Task Explanation and Approach

Task Explanation

The goal of these scripts is to automate the process of filling out an online form and then send a confirmation email with a screenshot of the submitted form.

1. **Automate Form Filling**:

   - Fill out a specific Google Form with given details (e.g., name, phone number, email, etc.).
   - Capture a screenshot of the confirmation page after submitting the form.

2. **Send Confirmation Email**:

   - Send an email to specified recipients with the subject, body, and the screenshot of the confirmation page as an attachment.

Approach

1. Automate Form Filling (`automateform.py`)

1. **Setup WebDriver**:

   - The script uses Selenium to control a web browser (Chrome).
   - Initializes a Chrome WebDriver instance.

2. **Navigate to Form**:

   - The WebDriver navigates to the specified Google Form URL.

3. **Fill the Form**:

   - Defines a function `fill_form` that takes several parameters (name, phone, email, address, pin, DOB, gender, verification code).
   - Locates the input fields on the form using their class names.
   - Fills the form fields with the provided data.

4. **Submit the Form**:

   - Locates and clicks the submit button to submit the form.
   - Waits for the confirmation page to load.

5. **Capture Screenshot**:

   - Takes a screenshot of the confirmation page and saves it as `confirmation_page.png`.

6. **Close Browser**:

   - Closes the browser once the form submission and screenshot capture are complete.

2. Send Confirmation Email (`app.py`)

1. **Setup Flask Application**:

   - Creates a Flask application instance.
   - Configures the Flask app with email server settings (using Gmail's SMTP server).

2. **Configure Email Sending**:

   - Sets up email sending using the `yagmail` library with the sender's email and app password.

3. **Define Email Route**:

   - Defines a route `/send_email` that triggers the email sending process.
   - Specifies the recipients, subject, body, and attachment (the screenshot `confirmation_page.png`).

4. **Ensure Attachment Exists**:

   - Before running the Flask app, checks if `confirmation_page.png` exists to ensure the form was successfully filled and the screenshot was taken.

5. **Run the Flask App**:

   - If the screenshot file is present, runs the Flask app in debug mode.
   - When the `/send_email` route is accessed, sends the email with the provided details and attachment.

This approach ensures that the form is filled out and submitted automatically, and a confirmation email is sent with proof of submission in the form of a screenshot.