

MASTER OF TECHNOLOGY (INTELLIGENT SYSTEMS)

USER GUIDE



- Enhancing Birdwatching with AI

GROUP MEMBERS

Ashiwin Rajendran (A0291191E)

Jithin Krishnan (A0249481W)

Prem Varijakzhan (A0291913B)

Shankar Sai Ganesh (A0168149U)

Accessing WingSpan Application

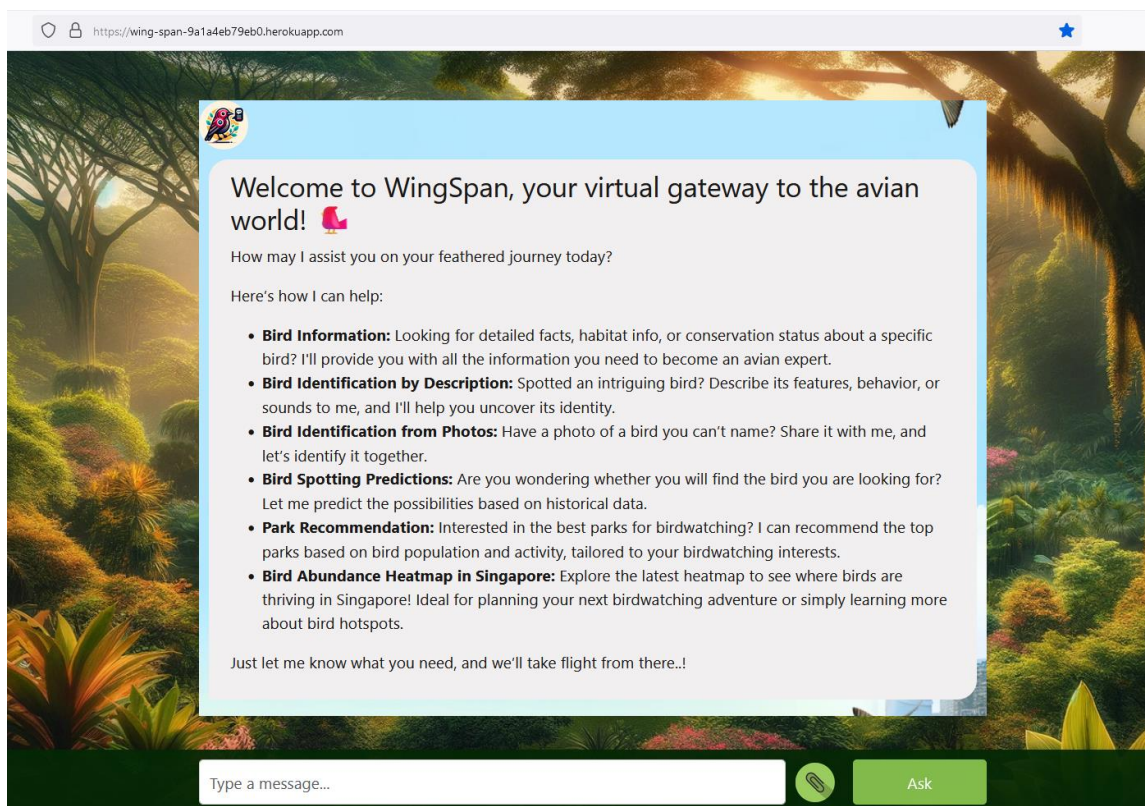
WingSpan is accessible both online and offline. This guide details on how to access the deployed online application as well as setup process for running the application locally (offline)

1. Accessing Online

The application is hosted on the Heroku cloud and can be accessed through the URL provided below.

URL: <https://wing-span-9a1a4eb79eb0.herokuapp.com/>

Interacting with the chat application is straightforward with the very intuitive user interface.



2. Running the Web application Locally

2.1 Setting Up a Python Environment Using Conda

1. Download and install Anaconda from the Anaconda website, ensuring you select a version that includes Conda.
2. Open Anaconda Prompt:
 - Launch from the Start Menu (Windows) or terminal (macOS and Linux).
3. Create a New Conda Environment with the runtime python version for the application:
 - Command: **`conda create -n wingspan python=3.11.7`**
4. Activate the Conda Environment:
 - Command: **`conda activate wingspan`**

2.2 Django Web Application Local Set up

1. Clone the below GitHub repository to your local machine.
<https://github.com/jithinkrn/IRS-PM-2024-05-05-IS06PT-WingSpan.git>

Command:

`git clone https://github.com/jithinkrn/IRS-PM-2024-05-05-IS06PT-WingSpan.git`

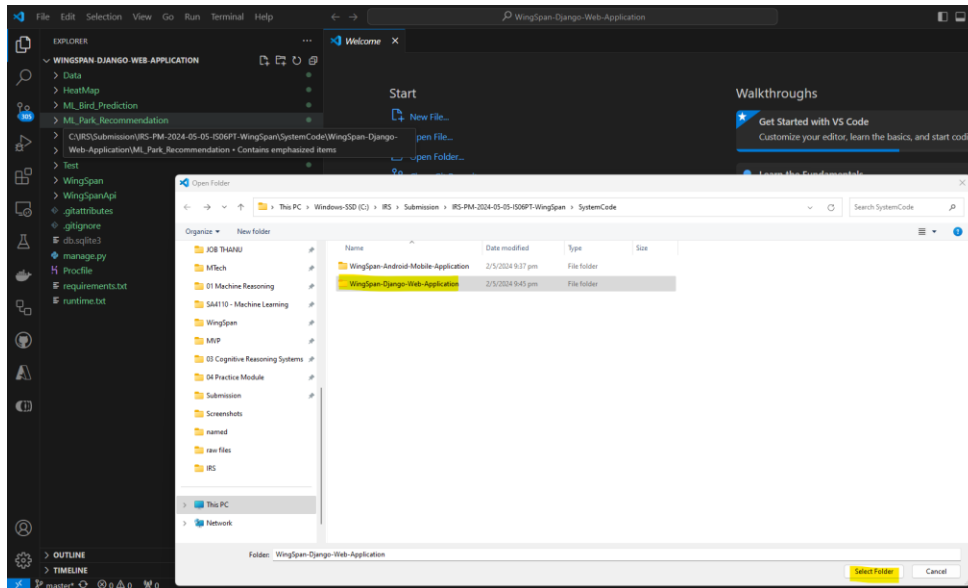
2. Install requirements.txt in your Conda environment created earlier. The requirements.txt file is located at:

`~/IRS-PM-2024-05-05-IS06PT-WingSpan\SystemCode\WingSpan-Django-Web-Application`

Command: **`pip install -r requirements.txt`**

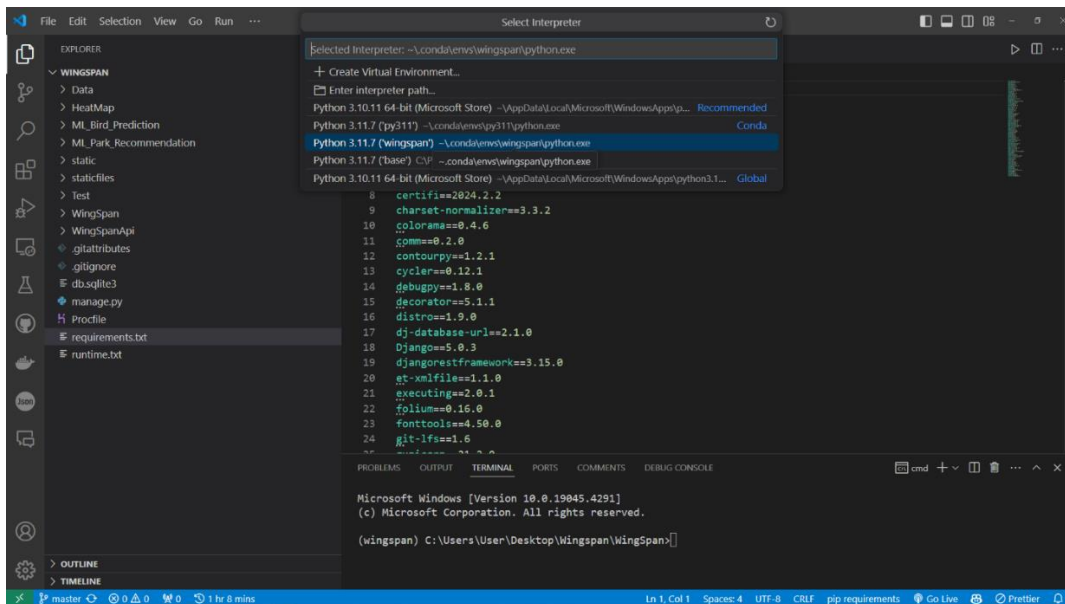
3. Launch Visual Studio Code and Open the Project Folder “WingSpan-Django-Web-Application” by navigating to below directory in the repo

~/IRS-PM-2024-05-05-IS06PT-WingSpan\SystemCode\WingSpan-Django-Web-Application



4. Selecting the Python Environments in Visual Studio Code

- Open Command Palette (Ctrl+Shift+P or Cmd+Shift+P).
- Choose the conda environment created earlier with *python=3.11.7* and *all the requirements installed*



5. Configure Open AI API Key as environment variable (The API key will be uploaded to Canvas with the submission documents in a text file). In visual studio menu go to Terminal > New Terminal and enter the command below

Windows Command:

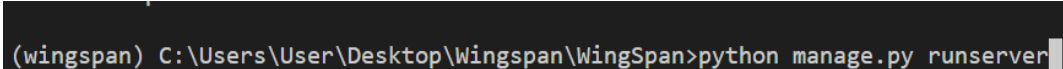
Run: `setx OPENAI_API_KEY "PROVIDED_API_KEY_HERE"`

Linux Command:

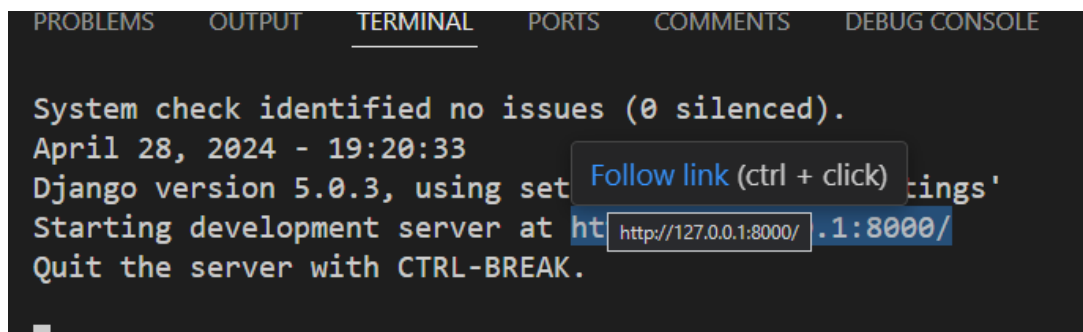
`export OPENAI_API_KEY "PROVIDED_API_KEY_HERE" :~/bin`

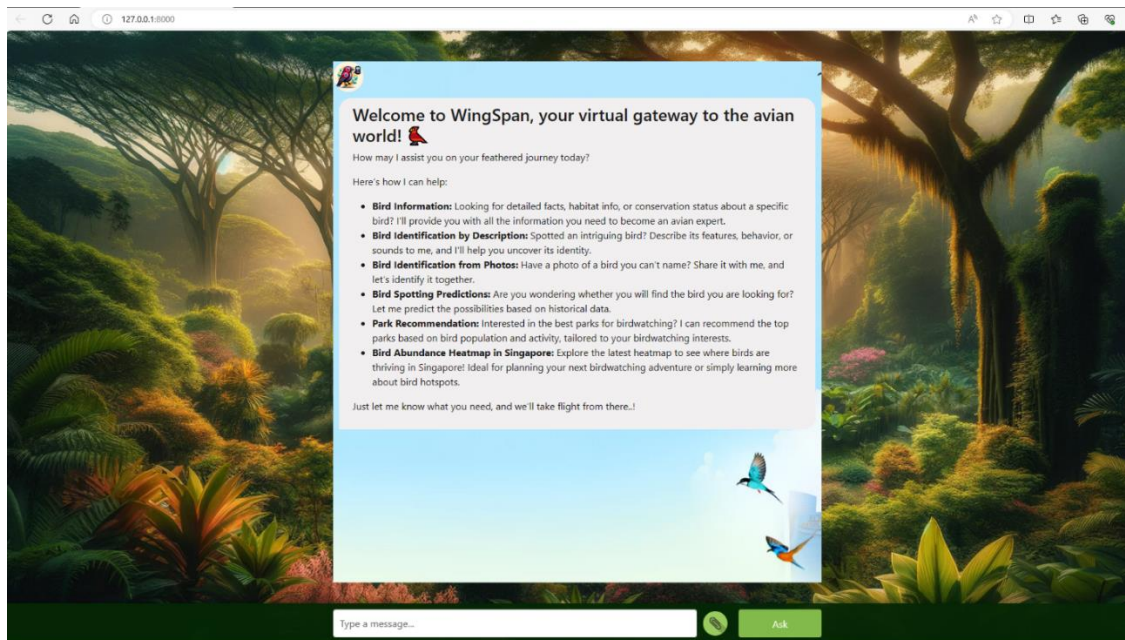
6. In the VS code terminal, enter the following command.

`python manage.py runserver`



Once the server runs. It will display the local host link. Open browser and navigate to `http://127.0.0.1:8000/` to view the application.





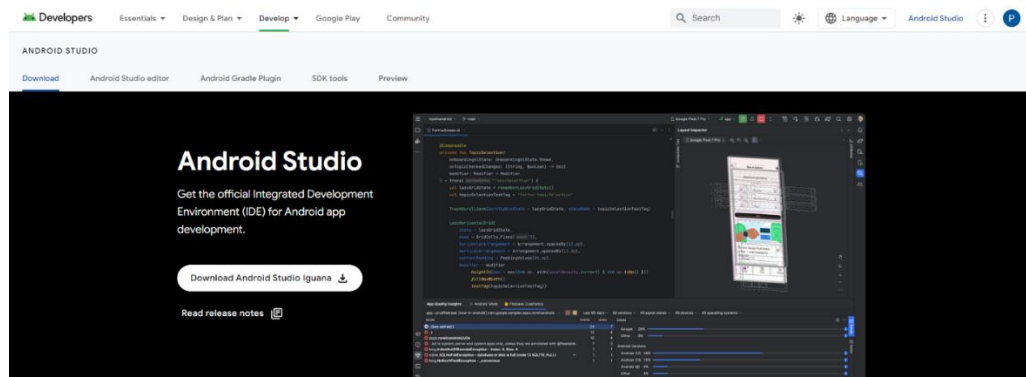
If the OPEN AI Key and other requirement setups are correct, you should be able to, chat with application.

7. To stop the server, you can press CTRL+C in the terminal.

3. Running the WingSpan Android Mobile Application Locally

3.1 Android Studio Setup.

- Download Android Studio from the below URL
https://developer.android.com/studio?_gl=1*datbhy*_up*MQ..&gclid=CjwKCAjw88yxBhBWEiwA7cm6pb48LLsVpv1FA3886J7c_aplXltuuFvBTcGujgREyaP5LpL7-NXlXoCjMUQAvD_BwE&gclsrc=aw.ds



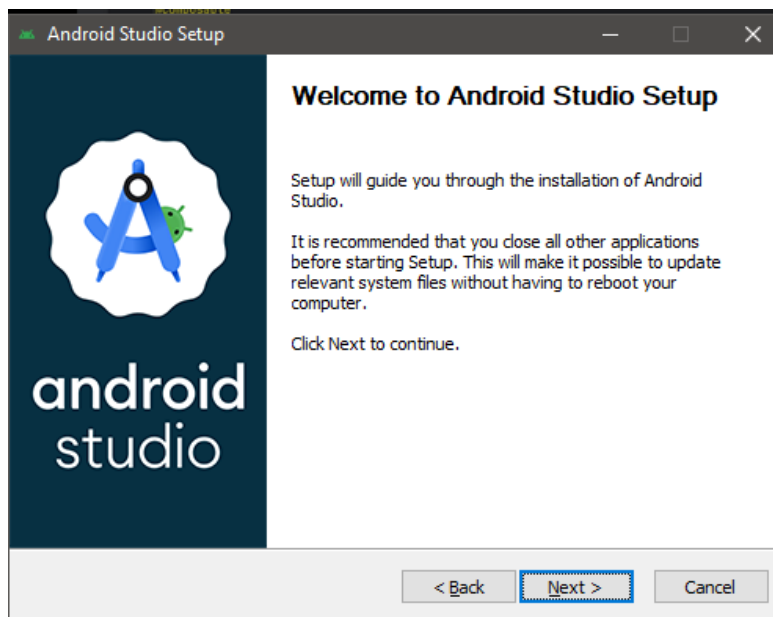
New features

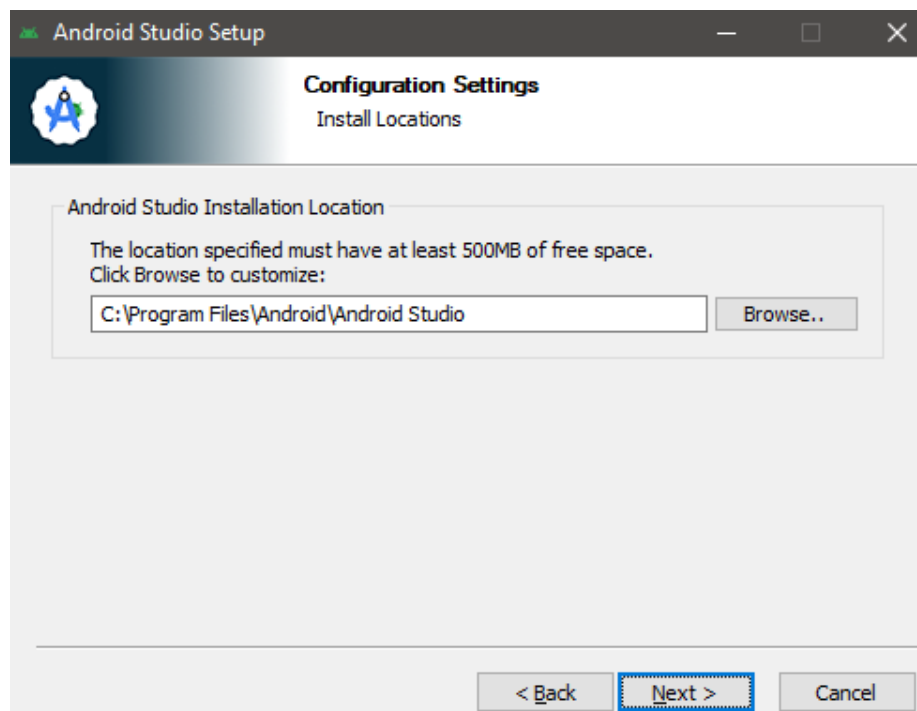
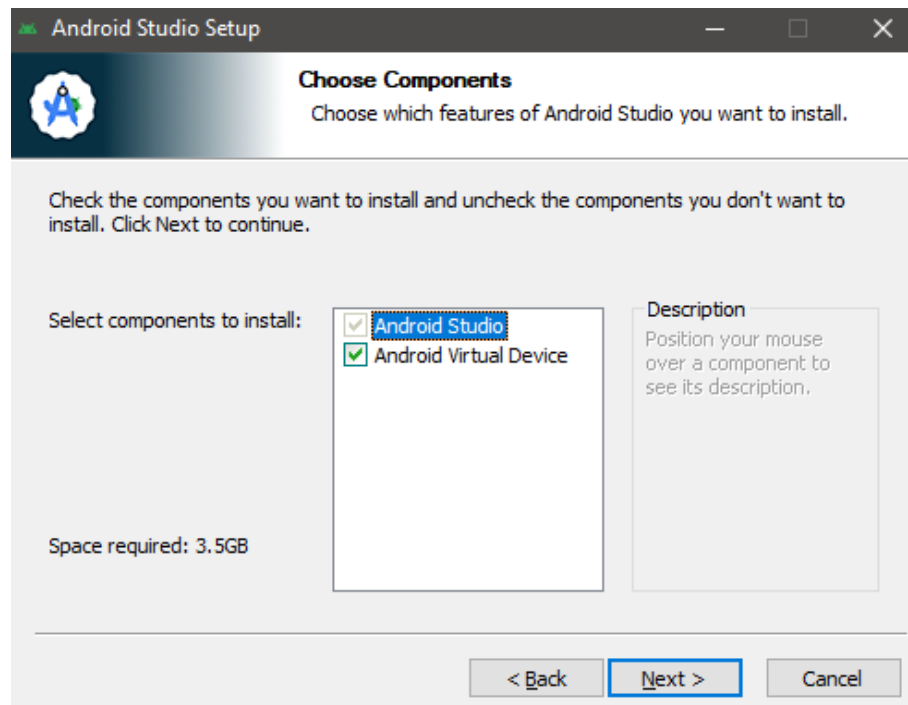


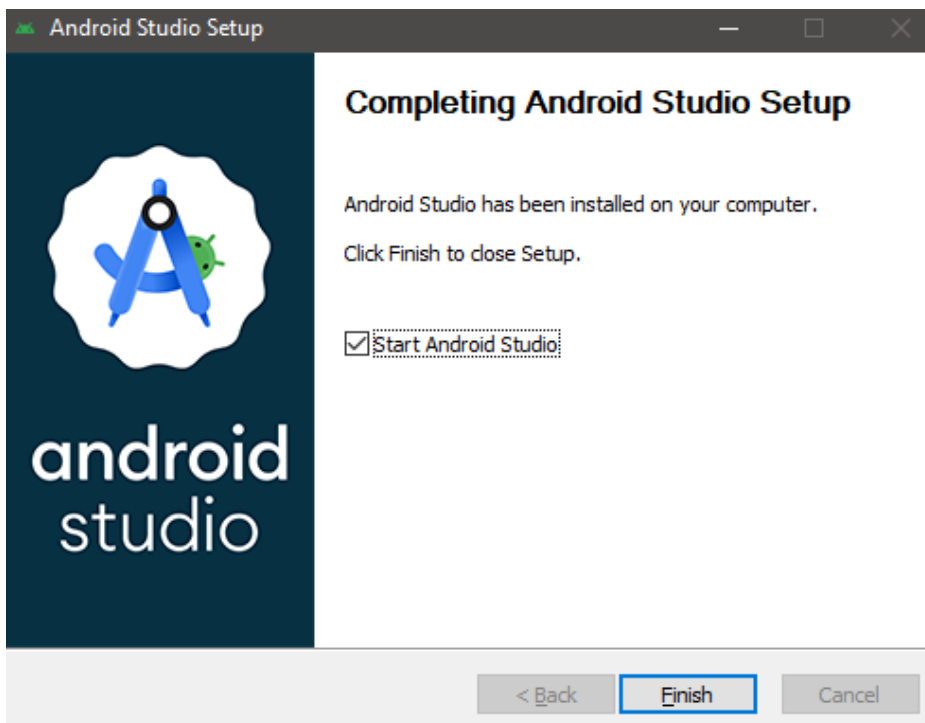
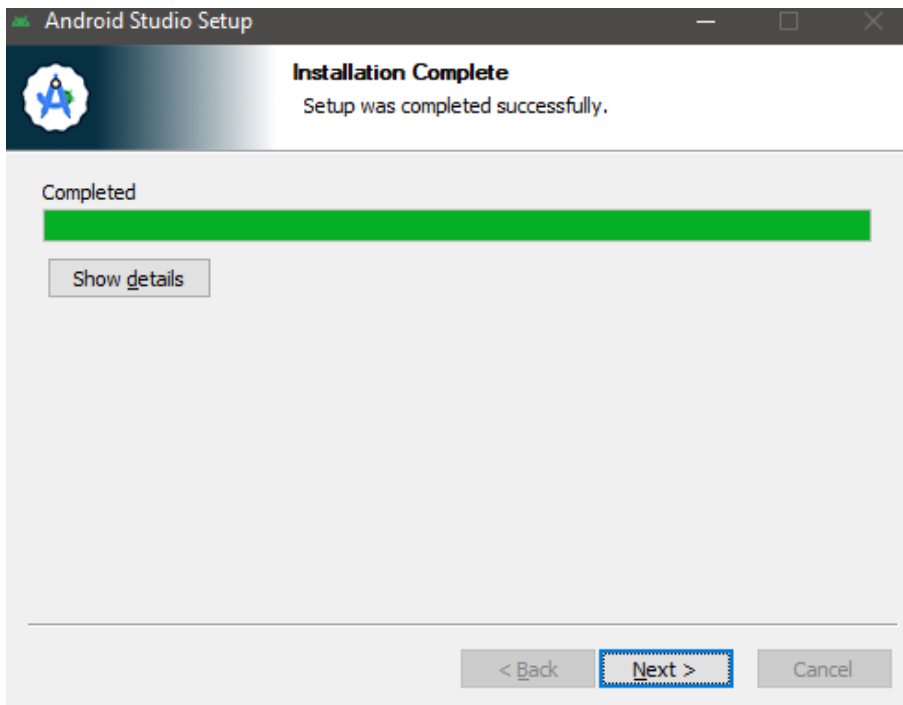
- Ensure your system meets the installation requirements.

3.2 Installing Android Studio

Follow the on-screen instructions to install Android Studio. This typically includes selecting an installation directory, choosing components to install, and configuring initial settings like memory allocations.





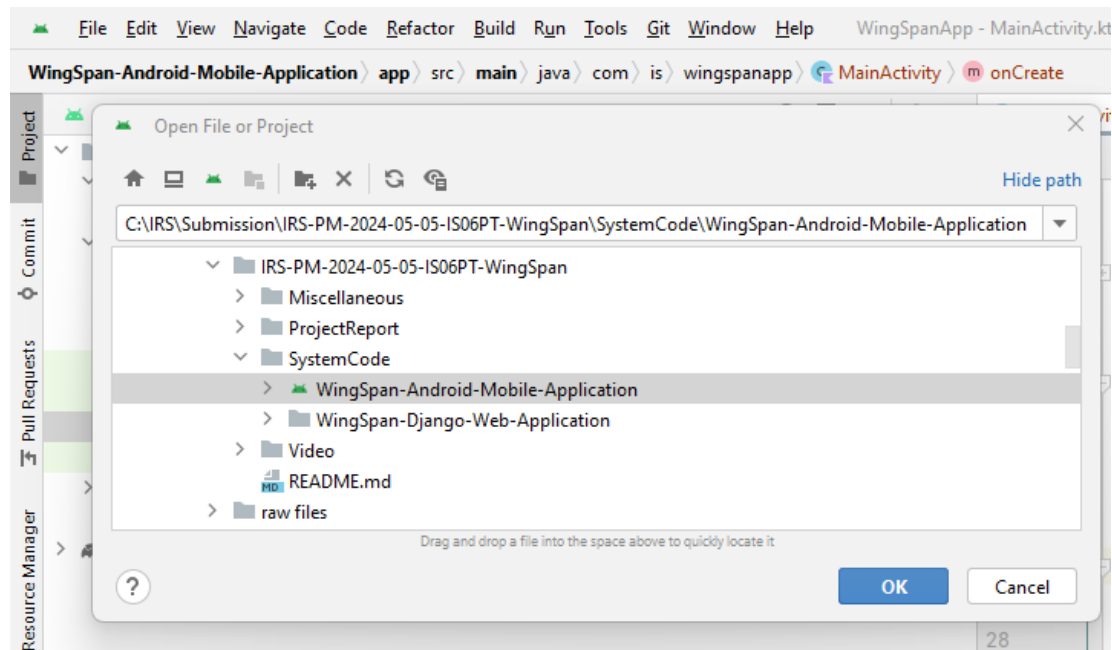


3.4 Opening the Project

Launch Android Studio and open the **WingSpan-Android-Mobile-Application** project with Android Studio.

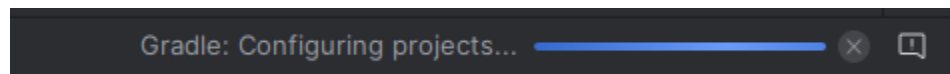
Location:

~\IRS-PM-2024-05-05-IS06PT-WingSpan\SystemCode\WingSpan-Android-Mobile-Application



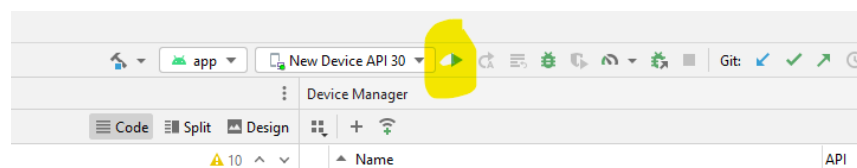
3.5 Gradle Build and Project Configuration

- Allow Android Studio to complete the Gradle build.



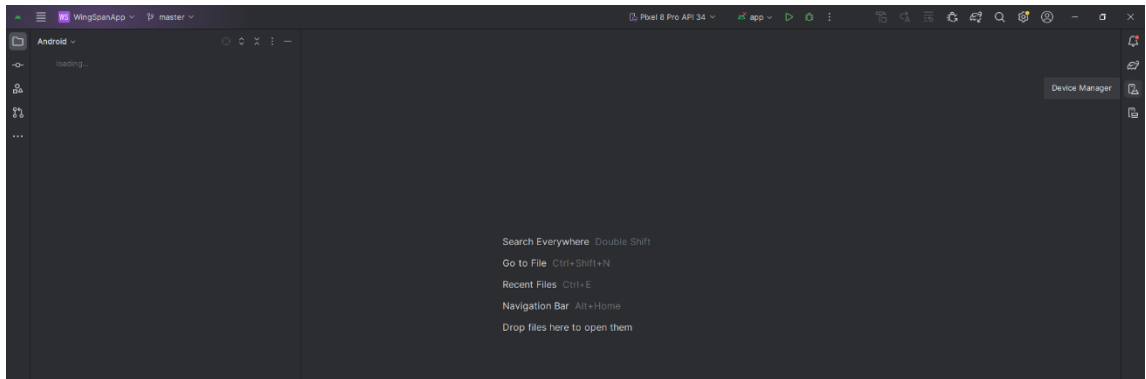
3.6 Running the application

Once the build is completed click the run Button from the menu.

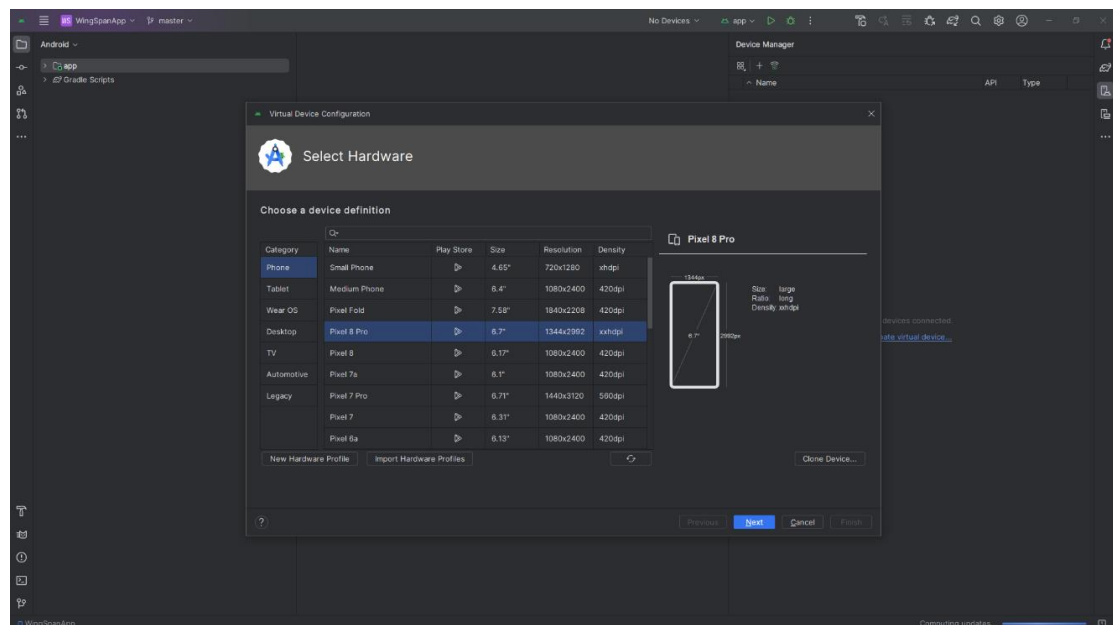


3.7 Additional Information: Setting Up and Running an Android Emulator in Android Studio

One the right panel click the Device Manager button.

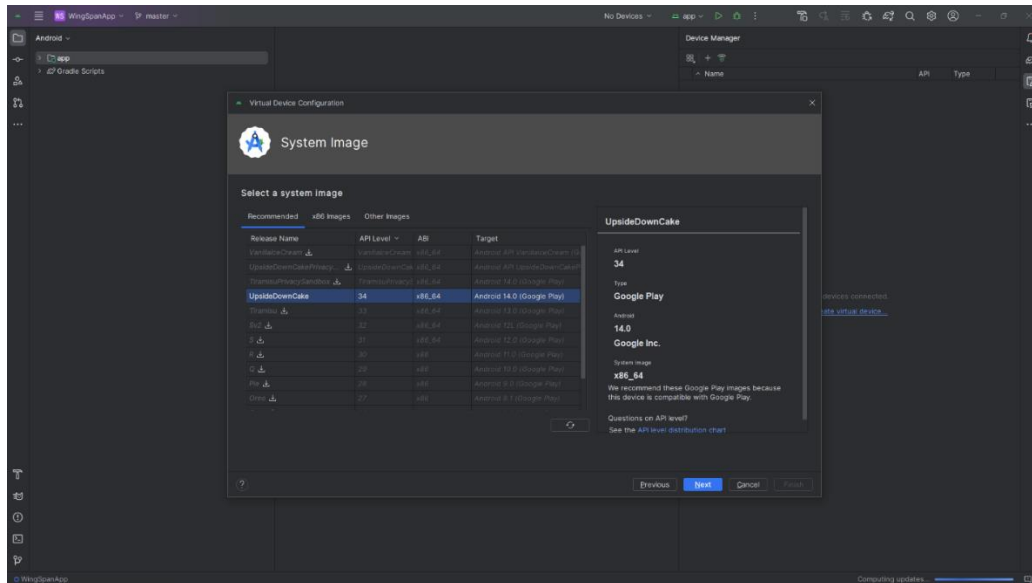


- Use the AVD Manager to create a virtual device (e.g., Pixel 4). Choose a device definition from the list of hardware profiles (e.g., Pixel 4) that matches the type of device you want to emulate. Click "Next" to continue.

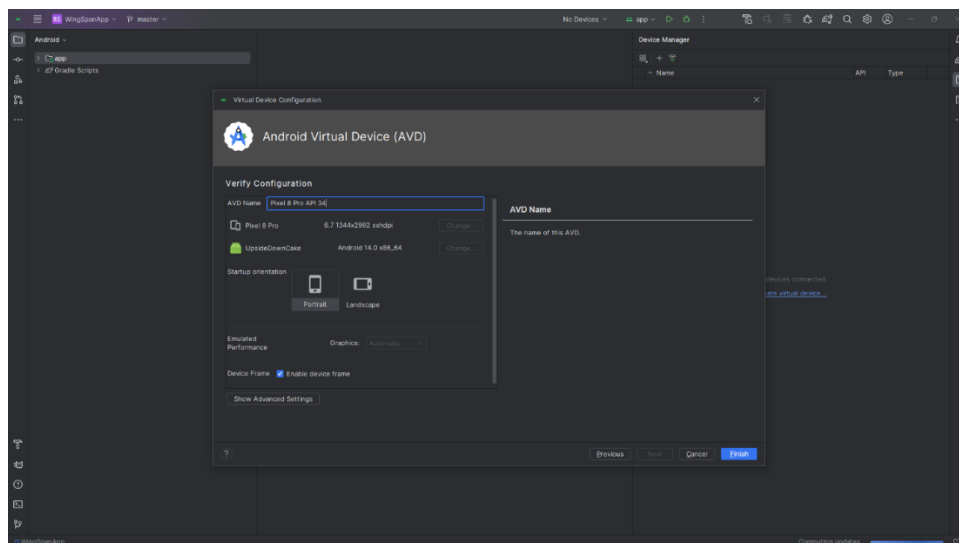


- Select a System Image. Download and select a system image. On the next screen, select a system image for your AVD. You will need to download a system image if you have not already. Choose an image with the desired API level and

Google Play services if you need them for your development. Click "Download" next to the system image if it is not already available on your system. After the download is completed, select the system image, and click "Next".

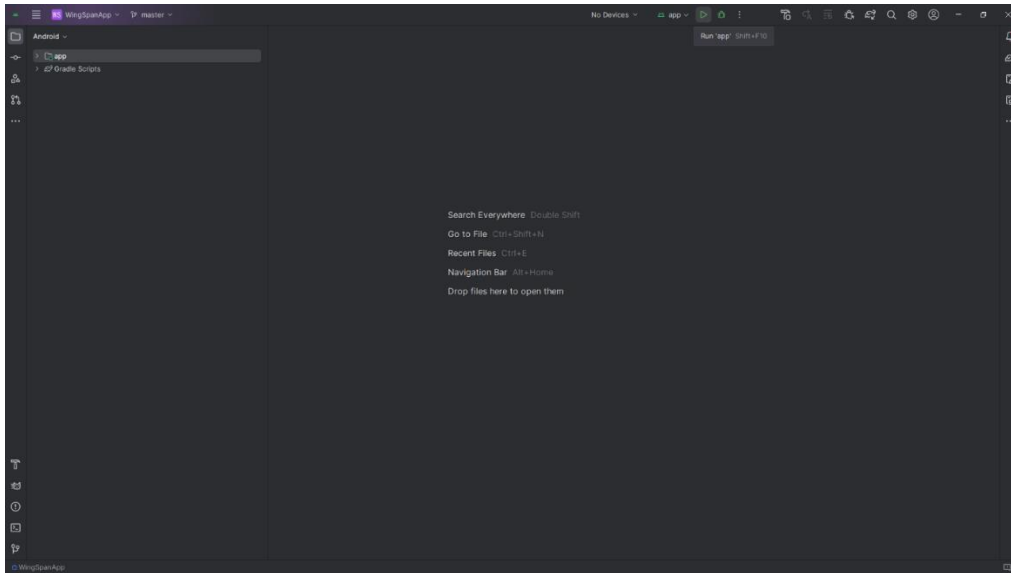


- Finish and Create AVD. Click "Finish" to create your Android Virtual Device.



- Run the Application on the Emulator: Launch the emulator and wait for it to boot. From the AVD Manager, find the virtual device you created and click on the "Play" button to launch the emulator. Wait for the emulator to start and load the

Android operating system. This can take a few minutes depending on your computer's specifications.



- Interact with the application on the emulator.

