

Set-Up[Downloads]

Visual Studio Code:-

- Open the browser and search for VS Code or click on the following link <https://code.visualstudio.com/>
- According to your OS download the installer.
- After the download is complete open the downloaded file.
- Accept all the legalities [as they are mandatory for successfully downloading the VS Code].
- Keep the download path as default [DO NOT CHANGE IT].
- Select the ‘Create a Desktop icon’ option for easily finding the app.
- Then click on ‘Install’ and wait till the download is complete.

Python v3.11.7:-

- Open the browser and search for Python v3.11.7 or click on the following link <https://www.python.org/downloads/release/python-3117/>
- According to your OS download the installer. [NOTE: Consider your device specification while downloading the installer.]
- After the download is complete open the downloaded file and click on ‘Install Now’.
- After completing the download set the System variables for Python.
- In ‘search’ type ‘System variables’. Then go in environment variables.
- In environment variables add a new path for the downloaded python version. In this new path you need to paste the address of your ‘Python311’ folder.
- Similarly copy the path of ‘Scripts’ folder in the ‘Python311’ folder and add it in the environment variables.
- If you have created environment variables for different python versions then you don’t have to delete those, just select the variables (paths) you just added and click on the ‘Move up’ button to align them above all kinds of python versions. Do this for the Scripts variable as well.

Set-Up[Downloads]

MongoDB:-

- Click on the following link <https://www.mongodb.com/try/download/shell>
- According to your OS download the installer.
- After the download is complete open the downloaded file.
- Keep clicking on ‘Next’ button and Accept all the legalities. In Set-up click on ‘Complete’. [NOTE: Do not forget to check the checkbox saying ‘Install MongoDB Compass’]. Finally click on ‘Install’.
- Doing this your MongoDB will be downloaded and you can access it using MongoDB Compass.



Set-Up[Setting up the downloaded content]

System Variables for Python:-

- Search for "System Variables" in the search bar and open "Edit the system environment variables".
- In the System Properties window, click on the "Advanced" tab, then click on the "Environment Variables" button at the bottom. This will open the Environment Variables window.
- In the Environment Variables window, you'll see two sections: User variables and System variables.
- Under the "System variables" section, look for the variable named "Path" and select it. Then click on the "Edit..." button.
- In the "Edit Environment Variable" window, click on the "New" button to add a new entry.
- For Python path:
 - Enter the path to the Python installation directory. This is typically something like C:\Users\YourUsername\AppData\Local\Programs\Python\Python3.11.
 - Replace "YourUsername" with your actual username.
- For Python Scripts path:
 - Enter the path to the Scripts directory within the Python installation directory. This is typically something like C:\Users\YourUsername\AppData\Local\Programs\Python\Python3.11\Scripts.
 - Again, replace "YourUsername" with your actual username.
- After entering each path, click "OK" to save the changes and close the windows.
- Once you've added both paths, you can verify that they were added correctly by opening a new Command Prompt window (or PowerShell) and typing python --version. This should display the installed Python version, confirming that Python is now accessible from the command line.
- You can also verify the Scripts path by typing pip --version, which should display the installed version of pip (Python package manager).
- By adding the Python and Python Scripts paths to the system variables, you're ensuring that the Python interpreter and its associated tools are accessible from any command prompt window without needing to specify the full path to their executable files. This makes it easier to run Python scripts and manage Python packages using pip.

Set-Up[Downloads]

Downloading the project and set-up:-

1. Download the project files from the provided link (drive):
 - In the upcoming prompt, you will receive a link to download the project files, which will likely be hosted on a cloud storage platform like Google Drive or Dropbox. Click on the provided link to access the project files.
2. Create a folder named "Authentiface" on the desktop:
 - Once you've downloaded the project files, navigate to your desktop on your computer.
 - Right-click on an empty space on your desktop to open the context menu.
 - From the context menu, select "New" and then "Folder" to create a new folder.
 - Rename the folder to "Authentiface" (without the quotation marks) to match the name of the project.
3. Extract all the project files into the "Authentiface" folder:
 - Locate the downloaded project files, which are likely in a compressed (ZIP) format.
 - Right-click on the ZIP file and select "Extract All..." from the context menu.
 - In the "Extract Compressed (Zipped) Folders" window, choose the location where you want to extract the files.
 - Navigate to the "Authentiface" folder you created on the desktop and click "Extract" to extract all the files from the ZIP archive into the folder.
4. Open Visual Studio Code and navigate to the "Authentiface" folder:
 - Launch Visual Studio Code, either from the Start menu or by searching for it in your computer's applications.
 - Once Visual Studio Code is open, go to the "File" menu and select "Open Folder...".
 - Navigate to your desktop and select the "Authentiface" folder that you created earlier.
 - Click "Open" to open the "Authentiface" folder in Visual Studio Code.

Set-Up

Virtual Environment Setup:-

1. Open a new terminal in Visual Studio Code:

- Launch Visual Studio Code if it's not already open.
- In Visual Studio Code, go to the top menu and click on "Terminal" or press Ctrl + ~ to open a new terminal panel at the bottom of the window.
- This terminal panel will serve as your command line interface within Visual Studio Code.

2. Create a virtual environment:

- In the newly opened terminal panel, you'll need to create a virtual environment for your project. A virtual environment is a self-contained directory that contains a Python installation for a particular version of Python, plus a number of additional packages.
- To create a virtual environment, run the following command in the terminal: “python -m venv venv”
- This command tells Python to run the venv module as a script (python -m), which creates a new virtual environment named venv in your current directory. The name venv is commonly used for virtual environments, but you can choose any name you prefer.

3. Activate the virtual environment:

- Once the virtual environment is created, you need to activate it to use it. Activating a virtual environment sets up your terminal to use the Python interpreter and other tools from the virtual environment rather than the global Python installation.
- To activate the virtual environment, you first need to navigate to the "Scripts" folder within the virtual environment directory. You can do this by running the following command in the terminal: “cd venv/Scripts”
- This command changes the current directory (cd) to the "Scripts" folder inside the venv directory.
- After navigating to the "Scripts" folder, you'll use the activate script to activate the virtual environment. Run the following command in the terminal: “.\activate”.
- On Windows, the activation script is named activate, and you execute it by typing .\activate (the .\ specifies the current directory).
- After running this command, you should see (venv) appear at the beginning of your terminal prompt, indicating that the virtual environment is now active.

Set-Up[Downloads]

Installing Python Dependencies:-

1. Install the Required Python Libraries:

- In the activated virtual environment, you can proceed to install the required Python libraries using pip. Run the following command in your terminal: "pip install pymongo passlib face_recognition flask flask_pymongo numpy opencv-python bcrypt tensorflow==2.14.0 keras==2.14.0"
- This command will utilize pip to install the specified Python libraries (pymongo, passlib, face_recognition, flask, flask_pymongo, numpy, opencv-python, bcrypt and tensorflow) within the active virtual environment. Pip will download and install the libraries along with any dependencies they require.

2. Verify Installation:

- After the installation process completes, you can verify that the libraries were installed successfully by running: "pip list".
- This command will display a list of installed Python packages, including the ones you just installed (pymongo, passlib, face_recognition, flask, flask_pymongo, numpy, opencv-python, bcrypt and tensorflow). Look for these package names in the output to confirm that they were installed correctly within the active virtual environment.

Database Setup:-

1. Connect to MongoDB Server:

- In MongoDB Compass, you'll be prompted to connect to a MongoDB server. Since this project doesn't specify any particular MongoDB server configuration, we'll assume you have a MongoDB server running locally on the default port (27017).
- In MongoDB Compass, click on the "Connect" button to initiate the connection process.

2. Create a New Database:

- After successfully connecting to the MongoDB server, you'll see the list of existing databases (if any). Since we're setting up a new database for the project, we'll need to create one.
- Click on the "Create Database" button or navigate to the "Databases" tab and click on the "+ Create Database" button.
- Enter the name "users" for the new database and press Enter to create it.

Set-Up

Database Setup(contd.):-

3. Create Collections:

- Once the "users" database is created, we need to create two collections within it: "log" and "base".
- To create a new collection, navigate to the "Collections" tab within the "users" database.
- Click on the "+ Add My Own Data" button or the "+ Create Collection" button.
- Enter the name "log" for the first collection and click "Create".
- Repeat the same process to create another collection named "base".

4. Verify Collections:

- After creating both collections, you should see them listed under the "Collections" tab within the "users" database in MongoDB Compass.
- Verify that both collections ("log" and "base") are successfully created and are ready to be used for the project.

Database Entries:-

1. Open the run.py File:

- Navigate to the directory where the run.py file is located. In the project setup described earlier, the run.py file is located within the "Authentiface" folder.
- Open the run.py file using a text editor or an integrated development environment (IDE) like Visual Studio Code.

2. Update Image Path:

- Within the run.py file, locate the section where the image path is specified. This is typically where the image file for facial recognition is loaded.
- Update the image path to point to the correct location of the image file on your system.
- If the original code contains a placeholder path like "path/to/image.jpg", replace it with the actual path where your image file is located.
- Ensure that the image file specified is accessible and in a supported format (e.g., JPG, PNG).

Set-Up

Database Entries(contd.):-

3. Update User Details:

- Locate the section of the code where user details are specified. This could be in a registration function or a configuration section.
- Update the user details with the correct information. This typically includes attributes like username, password, name, email, position, etc.
- Replace the placeholder values with the actual user details.
For example:
 - username = "your_username"
 - password = "your_password"
 - name = "Your Name"
 - email = "your_email@example.com"
 - position = "Your Position"
 - Update other user details as needed

4. Save and Close the File:

- After updating the image path and user details, save the changes to the run.py file.
- Close the file editor or IDE.

Running The Application

Starting the Project:-

1. Activate Virtual Environment (if not already activated):

- Navigate to the project directory in your terminal and activate the virtual environment using the following command:"cd path/to/Authentiface && .\venv\Scripts\activate"
- Replace path/to/Authentiface with the actual path to your Authentiface project directory.

2. Run the Project:

- Start the Authentiface application by running the following command in your terminal:"python main.py".
- This command initializes the Flask web server, which hosts the Authentiface application.

3. Access the Project in a Web Browser:

- Once the application is running, open a web browser and navigate to the provided URL. Typically, the URL will be something like http://localhost:5000.

4. User Authentication:

- Upon accessing the application in your web browser, you'll be prompted to log in.
- Enter the username and password configured in the project to proceed.

5. Compose and Send Emails:

- Within the Authentiface application, navigate to the email composition feature.
- Fill in the recipient's email address, subject, and message content.
- Optionally, attach files such as text documents or PDFs (limited to 100-150 KB).
- When ready, click the "Send" button to initiate the email sending process.

6. Facial Recognition for Email Sending:

- Before sending the email, the application performs another round of facial recognition.
- This time, the system verifies your identity to ensure that only authorized users can send emails.
- Follow the same facial expression prompts as described earlier.
- Successful completion of this authentication step allows the email to be sent.

Running The Application

Starting the Project(contd.):-

7. View Received Emails:

- Users can access their inbox to view received emails.
- Non-confidential emails can be viewed without additional authentication.
- However, viewing confidential emails requires another round of facial recognition for security purposes.

8. Terminate the Application:

- To stop the Authentiface application, return to your terminal where the Flask server is running.
- Press Ctrl + C to terminate the server.
- Optionally, deactivate the virtual environment by running deactivate in the terminal.

