F-14 - GitHub Copilot Workspaces (Cloud PR view)

SUMMARY

In this lesson, we covered **GitHub Workspaces** as a part of GitHub Copilot's ecosystem, although it's not a core feature of Copilot. Available to GitHub Copilot Enterprise users, GitHub Workspaces allows developers to seamlessly manage and edit code directly from their browser, eliminating the need to work on local machines. This cloud-based environment mimics VS Code, facilitating code editing, querying, and running.

1. **Access GitHub Workspaces**: It's bundled with GitHub Copilot for Enterprise users.
2. **Edit Code in the Cloud**: You can open code directly in a workspace without pulling it to your local machine. Changes are made in a browser-based virtual environment similar to VS Code.
3. **Branch Management and Commits**: We demonstrated the process of creating a new branch using `git checkout -b` and making changes, staging them, and committing them via the cloud environment, followed by a comparison in a Pull Request (PR).
4. **Codespaces**: By selecting "Codespaces" from the "Code" menu, you create a new virtual environment. This allows for straightforward branch switching, authentication, and code execution—all in a browser interface that mimics VS Code.
5. **Integration with Copilot**: While working in GitHub Workspaces, you can still use Copilot's assistant features, enhancing the cloud-based development experience.

WHAT WE LEARNED

- How to access and utilize **GitHub Workspaces** as a cloud-based development environment.
- Managing branch creations and commit processes in the browser.
- Simplifying code running and testing without needing local setups.
- Using **GitHub Codespaces** to create virtual coding environments.
- Integration of Copilot features within Workspaces.

HOW WE CAN APPLY IT

- **Collaborative Coding**: Team members can work on code directly in the browser, ensuring everyone is working on the same version without local discrepancies.
- **Code Reviews**: Facilitates instant testing and reviewing of code changes, making the review process quicker and more efficient.

- **Remote Development**: Provides access to development environments from any device with a browser, ideal for remote teams.
- **Resource Efficiency**: No need to use local machine resources; everything is handled in the cloud.
- **Quick Prototyping**: Fast setup helps in trying out new ideas without lengthy environment setups.

TIPS AND TRICKS

- Use **branch naming conventions** that are descriptive of the changes to keep work organized and understandable.
- Consider using **GitHub Actions** to automate tasks post-commit, given the easily accessible environment.
- Utilize **GitHub Codespaces** as a way to explore project dependencies and setup without altering your local machine setup.
- Remember to use **Copilot** within Workspaces for suggestions and code completion to enhance productivity.

EXAMPLES

Branch Creation and Push

```
# Create a new branch and switch to it
git checkout -b feature/crypto-list-update

# Make changes, then stage them
git add .

# Commit with a message
git commit -m "Remove Tron from crypto list"

# Push the branch to the repository
git push origin feature/crypto-list-update
```

Creating a Codespace

1. Navigate to the code tab of the desired repository.
2. Click on the **Code** button, select **Codespaces**.
3. Create a new codespace, which opens up a VS Code-like environment in your browser.

Sample Code Change Prompt

```
# Example of Python code that removes an item from a list
crypto_list = ["Bitcoin", "Ethereum", "Cardano", "Tron"]
crypto_list.remove("Tron")
print(crypto_list)
```

Leveraging GitHub Workspaces, we enhance our development process, making it more collaborative and efficient, especially with the seamless integration of Copilot features.