**Setup Instructions for "AI Job Card Application"**

These instructions guide you through setting up the Flask application environment after cloning the repository.

**Prerequisites:**

1. **Git:** Ensure Git is installed on your system. ([https://git-scm.com/downloads](https://www.google.com/url?sa=E&q=https%3A%2F%2Fgit-scm.com%2Fdownloads))
2. **Python:** Ensure Python 3.8 or newer is installed. ([https://www.python.org/downloads/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fwww.python.org%2Fdownloads%2F)) Verify by running python --version or python3 --version in your terminal.
3. **pip:** Python's package installer (usually comes with Python). Verify by running pip --version or pip3 --version.

**Steps:**

1. **Clone the Repository:**  
   Open your terminal or command prompt and navigate to the directory where you want to place the project. Then, clone the repository using its URL:
2. git clone <your\_repository\_url>

Replace <your\_repository\_url> with the actual HTTPS or SSH URL of your Git repository.

1. **Navigate to Project Directory:**  
   Change into the newly cloned project directory:
2. cd <repository\_directory\_name>

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

Replace <repository\_directory\_name> with the name of the folder created by git clone (usually the repository name).

1. **Create a Virtual Environment (Recommended):**  
   It's highly recommended to use a virtual environment to isolate project dependencies.
2. # On macOS/Linux
3. python3 -m venv venv
4. # On Windows
5. python -m venv venv

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

This creates a venv folder in your project directory.

1. **Activate the Virtual Environment:**  
   You need to activate the environment before installing packages.
2. # On macOS/Linux (bash/zsh)
3. source venv/bin/activate
4. # On Windows (Command Prompt)
5. .\venv\Scripts\activate
6. # On Windows (PowerShell)
7. .\venv\Scripts\Activate.ps1

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

Your terminal prompt should now indicate the active environment (e.g., (venv)).

1. **Install Dependencies:**  
   Install all the required Python packages listed in requirements.txt:
2. pip install -r requirements.txt

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

1. **Configure Environment Variables:**  
   This application requires certain configuration values set as environment variables for security and proper functioning.
   * **SECRET\_KEY:** Essential for Flask session security. Generate a strong random key.
   * **OCR\_SPACE\_API\_KEY:** Your API key for the OCR Space service (if you want image OCR to work). The code has a placeholder, but it's best to use your own key.

**How to Set Environment Variables:**

* + **Method A: .env File (Recommended for Development):**  
    a. Create a file named .env in the root of the project directory (<repository\_directory\_name>/.env).  
    b. Add the following lines to the .env file, replacing the placeholder values:  
    dotenv # .env file SECRET\_KEY='a\_very\_long\_and\_random\_secret\_key\_here\_32\_chars\_or\_more' OCR\_SPACE\_API\_KEY='YOUR\_ACTUAL\_OCR\_SPACE\_KEY'   
    *(You'll need to install python-dotenv: pip install python-dotenv. The provided app.py might need a line like from dotenv import load\_dotenv; load\_dotenv() added near the top to load this automatically)* ***Note:*** *The current app.py doesn't explicitly load .env, it relies on OS variables. Adding dotenv is a common pattern.*

c. **Crucially:** Add .env to your .gitignore file to avoid committing secrets!

* + **Method B: Operating System Environment Variables:**  
    a. **macOS/Linux:** Set variables in your terminal session (they only last for that session) or add them to your shell profile (.bashrc, .zshrc, .profile):  
    bash export SECRET\_KEY='a\_very\_long\_and\_random\_secret\_key\_here\_32\_chars\_or\_more' export OCR\_SPACE\_API\_KEY='YOUR\_ACTUAL\_OCR\_SPACE\_KEY'   
    b. **Windows (Command Prompt):**  
    cmd set SECRET\_KEY="a\_very\_long\_and\_random\_secret\_key\_here\_32\_chars\_or\_more" set OCR\_SPACE\_API\_KEY="YOUR\_ACTUAL\_OCR\_SPACE\_KEY"   
    c. **Windows (PowerShell):**  
    powershell $env:SECRET\_KEY = 'a\_very\_long\_and\_random\_secret\_key\_here\_32\_chars\_or\_more' $env:OCR\_SPACE\_API\_KEY = 'YOUR\_ACTUAL\_OCR\_SPACE\_KEY'   
    d. For persistent setting on Windows, search for "Environment Variables" in the system settings.

1. **Run the Flask Application:**  
   With the virtual environment still active and environment variables set, run the application:
2. python app.py

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

*(If you use python3 normally, use python3 app.py)*

You should see output indicating the Flask development server is running, typically on http://127.0.0.1:5000/ or http://0.0.0.0:5000/.

1. **Access the Application:**  
   Open your web browser and navigate to the URL shown in the terminal output (e.g., http://127.0.0.1:5000/). You should see the landing page.
   * Access **Admin Login** at http://127.0.0.1:5000/admin
   * Access **User Login** via the button on the landing page or directly at http://127.0.0.1:5000/login

**Troubleshooting:**

* **ModuleNotFoundError:** Make sure your virtual environment is activated ((venv) should be in your prompt) and that you ran pip install -r requirements.txt successfully within it.
* **OCR Errors:** Double-check that your OCR\_SPACE\_API\_KEY environment variable is set correctly and that the key itself is valid and active.
* **Permission Errors (macOS/Linux):** If you encounter permission issues running activate or python, you might need chmod +x.
* **Port Conflicts:** If port 5000 is already in use, Flask might show an error. You can run it on a different port: flask run --port=5001 (requires installing python-dotenv and setting FLASK\_APP=app.py environment variable first) or modify the app.run() call in app.py (e.g., app.run(debug=True, port=5001)).

**.gitignore Recommendation:**

Ensure your repository's .gitignore file includes at least the following to prevent committing unnecessary or sensitive files:

# Virtual Environment

venv/

\*.venv/

env/

\*/env/

\*/venv/

# Environment variables

.env

\*.env.\*

# Python cache files

\_\_pycache\_\_/

\*.py[cod]

\*$py.class

# Temporary files

temp/

# OS generated files

.DS\_Store

Thumbs.db

# IDE specific files

.idea/

.vscode/

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Gitignore

IGNORE\_WHEN\_COPYING\_END