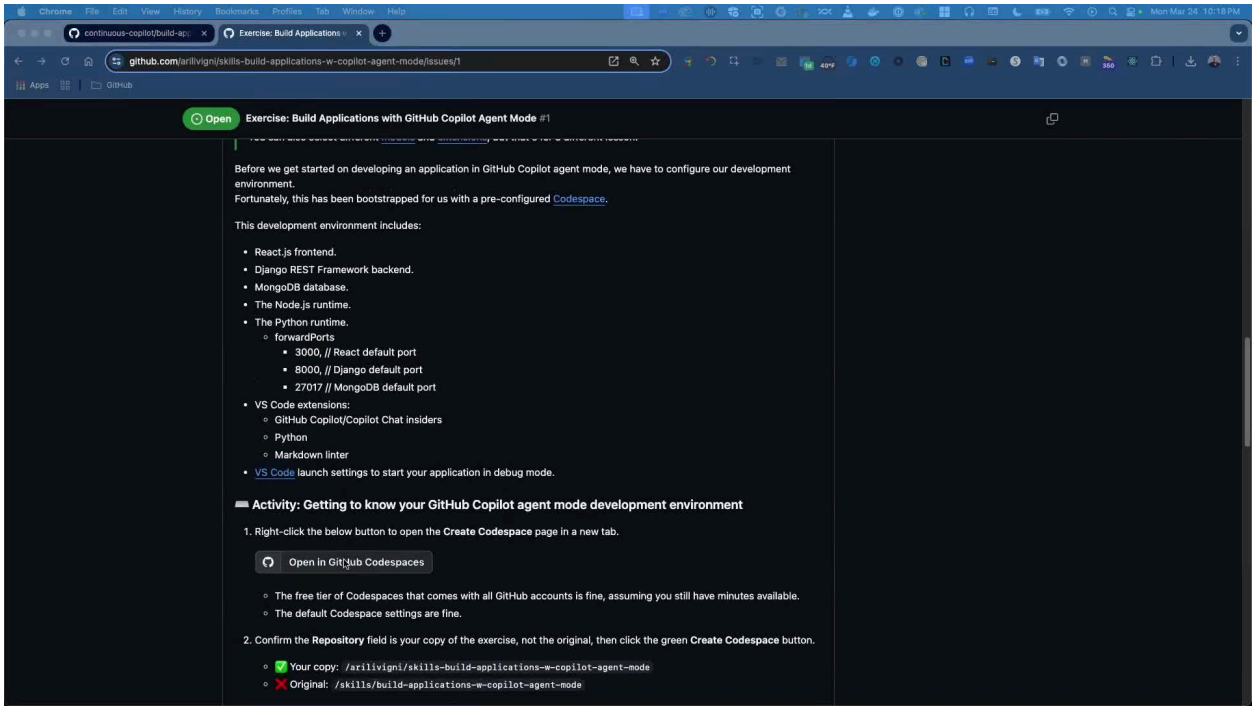
AI Skills Fest - Train-the-Trainer - Build Applications with GitHub Copilot Agent Mode

Step 1: Create a Repository 0:05

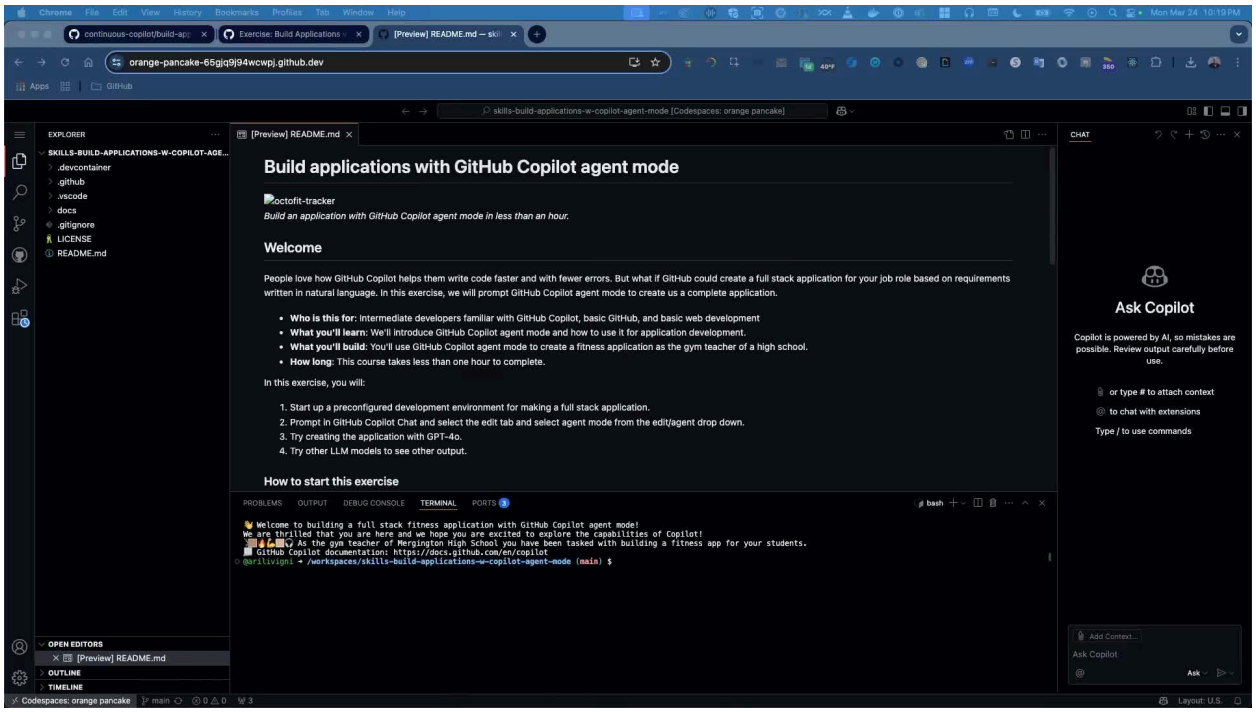
- Welcome to the tutorial on building applications with Co-Pilot Agent Mode.
- Copy the exercise to your GitHub handle or organization.
- Add a note to the description and create the repository.

Step 2: Open in GitHub Code Spaces 0:56



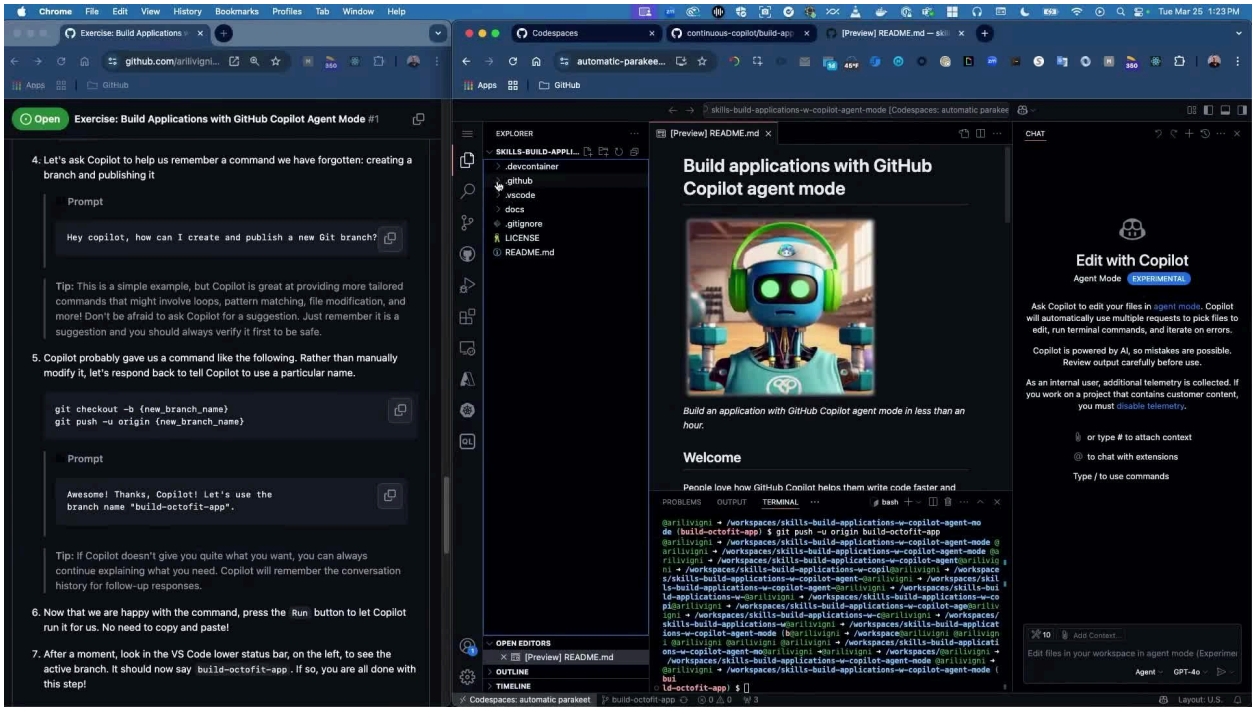
- Right-click to open in GitHub Code Spaces and create a new code space.
- This environment includes necessary extensions like GitHub Copilot and Copilot Chat.

Step 3: Create a Branch 1:27



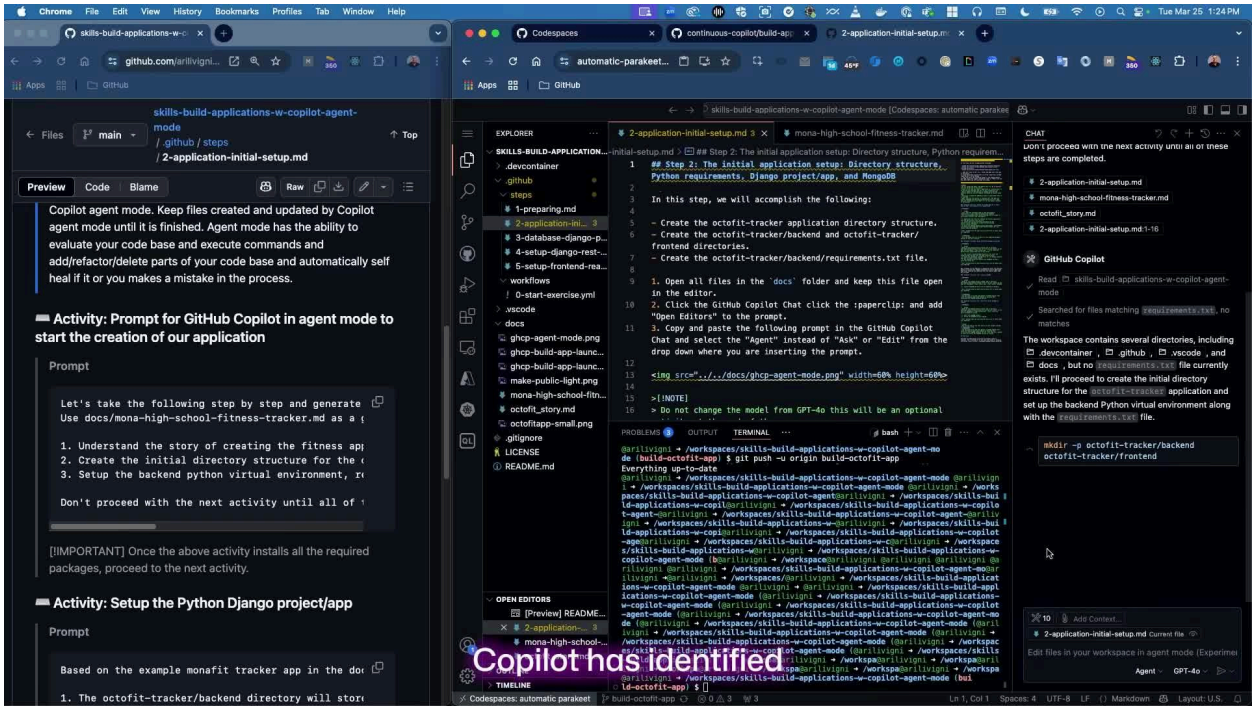
- Copy the first prompt and use Command-I (or Control-I on Windows) to ask Copilot for a command to create a branch.
- Create a specific branch called 'BuildOctoFitApp'.

Step 4: Prepare Files 2:14



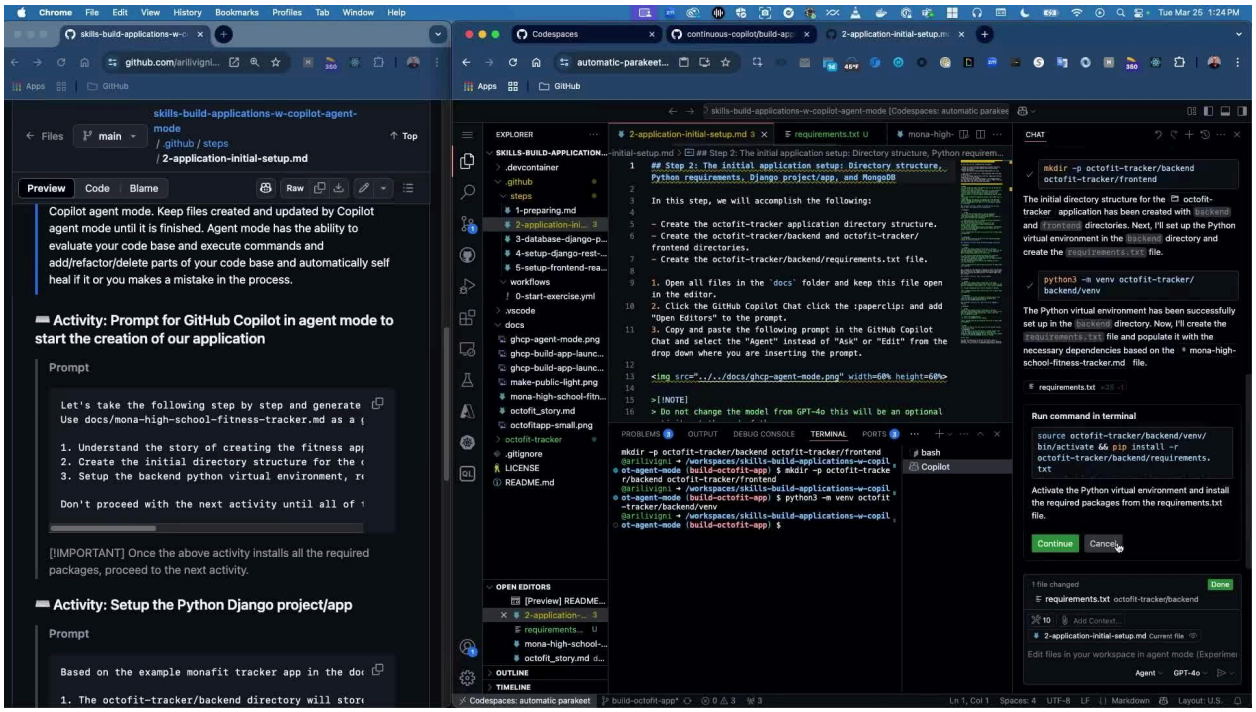
- Open the README file with information about the reference app.
- Ensure Copilot Chat Agent Mode can use this as a reference.

Step 5: Set Up Application Structure 3:01



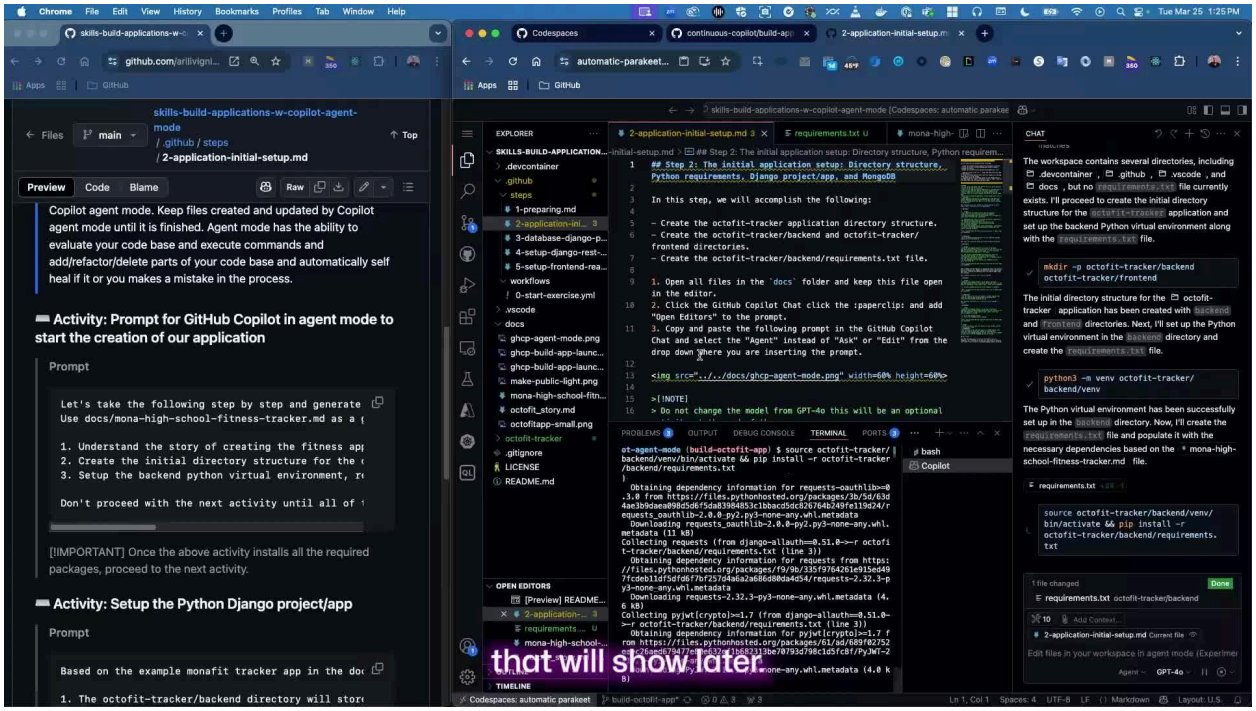
- Copilot identifies the need to create the structure for the OctoFit tracker app.
- Create a virtual environment for Python development.

Step 6: Create Requirements File 3:26



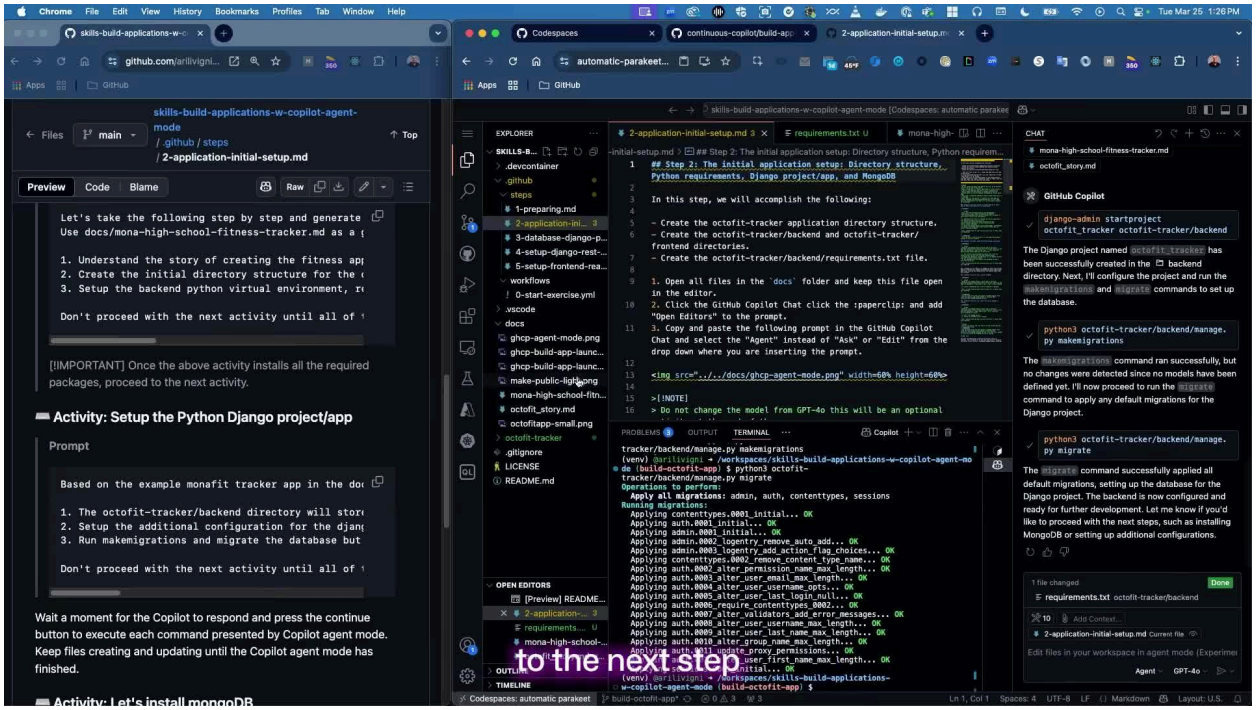
- Generate a requirements.txt file with necessary modules and libraries for the app.
- Install the required packages.

Step 7: Set Up Django Backend 4:33



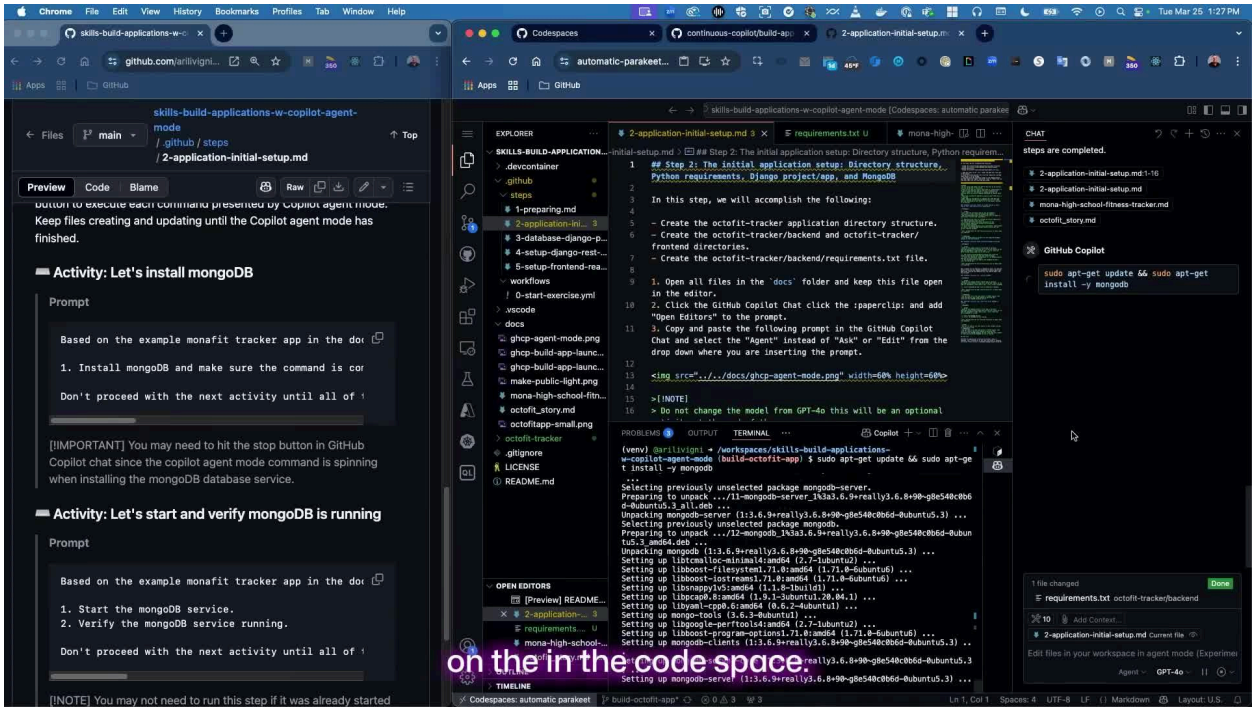
- Ask Copilot to create the Django backend for the project.
- Run system commands related to Django to migrate the database.

Step 8: Install MongoDB 5:08



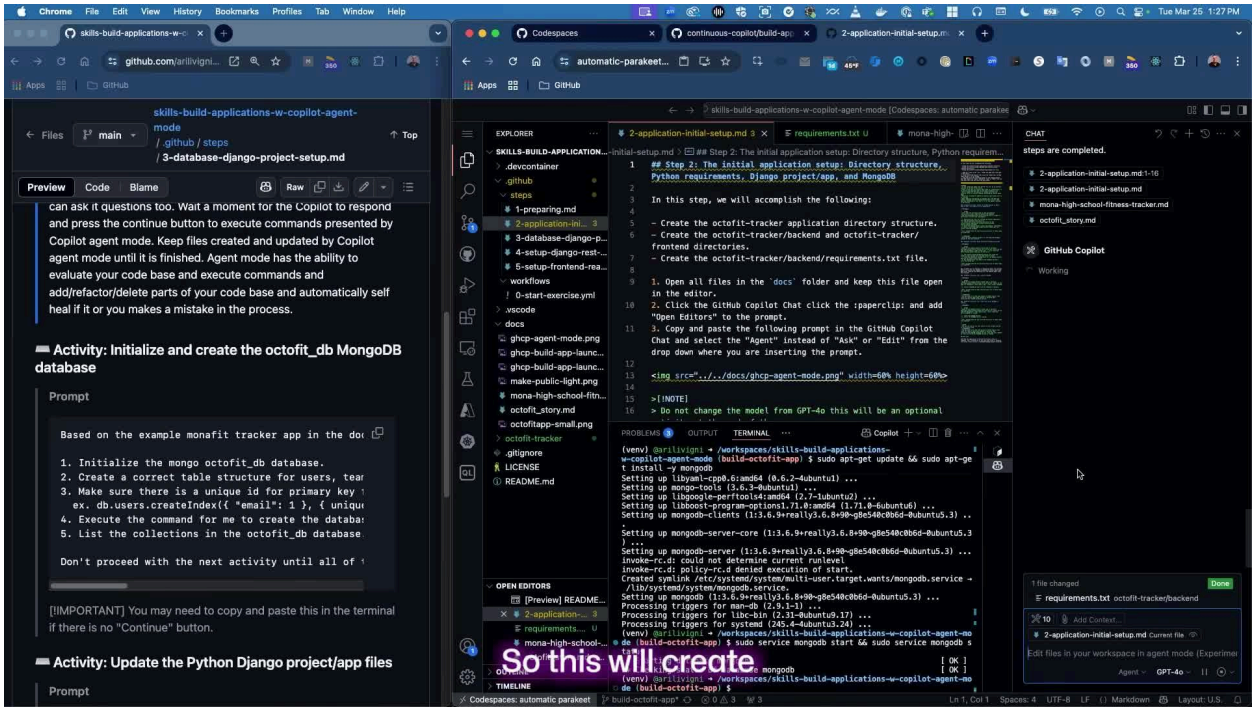
- Install MongoDB using Copilot's reference command.

Step 9: Start MongoDB Service 5:46



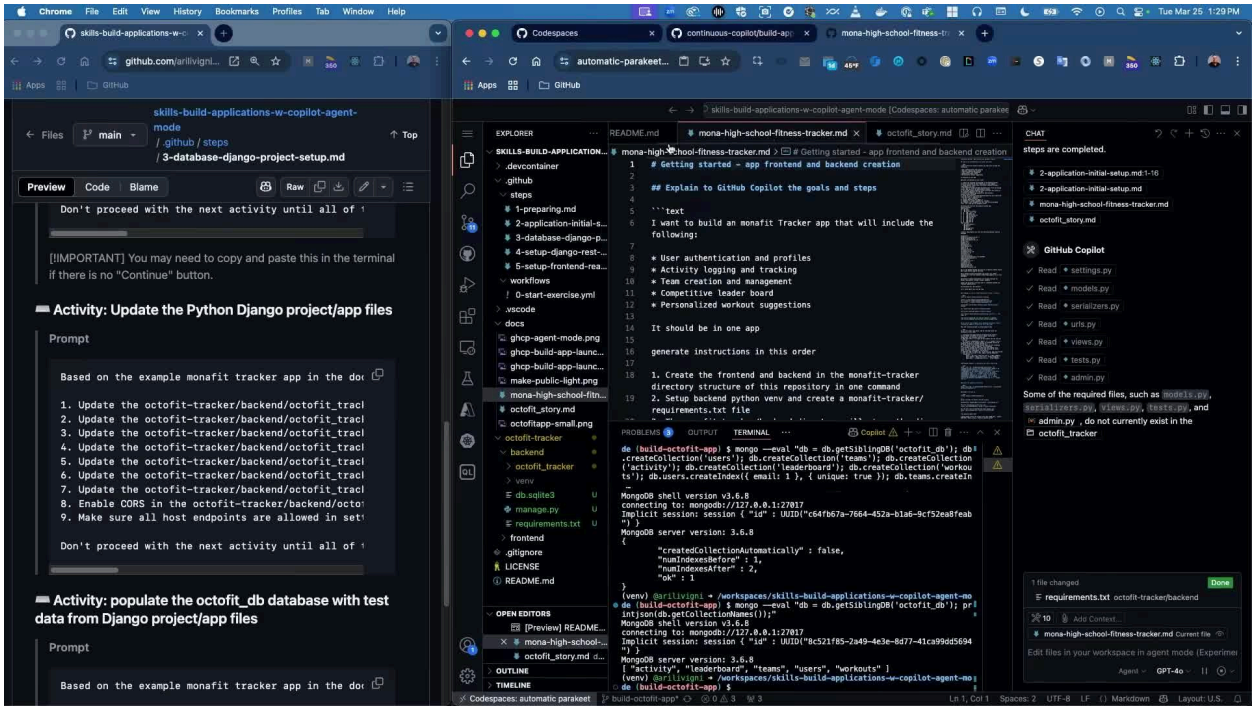
- Ensure the MongoDB service is running.

Step 10: Create Database Structure 6:01



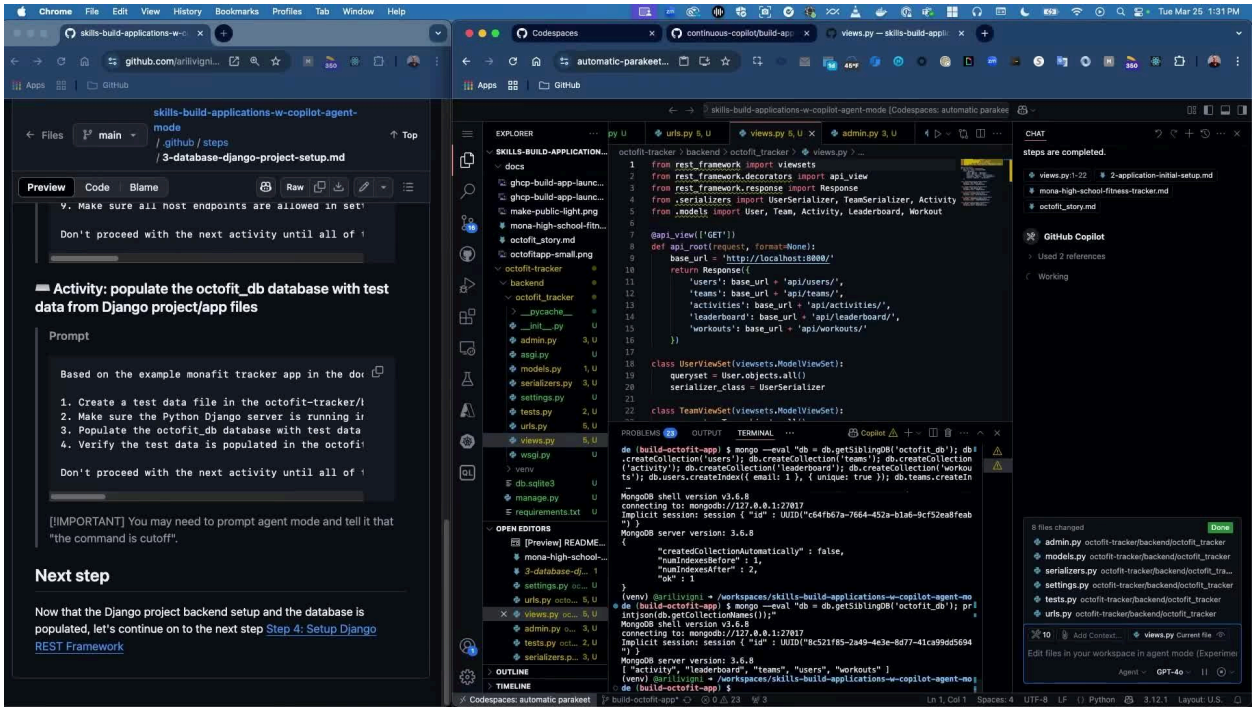
- Set up the database structure with collections for leaderboard, activities, users, teams, and workouts.

Step 11: Update Django Files 6:45



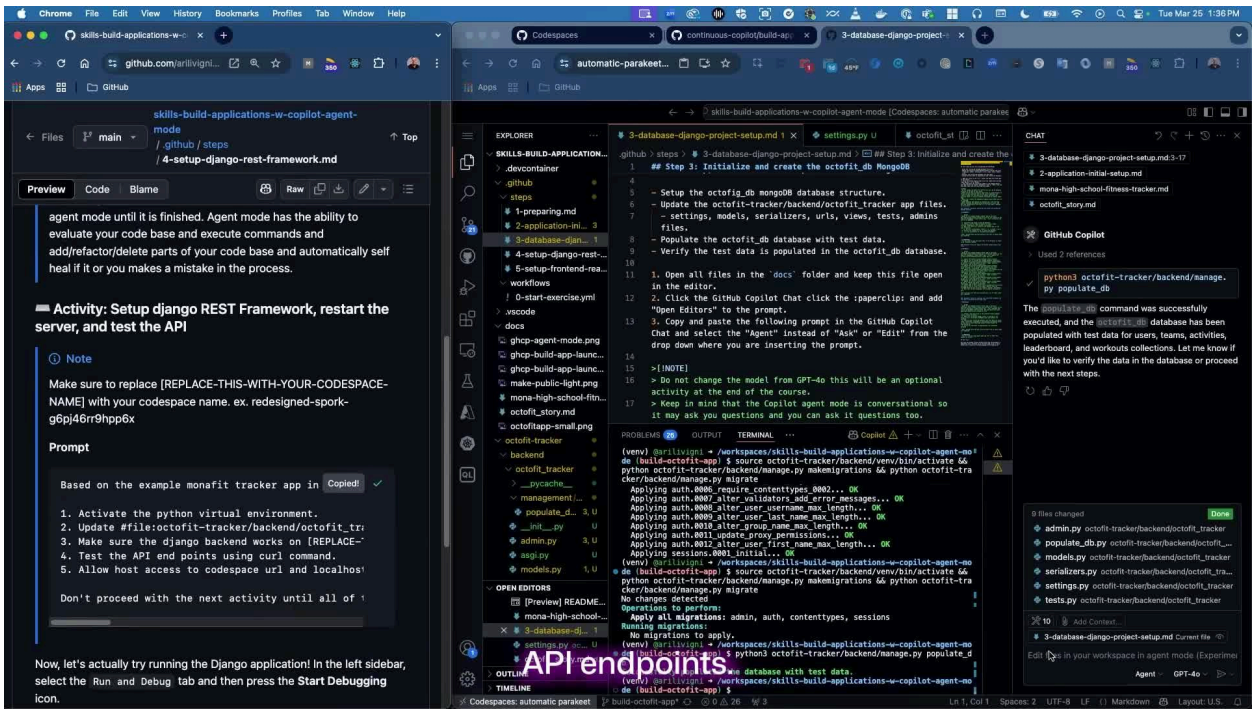
- Update Django app files including views, serializers, URLs, and models.

Step 12: Create Admin and Test Files 7:30



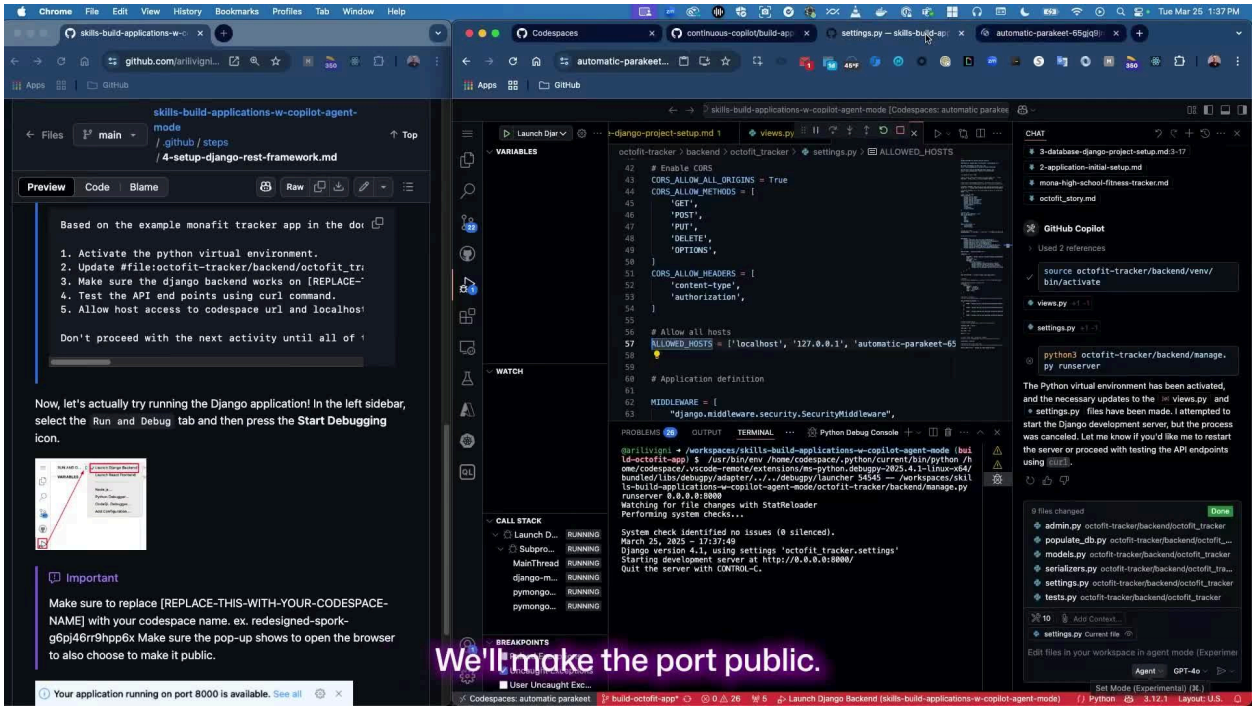
- Copilot creates admin.py and test.py files for the Django project.

Step 13: Write Database Population Script 8:03



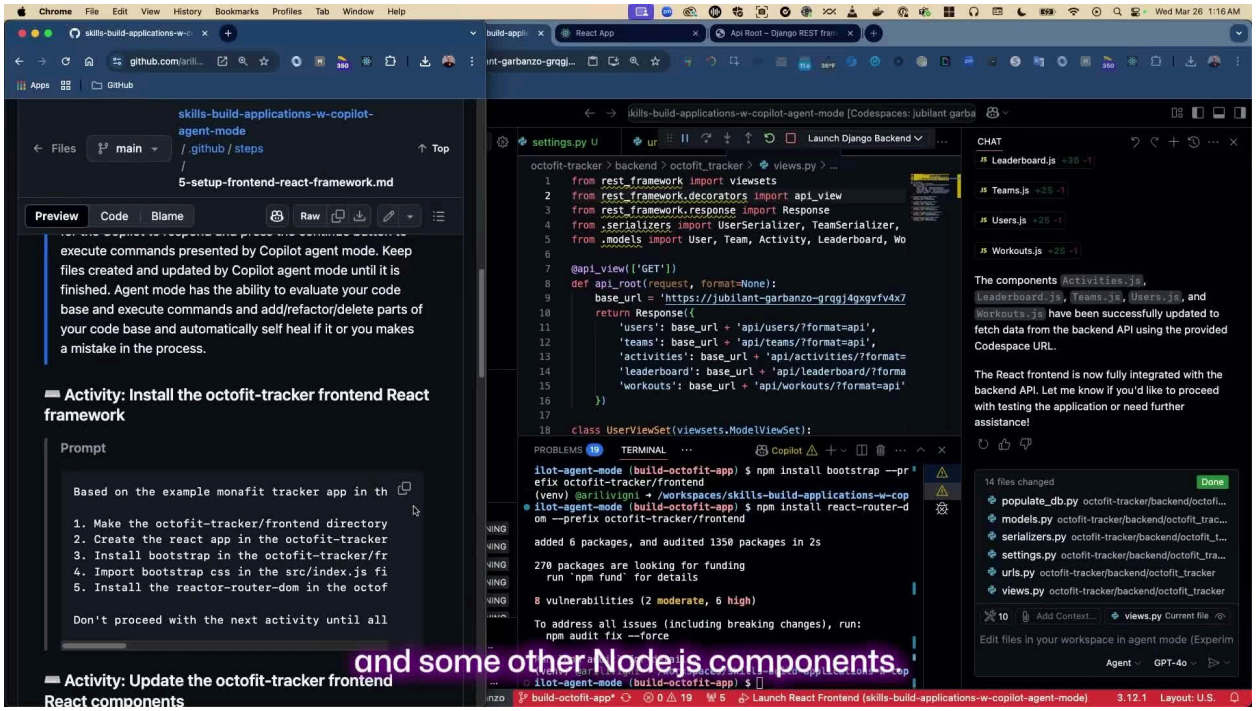
- Write a script called populate_db.py to populate the database.

Step 14: Create REST API Endpoints 9:53



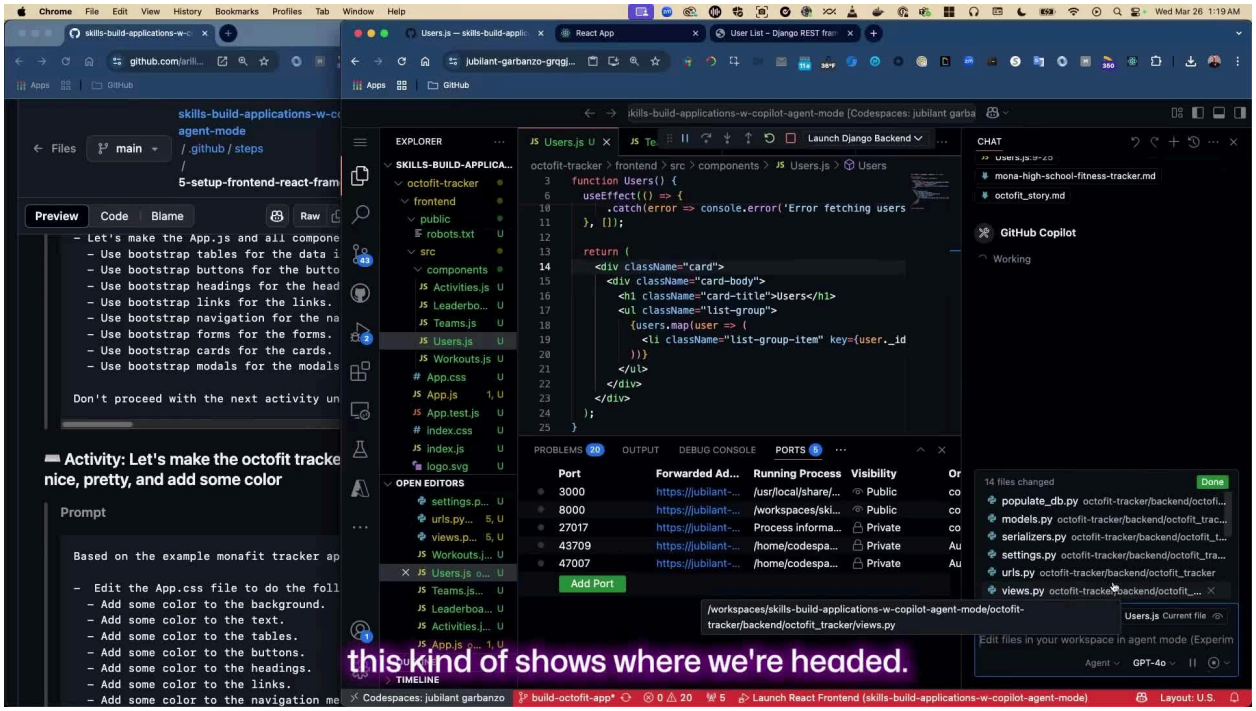
- Create REST API endpoints using the code space name and update views.py.

Step 15: Make Ports Public 11:30



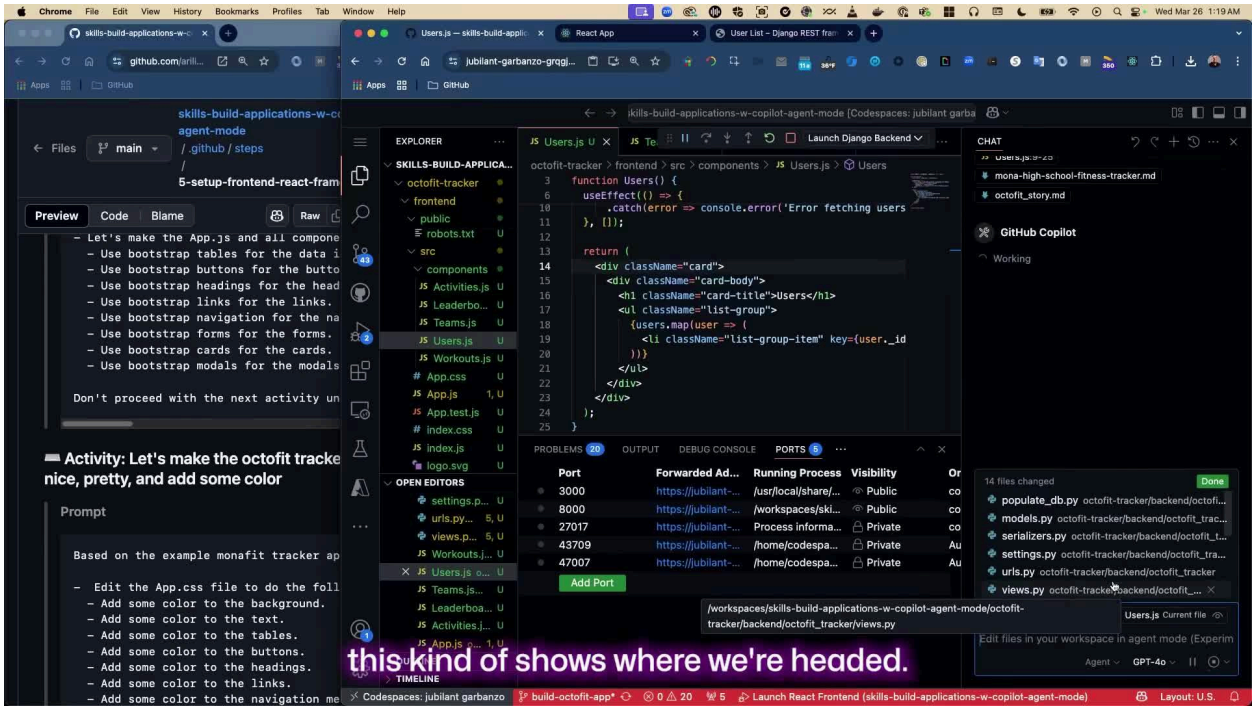
- Make both the React front end (port 3000) and Django back end (port 8000) public.

Step 16: Update Component Files 12:20



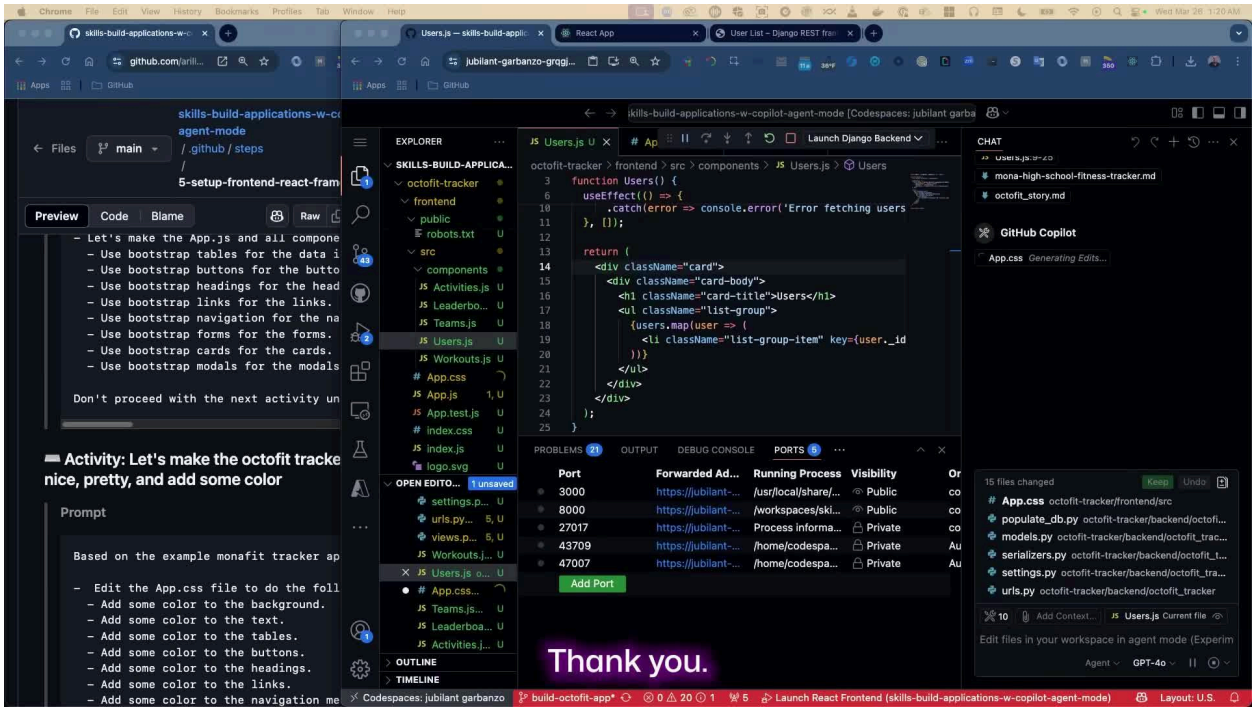
- Update JavaScript files for each component to improve formatting and appearance.

Step 17: Beautify the App 14:38



- Add color and background to enhance the app's visual appeal.

Step 18: Encourage User Interaction 15:02



- Allow users to suggest changes or features for the app in future exercises.

Link to Loom

https://www.loom.com/share/e5faf03cbc164cf6b0020dae469e2847