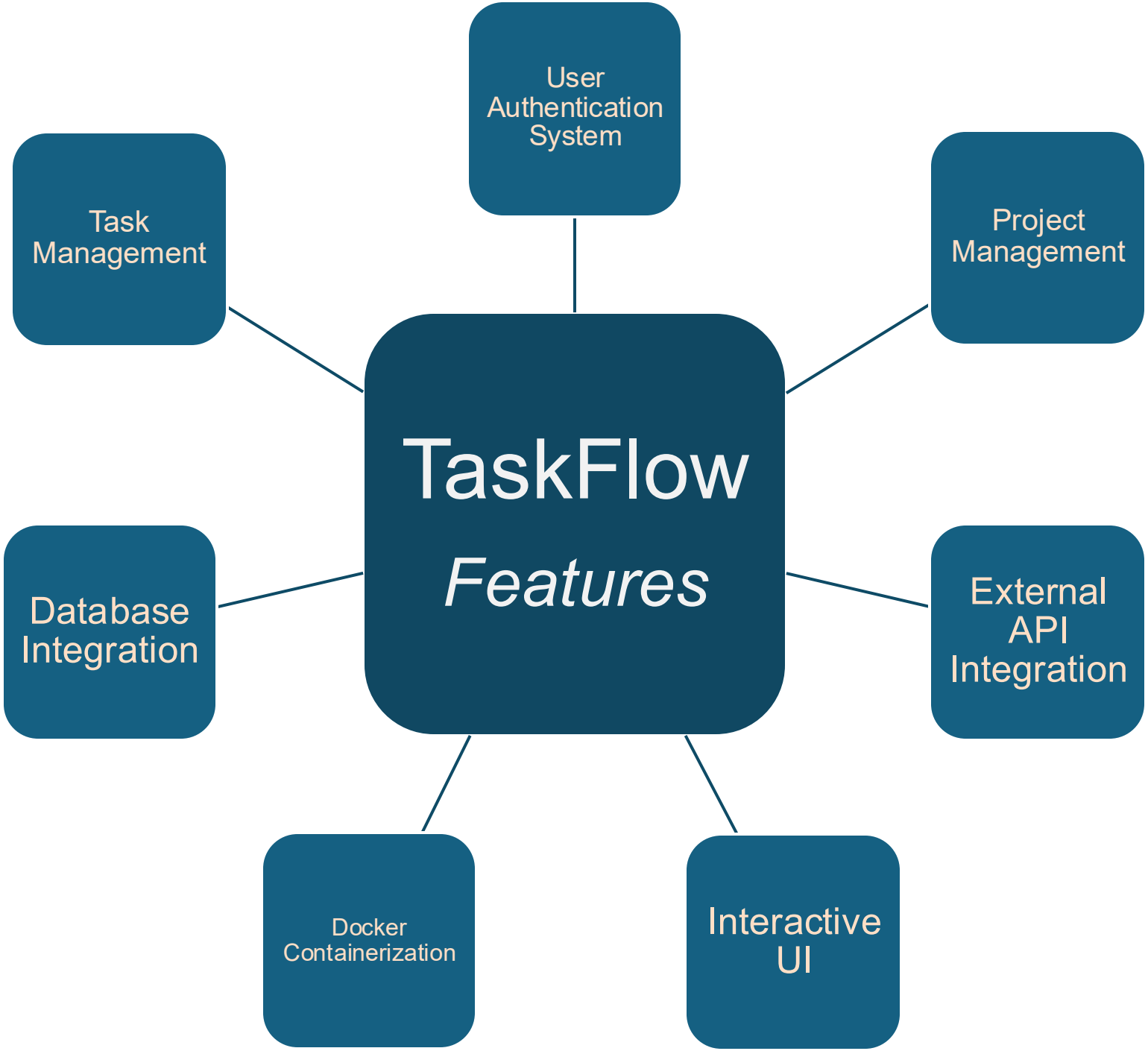
- Verify the correct Google Client ID is in your .env file
- Ensure `http://localhost:3000` is added to authorized JavaScript origins
- Check browser console for any CORS or authorization errors

GitHub OAuth Issues

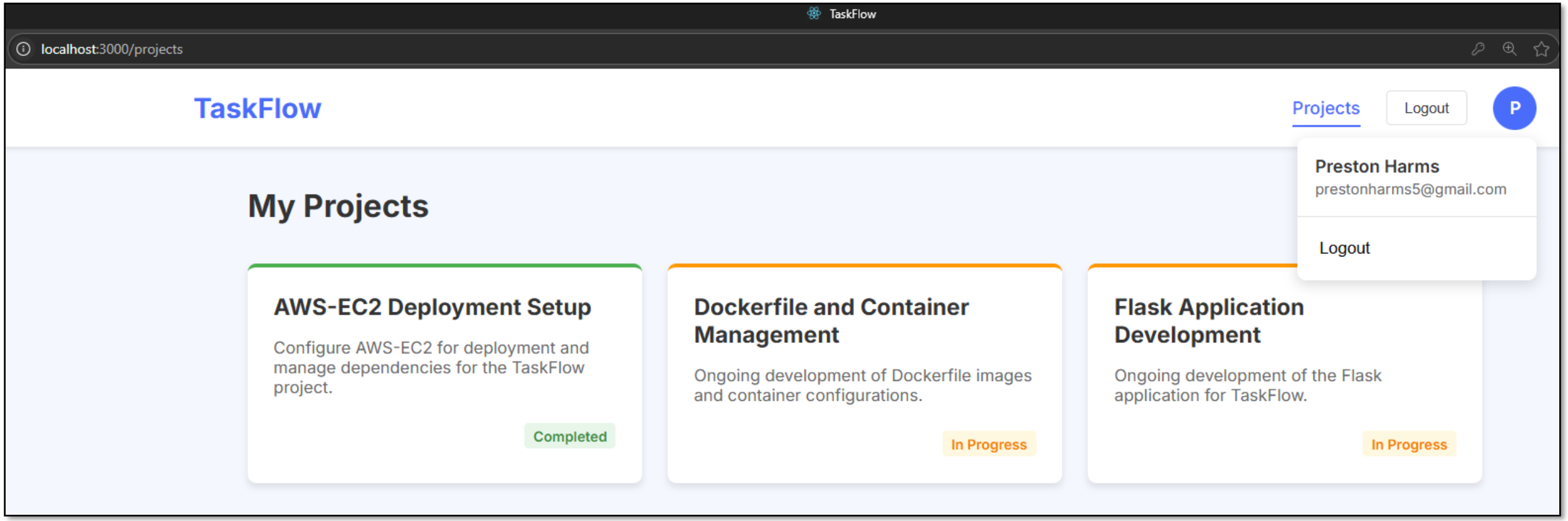
- Verify the GitHub Client ID and Secret are correctly set in your `.env` file
- Ensure the Authorization callback URL in your GitHub OAuth App matches exactly: `http://localhost:3000/github/callback`
- Check that `CLIENT_URL` is set to `http://localhost:3000` in your backend `.env`
- If you see "redirect_uri mismatch" errors, make sure there are no trailing slashes in your callback URL

API Connection Issues

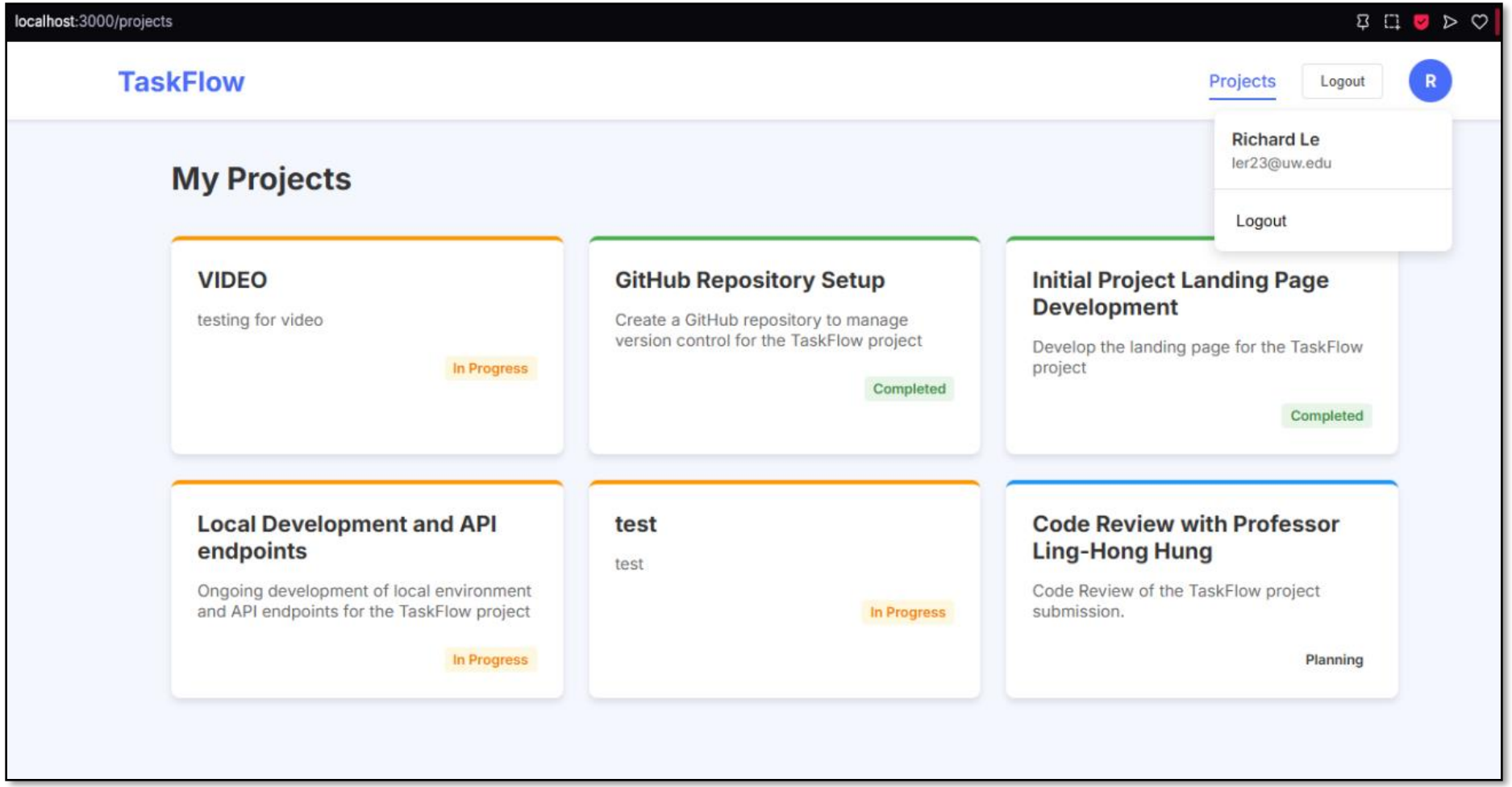
- Verify the backend server is running on port 5000
- Check that CORS is properly configured in the backend
- Ensure the frontend is using the correct API URL


TaskFlow
Features
User Authentication System
Task Management
Project Management
Database Integration
External API Integration
Docker Containerization
Interactive UI

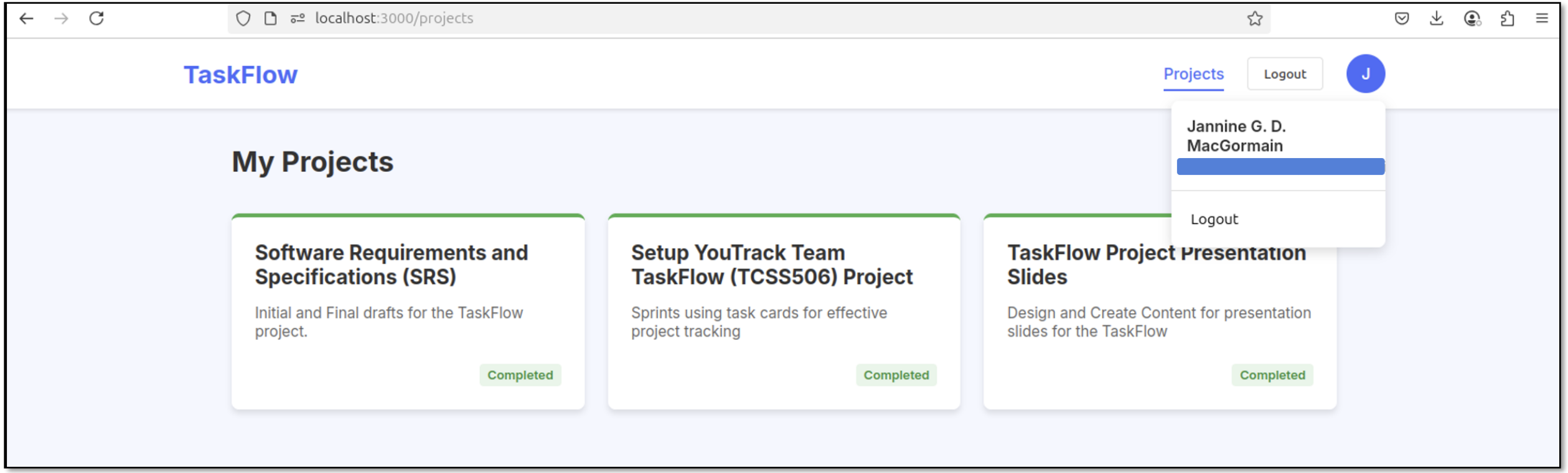
Team **TaskFlow** *Projects*



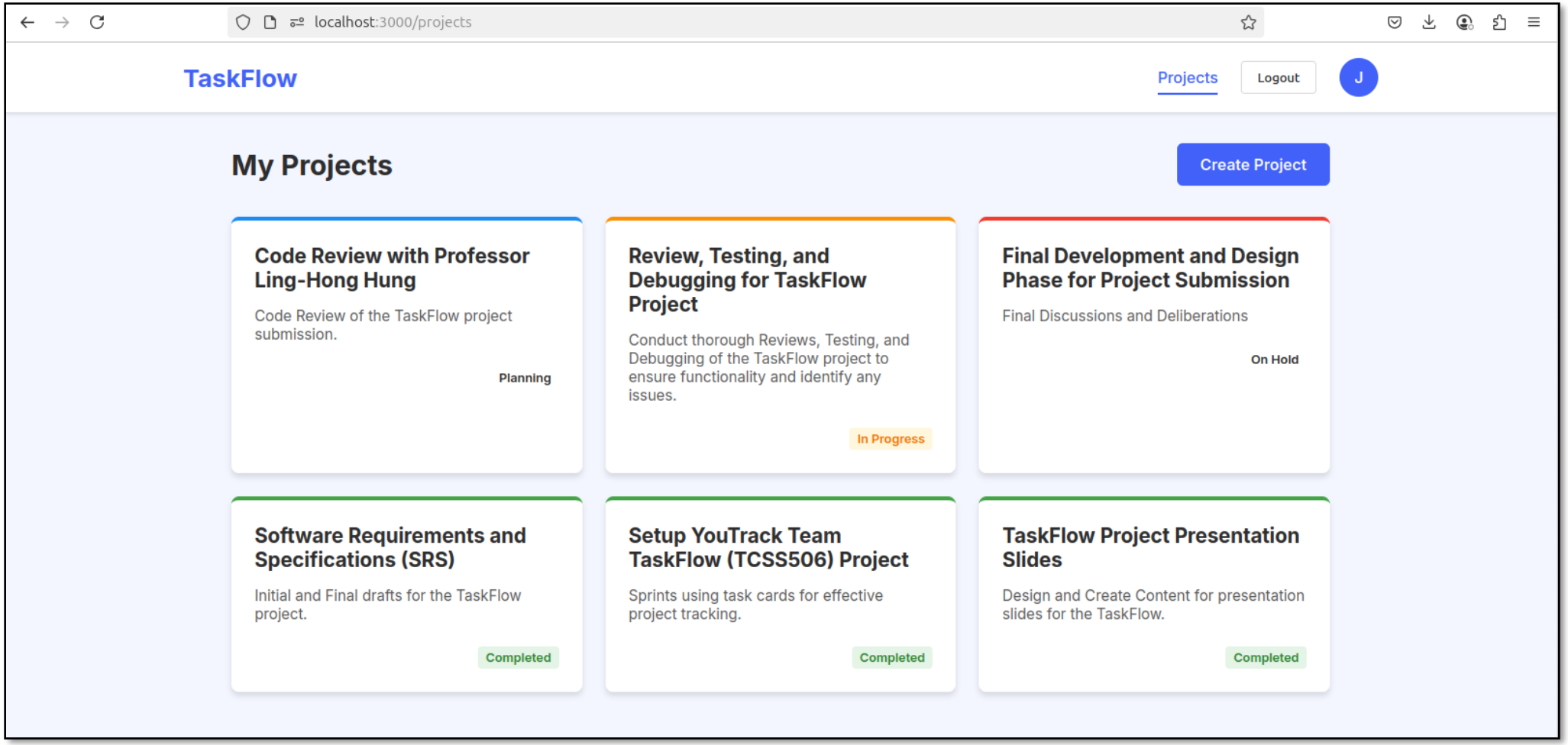
Team **TaskFlow** *Projects*



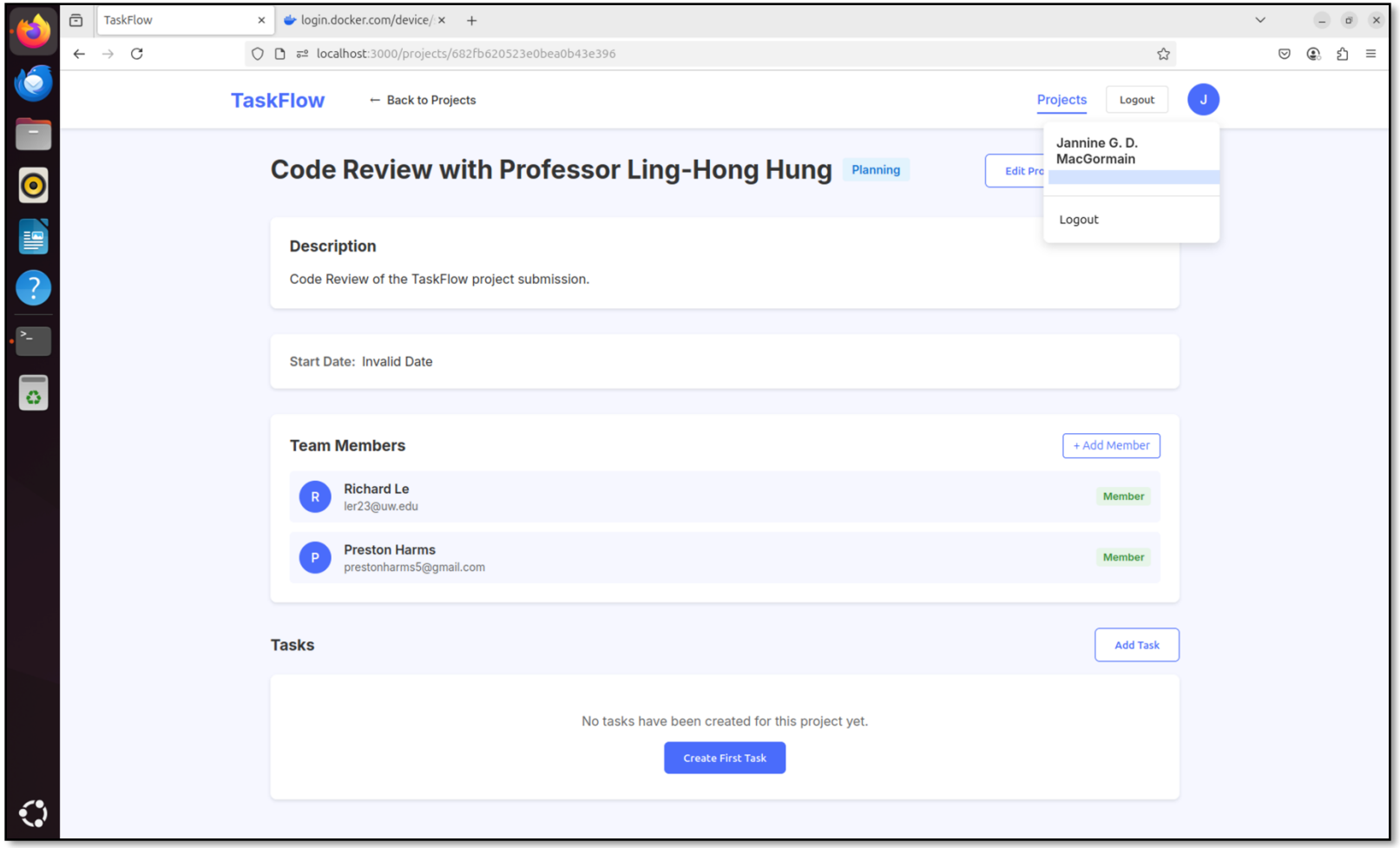
Team **TaskFlow** *Projects*



Team **TaskFlow** *Projects*

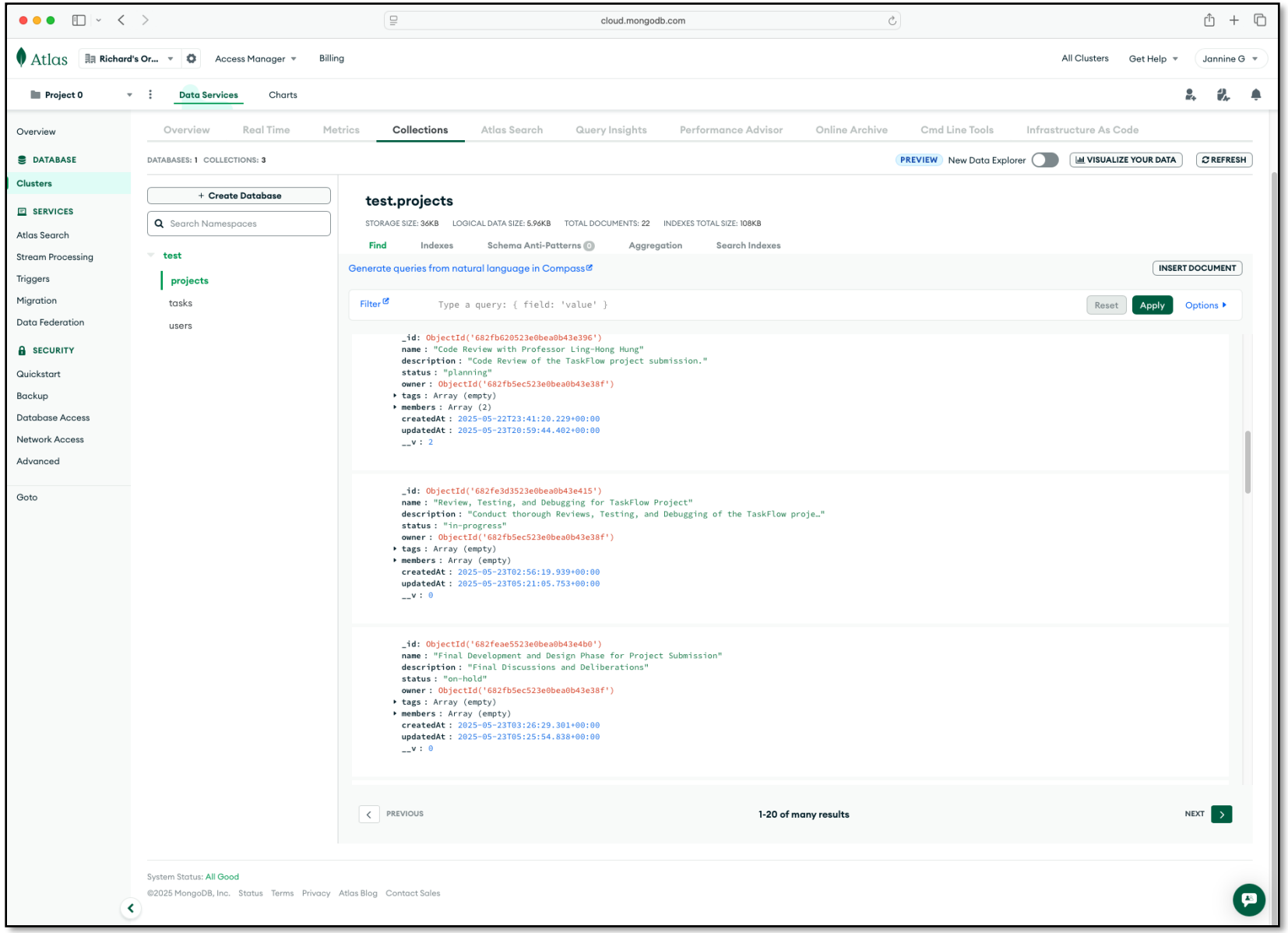


Team **TaskFlow** *Projects*



Team
TaskFlow
Projects

DATABASE



TaskFlow *(Practice Presentation)*

https://www.youtube.com/watch?v=tllhXl7f_6I

University of Washington Tacoma
School of Engineering and Technology
24-25 Graduate Certificate in Software Development Engineering program (GC-SDE)

TCSS 506 A Sp 25:

Practical Full Stack Development

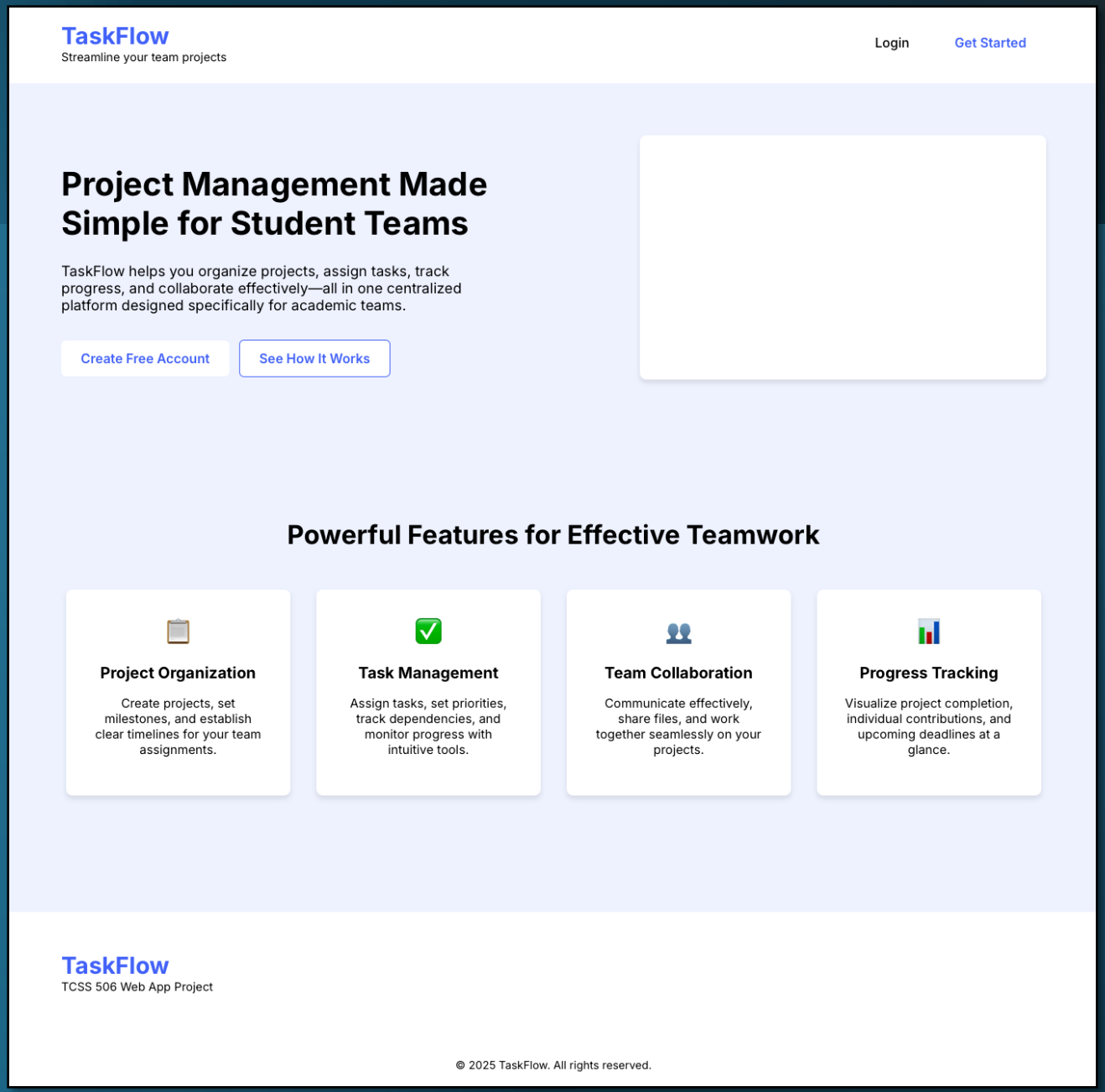
Preston Harms

AWS Project Manager

Richard Le

Git and GitHub Version Control Project Manager

Jannine G. D. MacGormain

YouTrack Project Manager

TaskFlow: A Project Management Platform