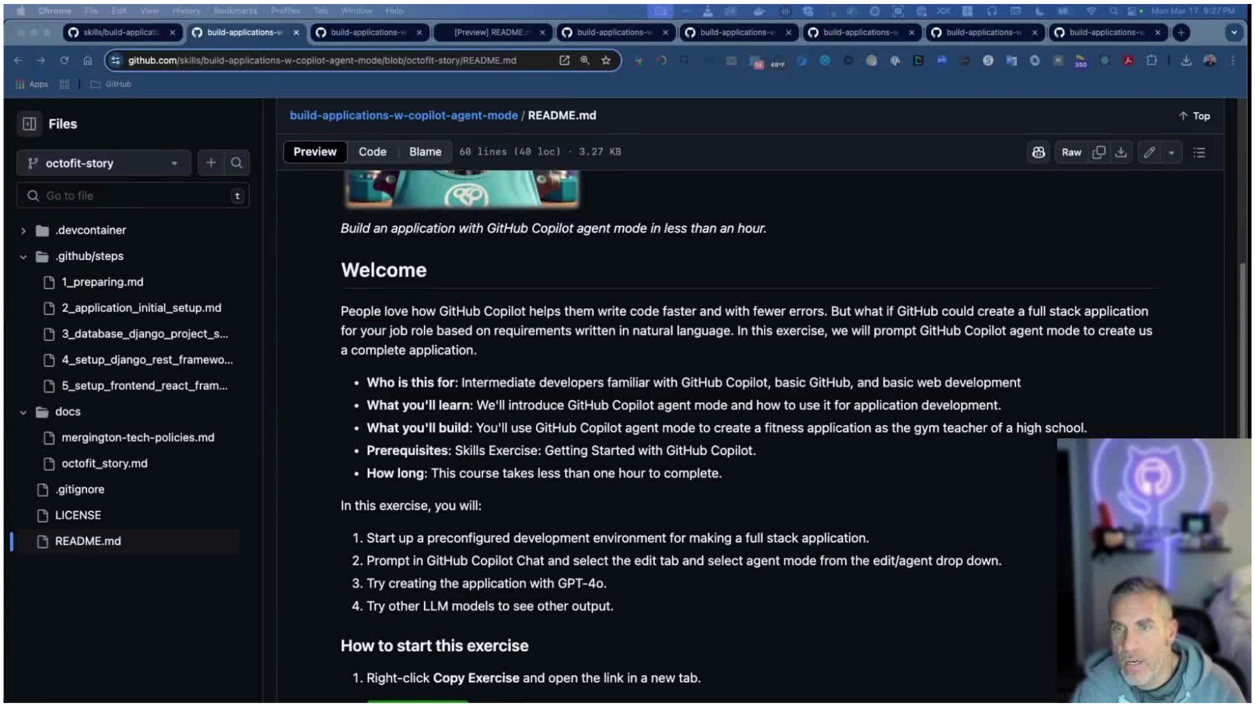
SOP: Train-the-trainer :: Building Applications with GitHub Copilot Agent Mode

Objective

To provide a clear and structured process for team members to effectively build applications using GitHub Copilot Agent Mode, ensuring minimal misunderstandings and maximizing productivity.

Key Steps

1. Create a New Repository 1:24


github.com/skills/build-applications-w-copilot-agent-mode/blob/octofit-story/README.md
Files
octofit-story
Go to file
.devcontainer
.github/steps
1_preparing.md
2_application_initial_setup.md
3_database_django_project_s...
4_setup_django_rest_framewo...
5_setup_frontend_react_fram...
docs
mergington-tech-policies.md
octofit_story.md
.gitignore
LICENSE
README.md
build-applications-w-copilot-agent-mode / README.md
Top
Preview
Code
Blame
60 lines (40 loc) · 3.27 KB
Raw

Build an application with GitHub Copilot agent mode in less than an hour.

Welcome

People love how GitHub Copilot helps them write code faster and with fewer errors. But what if GitHub could create a full stack application for your job role based on requirements written in natural language. In this exercise, we will prompt GitHub Copilot agent mode to create us a complete application.

- **Who is this for:** Intermediate developers familiar with GitHub Copilot, basic GitHub, and basic web development
- **What you'll learn:** We'll introduce GitHub Copilot agent mode and how to use it for application development.
- **What you'll build:** You'll use GitHub Copilot agent mode to create a fitness application as the gym teacher of a high school.
- **Prerequisites:** Skills Exercise: Getting Started with GitHub Copilot.
- **How long:** This course takes less than one hour to complete.

In this exercise, you will:

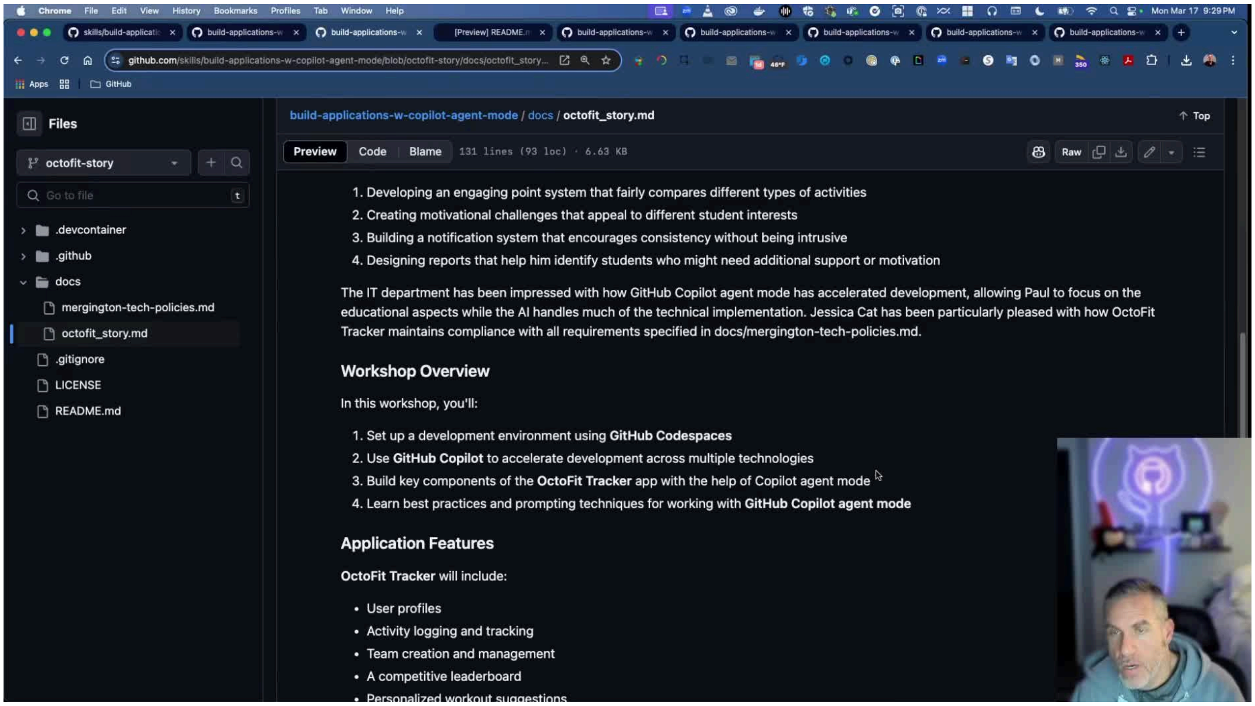
1. Start up a preconfigured development environment for making a full stack application.
2. Prompt in GitHub Copilot Chat and select the edit tab and select agent mode from the edit/agent drop down.
3. Try creating the application with GPT-4o.
4. Try other LLM models to see other output.

How to start this exercise

1. Right-click **Copy Exercise** and open the link in a new tab.

- Right-click and select 'Open a new tab' to create a repository under your username or organization.
- Provide a description for the repository to start developing.

2. Set Up Development Environment 3:33


github.com/skills/build-applications-w-copilot-agent-mode/blob/octofit-story/docs/octofit_story...
Files
octofit-story
Go to file
.devcontainer
.github
docs
mergington-tech-policies.md
octofit_story.md
.gitignore
LICENSE
README.md
build-applications-w-copilot-agent-mode / docs / octofit_story.md
Top
Preview
Code
Blame
131 lines (93 loc) · 6.63 KB
Raw

1. Developing an engaging point system that fairly compares different types of activities
2. Creating motivational challenges that appeal to different student interests
3. Building a notification system that encourages consistency without being intrusive
4. Designing reports that help him identify students who might need additional support or motivation

The IT department has been impressed with how GitHub Copilot agent mode has accelerated development, allowing Paul to focus on the educational aspects while the AI handles much of the technical implementation. Jessica Cat has been particularly pleased with how OctoFit Tracker maintains compliance with all requirements specified in docs/mergington-tech-policies.md.

Workshop Overview

In this workshop, you'll:

1. Set up a development environment using **GitHub Codespaces**
2. Use **GitHub Copilot** to accelerate development across multiple technologies
3. Build key components of the **OctoFit Tracker** app with the help of Copilot agent mode
4. Learn best practices and prompting techniques for working with **GitHub Copilot agent mode**

Application Features

OctoFit Tracker will include:

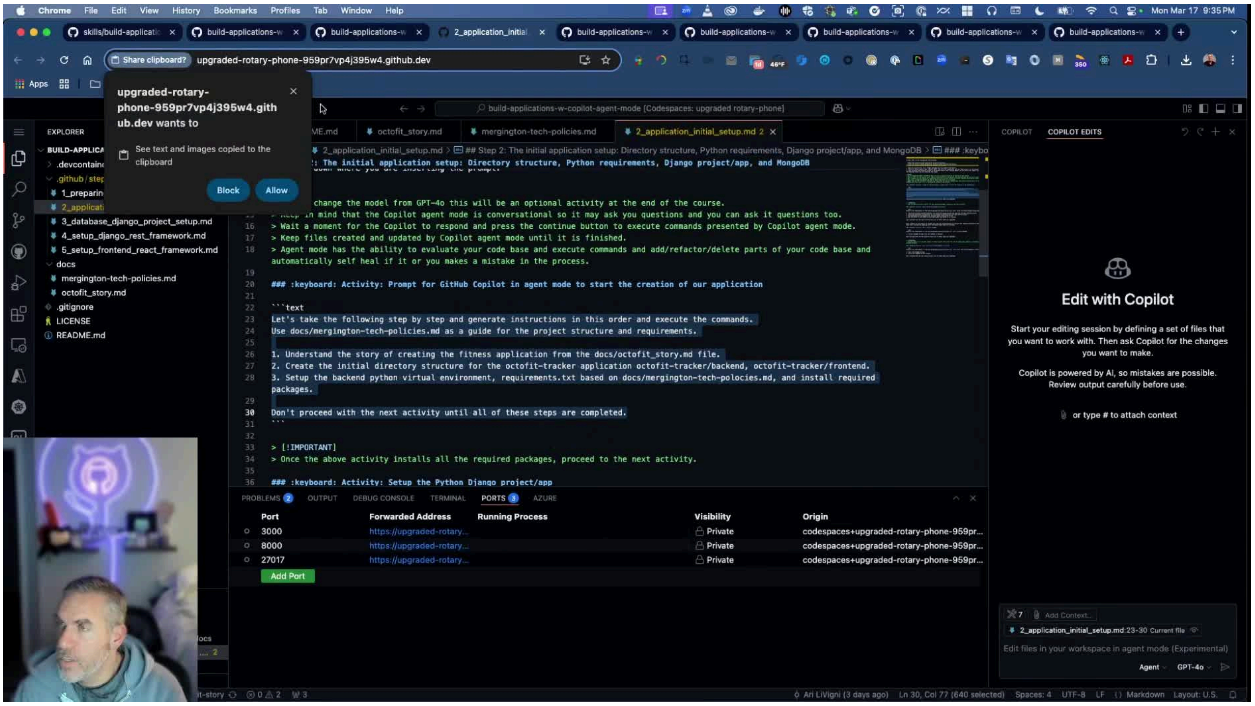
- User profiles
- Activity logging and tracking
- Team creation and management
- A competitive leaderboard
- Personalized workout suggestions

- Open GitHub Codespaces to create a full development environment with VS Code.
- Ensure all necessary repositories are checked out and ports are opened.

3. Prepare Application Structure 7:00

- Use prompts in GitHub Copilot to set up the application directory structure, Python requirements, Django project app, and MongoDB.
- Open relevant files and keep them accessible for reference.

4. Add Context for GitHub Copilot 8:19



upgraded-rotary-phone-959pr7vp4j395w4.github.dev wants to

See text and images copied to the clipboard

Block Allow

~~~
: The initial application setup: Directory structure, Python requirements, Django project/app, and MongoDB

change the model from GPT-4o this will be an optional activity at the end of the course.
> Keep in mind that the Copilot agent mode is conversational so it may ask you questions and you can ask it questions too.
> Wait a moment for the Copilot to respond and press the continue button to execute commands presented by Copilot agent mode.
> Keep files created and updated by Copilot agent mode until it is finished.
> Agent mode has the ability to evaluate your code base and execute commands and add/refactor/delete parts of your code base and automatically self heal if it or you makes a mistake in the process.

### :keyboard: Activity: Prompt for GitHub Copilot in agent mode to start the creation of our application

```text
Let's take the following step by step and generate instructions in this order and execute the commands.
Use docs/mergington-tech-policies.md as a guide for the project structure and requirements.

1. Understand the story of creating the fitness application from the docs/octofit_story.md file.
2. Create the initial directory structure for the octofit-tracker application octofit-tracker/backend, octofit-tracker/frontend.
3. Setup the backend python virtual environment, requirements.txt based on docs/mergington-tech-polocies.md, and install required packages.

Don't proceed with the next activity until all of these steps are completed.
```

> [!IMPORTANT]
> Once the above activity installs all the required packages, proceed to the next activity.

### :keyboard: Activity: Setup the Python Django project/app
~~~

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS 3 AZURE

Port	Forwarded Address	Running Process	Visibility	Origin
3000	https://upgraded-rotary...		Private	codespaces+upgraded-rotary-phone-959pr...
8000	https://upgraded-rotary...		Private	codespaces+upgraded-rotary-phone-959pr...
27017	https://upgraded-rotary...		Private	codespaces+upgraded-rotary-phone-959pr...

Add Port

Edit with Copilot

Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make.

Copilot is powered by AI, so mistakes are possible. Review output carefully before use.

or type # to attach context

Add Context...

2_application_initial_setup.md:23-30 Current file

Edit files in your workspace in agent mode (Experimental)

Agent GPT-4o

Ln 30, Col 77 (640 selected) Spaces: 4 UTF-8 LF Markdown Layout: U.S.

- Select the agent mode in GitHub Copilot and allow it to access the editor context.
- Paste the initial prompt to guide the setup process.

5. Install Python Packages 9:56

- Confirm the installation of necessary Python packages for the backend.
- Proceed to set up the Django project and any additional configurations.

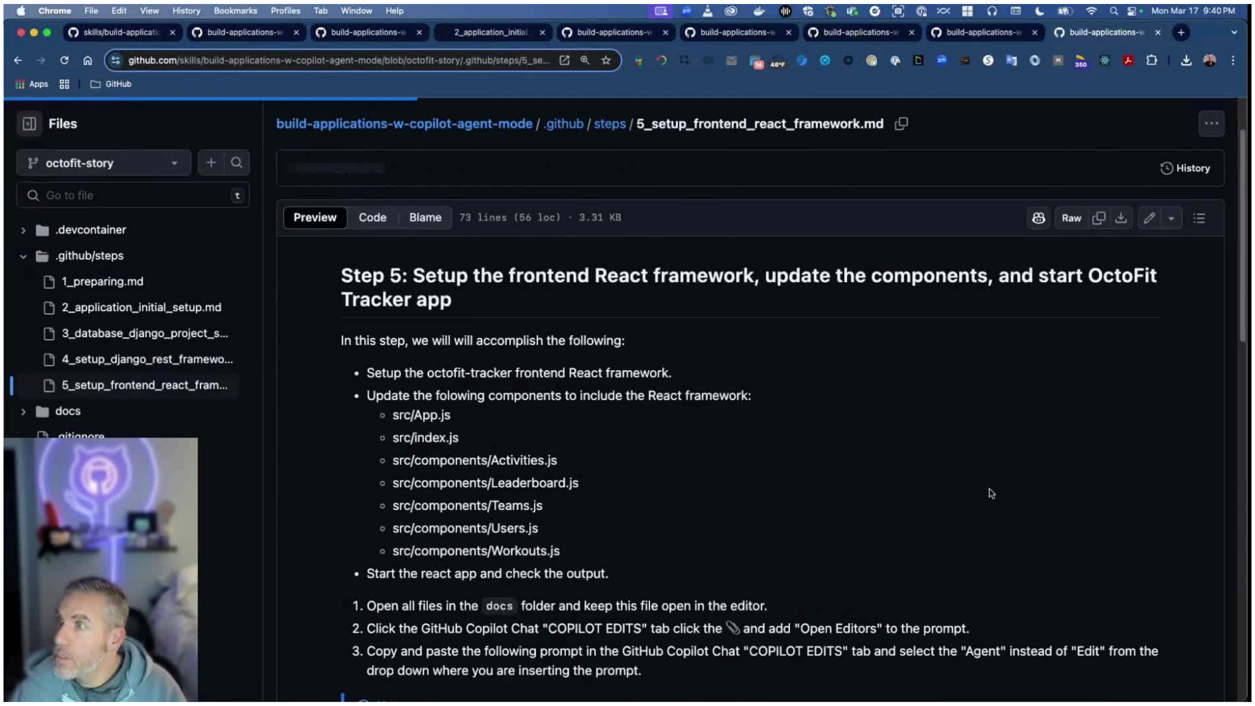
6. Create Database Structure 10:41

- Prompt GitHub Copilot to create the database structure for MongoDB.
- Edit Django files for URLs, views, model serializers, and settings.

7. Populate Database with Test Data 12:05

- Create a Python script to populate the database with test data.
- Ensure the REST API framework is set up correctly for data retrieval.

8. Set Up Frontend Framework 13:12


github.com/skills/build-applications-w-copilot-agent-mode/blob/octofit-story/.github/steps/5_se...
Files
octofit-story
Go to file
.devcontainer
.github/steps
1_preparing.md
2_application_initial_setup.md
3_database_django_project_s...
4_setup_django_rest_framewo...
5_setup_frontend_react_fram...
docs
build-applications-w-copilot-agent-mode / .github / steps / 5_setup_frontend_react_framework.md
History
Preview
Code
Blame
73 lines (56 loc) · 3.31 KB
Raw

Step 5: Setup the frontend React framework, update the components, and start OctoFit Tracker app

In this step, we will will accomplish the following:

- Setup the octofit-tracker frontend React framework.
- Update the folowing components to include the React framework:
  - src/App.js
  - src/index.js
  - src/components/Activities.js
  - src/components/Leaderboard.js
  - src/components/Teams.js
  - src/components/Users.js
  - src/components/Workouts.js
- Start the react app and check the output.

1. Open all files in the `docs` folder and keep this file open in the editor.
2. Click the GitHub Copilot Chat "COPILOT EDITS" tab click the 📎 and add "Open Editors" to the prompt.
3. Copy and paste the following prompt in the GitHub Copilot Chat "COPILOT EDITS" tab and select the "Agent" instead of "Edit" from the drop down where you are inserting the prompt.

- Establish the React framework and create necessary components.
- Update app and index files to ensure data is fetched from the database.

9. Finalize Application 14:07

- Review the application for completeness and functionality.
- Consider running peer reviews and using Copilot for final checks before deployment.

Cautionary Notes

- Ensure that all prompts provided to GitHub Copilot are clear and specific to avoid confusion in the generated code.
- Regularly save your work and check for any errors in the code generated by Copilot.
- Be aware of the dependencies and compatibility of the packages being installed.

Tips for Efficiency

- Familiarize yourself with GitHub Copilot's capabilities to leverage its full potential during development.
- Keep documentation and tech specs readily available for quick reference while working on the application.
- Use version control effectively to track changes and collaborate with team members seamlessly.