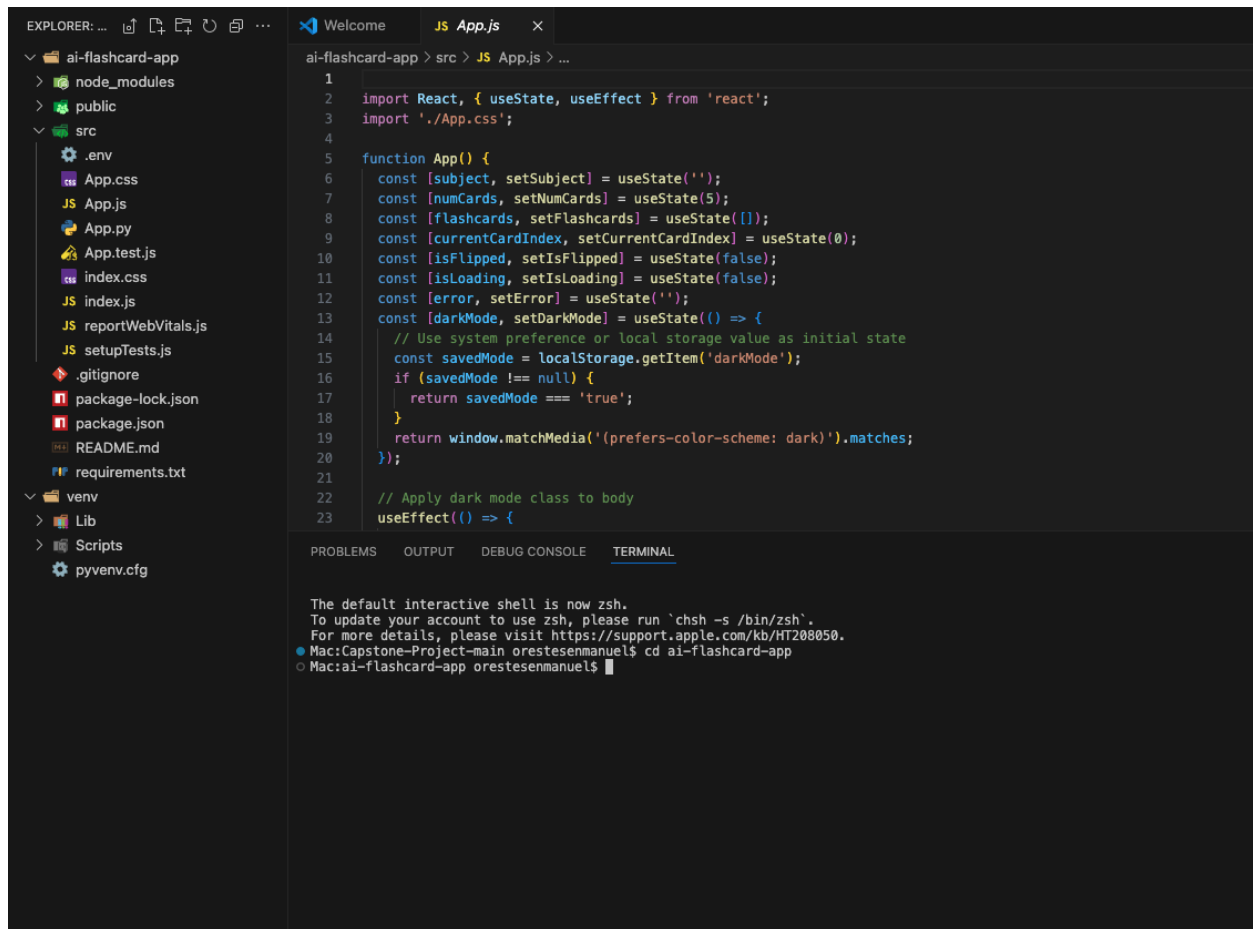


INSTRUCTIONS

Download Ollama from this link before starting: <https://github.com/ollama/ollama>

1. Open project folder (named "Capstone-Project-Main") in vscode after uncompressing it.
2. Create a powershell with the command "control" + "" + "+" .
3. In this first powershell you should input the commands "cd ai-flashcard-app" to enter the directory where the dependencies for the backend are.



The screenshot shows the VS Code interface with the 'ai-flashcard-app' project open. The Explorer sidebar on the left shows the project structure, including 'node_modules', 'public', 'src', and 'venv'. The main editor displays the 'App.js' file, which is a React application using useState and useEffect hooks. The code includes state for subject, numCards, flashcards, currentCardIndex, isFlipped, isLoading, error, and darkMode. It also handles a savedMode from localStorage. The bottom panel shows the 'TERMINAL' tab with a message indicating the default interactive shell is now zsh and providing instructions to update the account to use zsh.

```
ai-flashcard-app > src > JS App.js > ...
1
2 import React, { useState, useEffect } from 'react';
3 import './App.css';
4
5 function App() {
6   const [subject, setSubject] = useState('');
7   const [numCards, setNumCards] = useState(5);
8   const [flashcards, setFlashcards] = useState([]);
9   const [currentCardIndex, setCurrentCardIndex] = useState(0);
10  const [isFlipped, setIsFlipped] = useState(false);
11  const [isLoading, setIsLoading] = useState(false);
12  const [error, setError] = useState('');
13  const [darkMode, setDarkMode] = useState(() => {
14    // Use system preference or local storage value as initial state
15    const savedMode = localStorage.getItem('darkMode');
16    if (savedMode !== null) {
17      return savedMode === 'true';
18    }
19    return window.matchMedia('(prefers-color-scheme: dark)').matches;
20  });
21
22  // Apply dark mode class to body
23  useEffect(() => {
```

4. When inside the source code directory of the project you pip install the "requirements.txt" file with the command "pip install -r requirements.txt" to download all of the dependencies.



The screenshot shows a terminal window with the command 'pip install -r requirements.txt' executed. The output lists various dependencies and their versions, including flask, python-dotenv, opencv-python, werkzeug, Jinja2, itsdangerous, click, blinker, anyio, distro, httpx, pydantic, tqdm, typing-extensions, idna, sniffio, certifi, MarkupSafe, and pydantic-core. The terminal also shows a notice about a new release of pip being available.

```
Mac:ai-flashcard-app orestesenmanuel$ pip install -r requirements.txt
Requirement already satisfied: flask==2.3.3 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from -r requirements.txt (line 1)) (2.3.3)
Requirement already satisfied: flask-cors==4.0.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from -r requirements.txt (line 2)) (4.0.0)
Requirement already satisfied: python-dotenv==1.0.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from -r requirements.txt (line 3)) (1.0.0)
Requirement already satisfied: opencv-python==4.10.0.84 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from -r requirements.txt (line 4)) (4.10.0.84)
Requirement already satisfied: werkzeug==2.3.7 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from flask==2.3.3->-r requirements.txt (line 1)) (2.3.7)
Requirement already satisfied: Jinja2==3.1.2 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from flask==2.3.3->-r requirements.txt (line 1)) (3.1.2)
Requirement already satisfied: itsdangerous==2.1.2 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from flask==2.3.3->-r requirements.txt (line 1)) (2.2.0)
Requirement already satisfied: click==8.1.3 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from flask==2.3.3->-r requirements.txt (line 1)) (8.2.1)
Requirement already satisfied: blinker==1.6.2 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from flask==2.3.3->-r requirements.txt (line 1)) (1.9.0)
Requirement already satisfied: anyio==3.5.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (3.7.1)
Requirement already satisfied: distro==2.1.7 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (1.9.0)
Requirement already satisfied: httpx==0.23.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (0.28.1)
Requirement already satisfied: pydantic==2.10.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (2.11.7)
Requirement already satisfied: tqdm==4.67.1 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (4.67.1)
Requirement already satisfied: typing-extensions==4.5.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from opencv-python==4.10.0.84->-r requirements.txt (line 4)) (4.12.2)
Requirement already satisfied: idna==3.5 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from anyio==3.5.0->-r requirements.txt (line 4)) (3.10)
Requirement already satisfied: sniffio==1.1 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from anyio==3.5.0->-r requirements.txt (line 4)) (1.3.1)
Requirement already satisfied: certifi==2025.1.31 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from httpx==0.23.0->-r requirements.txt (line 4)) (2025.1.31)
Requirement already satisfied: MarkupSafe==2.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from Jinja2==3.1.2->-r requirements.txt (line 1)) (3.0.2)
Requirement already satisfied: annotated-types==0.6.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from pydantic==2.10.0->-r requirements.txt (line 4)) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from pydantic==2.10.0->-r requirements.txt (line 4)) (2.33.2)
Requirement already satisfied: typing-inspection==0.4.0 in /Library/Frameworks/Python.framework/Versions/3.13/Lib/python3.13/site-packages (from pydantic==2.10.0->-r requirements.txt (line 4)) (0.4.1)

[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: pip install --upgrade pip
Mac:ai-flashcard-app orestesenmanuel$
```

5. After the dependencies are installed you should input "cd ai-flashcard-app" and "cd src" again to reenter the directory of the source code.

6. Now that you are inside the source code run the command "python App.py" to run the backend of the application (have in mind that the command might slightly change depending on the version of python in your device).

```
Mac:src orestesenmanuel$ python App.py
bash: python: command not found
Mac:src orestesenmanuel$ python3 App.py
Using Ollama model: gemma3
* Serving Flask app 'App'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Using Ollama model: gemma3
* Debugger is active!
* Debugger PIN: 566-069-659
```

7. Now that you have the backend running press "control" + "``" + "+" so open another powershell and get the frontend running.

8. Inside this new powershell you should input "cd ai-flashcard-app" and "cd src" to enter the source code again.

9. When inside the source code enter the command "npm install" to set up the dependencies of the frontend.

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Mac:Capstone-Project-main orestesenmanuel$ cd ai-flashcard-app
Mac:ai-flashcard-app orestesenmanuel$ cd src
Mac:src orestesenmanuel$ npm install

up to date, audited 1343 packages in 2s

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

13 vulnerabilities (3 low, 3 moderate, 6 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

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

Run `npm audit` for details.
Mac:src orestesenmanuel$
```

10. After the dependencies are downloaded you should now enter the command "npm start" to boot up the frontend and run the webapp to which the link will be given to you. If something else is running in port 3000 then it will ask you if you want to host in another port which you have to press 'y' on the keyboard to agree if you get that message.

Compiled successfully!

You can now view **ai-flashcard-app** in the browser.

Local: http://localhost:3001
On Your Network: http://192.168.1.226:3001

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

webpack compiled **successfully**

11. Now open your computer's terminal and run the command "ollama run gemma3" (assuming you downloaded ollama) which will finish setting up the webapp to work properly.

```
[Mac:~ orestesenmanuel$ ollama run gemma3
```

```
[>>> hello
```

```
Hello there! How can I help you today? 🤖
```

```
Do you want to:
```

- * Chat about something?
- * Ask me a question?
- * Play a game?
- * Get some information?

```
>>> █end a message (/? for help)
```

