

### **Taipei Day Trip: an e-commerce website for booking one day trip in Taipei**

In the following weeks, we will complete an e-commerce website where we have to build both a front-end interface and a back-end system, based on a RESTful APIs specification. We also have to integrate a third-party payment service to provide online shopping features, including shopping cart, order and payment system.

Besides, we have to use Git and GitHub to do version control and practice a working flow with code review. Finally, we will deploy our website system and database system on a AWS EC2 instance.

Let's start!

#### **Check Layout Design and API Specifications:**

In the future, we will NEVER describe the details which are already included in the following design and API specifications. **You should always take care of details based on these documents.**

#### **Figma Design:**

<https://www.figma.com/design/7R1pj3NopVOMvGCqBoUkBe>

#### **API Specifications:**

<https://app.swaggerhub.com/apis-docs/padax/taipei-a-day/1.1.0>

## **Working Flow: Submit your first task by Git and GitHub**

In stage 2, we will use Git and GitHub to manage the entire project, submit weekly tasks, get reviewed and approved. Refer to this [Git cheat sheet](#) helpful to me. **Complete the following procedure to submit your initial task before the end of Tuesday.**

### **Set Up GitHub Repository:**

#### **1. Local Machine**

Download and unzip our initial project ([here](#)), get a local project folder.

#### **2. Repository**

Create a new **GitHub Repository** in your GitHub account. Follow the guide listed in the empty GitHub Repository page, initialize your **Local Repository** based on the **local project folder**, and push the main branch of your **Local Repository** to your **GitHub Repository**.

#### **3. Collaborator**

Manage Collaborators in the settings tag of your **GitHub Repository**. Invite Chao-Wei Peng ([cwpeng](#)) as your collaborator.

#### **4. Review Rule**

Manage Branches in the settings tag of your **GitHub Repository**. Add branch protection rules for your main branch. Check "require a pull request before merging" and choose "require at least 1 approvals".

### **Set Up Local Repository in Your Machine:**

#### **5. Branch & Checkout**

Open your **Local Repository** with a code editor. Use terminal or any command line tool to create a new develop branch from the main branch, and check out to develop branch.

#### **6. Add & Commit**

Open the app.py file in your project, correct fastapi() to FastAPI() in line 3. Use the "git add" command to add changes to the index, and use the "git commit" command to create a new version in the develop branch of your **Local Repository**.

#### **7. Push**

Use the "git push" command to push the develop branch of your **Local Repository** to the develop branch of your **GitHub Repository**. You can check the latest code version in the web interface of your **GitHub Repository**.

## **On the GitHub Web Page:**

### **8. Pull Request**

Create a Pull Request from the webpage interface of your GitHub Repository where you can merge the develop branch into the main branch. But before merging, invite Chao-Wei Peng (cwpeng) as your code reviewer, [tell me your name in the comment and wait for the review feedback.](#)

### **9. Review & Merge**

If your Pull Request is approved, you can click the "Merge Pull Request" button to complete merging in the pull request management page. (For concepts, the develop branch tracks all development history, and the main branch keeps only the approved version.)

### **10. Fix and Re-Request Review**

If your Pull Request is rejected or requested to make changes, back to your Local Repository, make necessary changes, push the fixed version to the develop branch in your GitHub Repository, go to Pull Request web page, request Chao-Wei Peng (cwpeng) re-review your code, until review approval. [In this working flow, you don't need to create a new Pull Request, just push the latest version to the develop branch and re-request review in the original Pull Request.](#)

**We will continue using the working flow described above to complete all the tasks in stage 2. Read carefully, take care of every word you do not understand, make sure you know what you are doing and why.**