Explanation of each line in app.py:

pythonCopy code

# Import Flask library and other necessary modules from flask import Flask, render\_template, request, redirect, url\_for, session, flash # Import functions from other Python files from home import home\_page from signup import signup\_page from login import login\_page, logout # Initialize the Flask app app = Flask(\_\_name\_\_)

* The first line imports the Flask library and other necessary modules for creating a web application.
* The next three lines import functions from the **home.py**, **signup.py**, and **login.py** files, which will handle the corresponding pages of the web application.
* Finally, **app = Flask(\_\_name\_\_)** initializes the Flask app.

pythonCopy code

# Set secret key for session management app.secret\_key = "secret\_key"

* This line sets a secret key for session management in the Flask app.

pythonCopy code

# Define routes for the Flask app app.add\_url\_rule("/", view\_func=home\_page) app.add\_url\_rule("/signup", view\_func=signup\_page, methods=["GET", "POST"]) app.add\_url\_rule("/login", view\_func=login\_page, methods=["GET", "POST"]) app.add\_url\_rule("/logout", view\_func=logout)

* These lines define the routes for the Flask app. Each route is associated with a corresponding view function that will handle the request and return a response.

pythonCopy code

# Define main function for running the Flask app if \_\_name\_\_ == "\_\_main\_\_": app.run(debug=True)

* This code block defines the main function for running the Flask app. The **if \_\_name\_\_ == "\_\_main\_\_":** line ensures that the app only runs if it is executed directly, and not if it is imported as a module in another file. The **app.run(debug=True)** line runs the Flask app in debug mode, allowing for debugging messages to be printed to the console.

Explanation of each line in home.py:

pythonCopy code

# Import necessary modules from flask import render\_template, session # Define view function for home page def home\_page(): # Check if user is logged in and set context variable accordingly if "username" in session: logged\_in = True else: logged\_in = False # Render home.html template with context variables return render\_template("home.html", logged\_in=logged\_in)

* The first line imports necessary modules for rendering a template and managing sessions in Flask.
* The **home\_page()** function is a view function for the home page of the web application.
* The **if "username" in session:** line checks if a username is stored in the session, indicating that the user is logged in. If so, **logged\_in** is set to **True**; otherwise, it is set to **False**.
* The **return render\_template("home.html", logged\_in=logged\_in)** line renders the **home.html** template and passes the **logged\_in** context variable to the template.

Explanation of each line in signup.py:

pythonCopy code

# Import necessary modules from flask import render\_template, request, redirect, url\_for, session, flash import bcrypt # Import database functions from database import create\_user # Define view function for signup page def signup\_page(): # If user submits signup form if request.method == "POST": # Get username, password, and email from form username = request.form["username"] password = request.form["password"].encode("utf-8") email = request.form["email"]