

FINAL PROJECT PYTHON (Bano Qabil 3.0)

---HOME_GUARDIAN---

Home_Guardian is a simple and easy-to-use web application designed to help you manage and protect your home. With this tool, you can keep track of various home services, get reminders for maintenance, and ensure that your home is running smoothly.

Purpose:

The purpose of **Home_Guardian** is to offer a one-stop solution for homeowners to:

- ❖ Book cleaning and other home services.
- ❖ Stay organized with home tasks.
- ❖ Easily get in touch with service providers.

Main Features:

1. **Home Page:** This is the landing page that welcomes users and shows the best cleaning services available.
2. **Services Page:** A page that lists all the services offered, like cleaning, repairs, etc. You can also book services from here.
3. **About Page:** This page gives information about what the **Home_Guardian** platform is, and what it aims to achieve.
4. **Contact Page:** A form where you can reach out to the team for any inquiries or assistance.
5. **Place Order:** Users can place an order directly from the home page by filling out a form with their details and service needs.

Technology Stack:

- ❖ **Backend:** Python (Flask)
- ❖ **Frontend:** HTML, CSS
- ❖ **Version Control:** Git & GitHub

1. Setting Up the Project

- ❖ **Clone the Project:** Download the project to your computer.
- ❖ **Install Required Libraries:** Ensure all necessary Python packages are installed.
- ❖ **Run the Application:** Start the web application on your local machine.
- ❖ **View the Website:** Access the website via your web browser.

2. Placing an Order

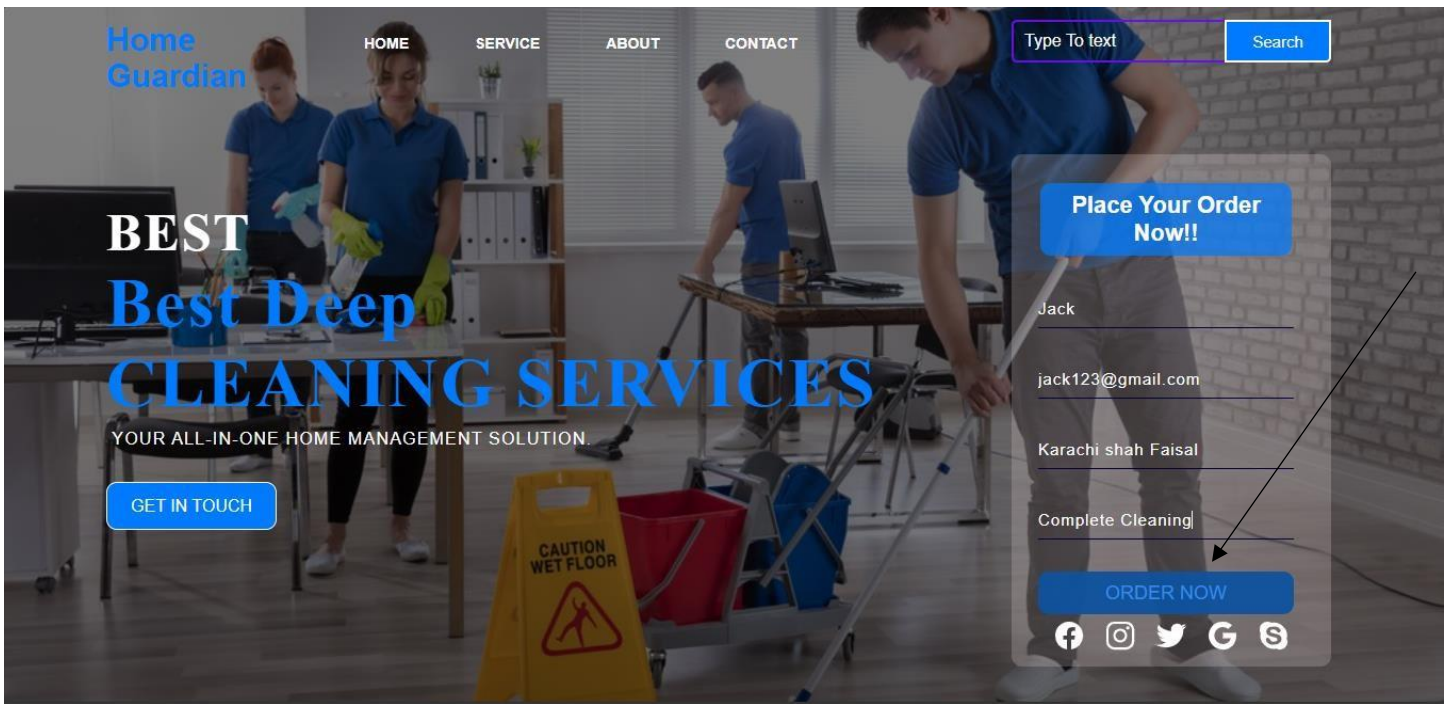
- ❖ Fill out the order form on the Home Page to request services.
- ❖ Submit the form and receive a confirmation.

Clone the repository:

- `git clone https://github.com/jawadhassan1112003/Home_Guardian.git`
- `cd Home Guardian`
- `pip install -r requirements.txt`
- `python app.py`
- Open your web browser and go to `http://127.0.0.1:5000/` to view the website.

SCREEN SHOTS

```
1 from flask import Flask, render_template, request, redirect, url_for
2
3 # Create a Flask application instance
4 app = Flask(__name__)
5
6 # Route for the home page
7 @app.route('/')
8 def home():
9     return render_template('Home_Guardian.html')
10
11 # Route to handle order placement when user submits the form
12 @app.route('/place_order', methods=['POST'])
13 def place_order():
14     # Get form data from the POST request
15     name = request.form.get('name')
16     email = request.form.get('email')
17     area = request.form.get('area')
18     services = request.form.get('services')
19
20     # Print order details to the console for debugging purposes
21     print(f"Name: {name}")
22     print(f"Email: {email}")
23     print(f"Area: {area}")
24     print(f"Services: {services}")
25
26     # Save the order details into 'orders.txt' file
27     with open('orders.txt', 'a') as file:
28         file.write(f"Name: {name}\n")
29         file.write(f"Email: {email}\n")
30         file.write(f"Area: {area}\n")
31         file.write(f"Services: {services}\n")
32         file.write("\n")
33
34     # Redirect to a thank you page after form submission
35     return redirect(url_for('thank_you'))
36
37 # Route for the thank you page
38 @app.route('/thank_you')
39 def thank_you():
40     return "Thank you for your order! We will contact you soon."
41
42 # Route for the services page
43 @app.route('/service')
44 def services():
45     return render_template('service.html')
46
47 # Route to handle booking of services from the services page
48 @app.route('/book_service', methods=['POST'])
49 def book_service():
50     # Get service details from the POST request
51     service_name = request.form.get('service_name')
52     service_price = request.form.get('service_price')
53
54     # Print booking details to the console for debugging purposes
55     print(f"Service booked: {service_name} at Rs.{service_price}")
56
57     # Save the booking details into 'bookings.txt' file
58     with open('bookings.txt', 'a') as file:
59         file.write(f"Service booked: {service_name} at Rs.{service_price}\n")
60
61     # Return a confirmation message to the user
62     return f"Thank you for booking {service_name}"
63     ! Kindly fill the form from the Home Page! We will contact you soon."
64
65 # Route for the about page
66 @app.route('/about')
67 def about():
68     return render_template('about.html')
69
70 # Route for the contact page
71 @app.route('/contact')
72 def contact():
73     return render_template('contact.html')
74
75 # Route to handle form submission on the contact page
76 @app.route('/submit_contact', methods=['POST'])
77 def submit_contact():
78     # Get contact form data from the POST request
79     name = request.form['name']
80     email = request.form['email']
81     phone = request.form.get('phone')
82     message = request.form['message']
83
84     # Print contact details to the console for debugging purposes
85     print(f"Name: {name}")
86     print(f"Email: {email}")
87     print(f"Phone: {phone}")
88     print(f"Message: {message}")
89
90     # Save the contact form details into 'contacts.txt' file
91     with open('contacts.txt', 'a') as file:
92         file.write(f"Name: {name}\n")
93         file.write(f"Email: {email}\n")
94         file.write(f"Phone: {phone}\n")
95         file.write(f"Message: {message}\n")
96         file.write("\n")
97
98     # Return the contact page with a success message after form submission
99     return render_template('contact.html', success=True)
100
101 # Run the application in debug mode
102 if __name__ == '__main__':
103     app.run(debug=True)
```



Thank you for your order! We will contact you soon.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Jawad Hassan\Desktop\Final Project python> python -u "c:\Users\Jawad Hassan\Desktop\Final Project python\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 105-788-731
127.0.0.1 - - [31/Aug/2024 17:42:32] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:42:32] "GET /static/css/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:42:32] "GET /static/images/background.jpeg HTTP/1.1" 304 -
Name: Jack
Email: jack123@gmail.com
Area: Karachi shah Faisal
Services: Complete Cleaning
127.0.0.1 - - [31/Aug/2024 17:43:41] "POST /place_order HTTP/1.1" 302 -
127.0.0.1 - - [31/Aug/2024 17:43:41] "GET /thank_you HTTP/1.1" 200 -
```

```
app.py A M orders.txt U X Re
orders.txt
1 Name: jack
2 Email: jack123@gmail.com
3 Area: Karachi shah Faisal
4 Services: Complete Cleaning
5
6
```

OUR SERVICES



Professional Carpet
Cleaning
Rs.499

Book Now



Professional Home
Cleaning Service
Rs.2499

Book Now



Deep Cleaning
Services
Rs.4999

Book Now

Back Now

127.0.0.1:5000/book_service

Thank you for booking Professional Carpet Cleaning! Kindly fill form; from the Home Page! We will contact you soon.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Jawad Hassan\Desktop\Final Project python> python -u "c:\Users\Jawad Hassan\Desktop\Final Project python\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 105-788-731
127.0.0.1 - - [31/Aug/2024 17:52:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:52:50] "GET /static/css/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:50] "GET /static/images/background.jpeg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /service HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/css/service.css HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image1.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image2.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image3.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image4.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image5.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image6.jpg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image7.png HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image9.png HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:52:52] "GET /static/images/image8.jpg HTTP/1.1" 304 -
Service booked: Professional Carpet Cleaning at Rs.499
127.0.0.1 - - [31/Aug/2024 17:52:59] "POST /book_service HTTP/1.1" 200 -
```

bookings.txt

1 Service booked: Professional Carpet Cleaning at Rs.499

2

Home Services

[Home](#)[Services](#)[Contact](#)

Contact Us

Have questions? We're here to help! Reach out to us using the form below.

Name

Email

Phone Number

Message

Your message has been sent successfully!

© 2024 Home Services. All Rights Reserved.

Contact us: info@homeservices.com

[Back Now](#)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Jawad Hassan\Desktop\Final Project python> python -u "c:\Users\Jawad Hassan\Desktop\Final Project python\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 105-788-731
127.0.0.1 - - [31/Aug/2024 17:58:40] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:58:40] "GET /static/css/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:58:40] "GET /static/images/background.jpeg HTTP/1.1" 304 -
127.0.0.1 - - [31/Aug/2024 17:58:44] "GET /contact HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:58:44] "GET /static/css/contact.css HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:59:26] "GET /contact HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 17:59:26] "GET /static/css/contact.css HTTP/1.1" 304 -
Name: Jack
Email: jack123@gmail.com
Phone: 03186746904
Message: I want complete home cleaning package is their any discount package available..
127.0.0.1 - - [31/Aug/2024 18:01:03] "POST /submit_contact HTTP/1.1" 200 -
127.0.0.1 - - [31/Aug/2024 18:01:03] "GET /static/css/contact.css HTTP/1.1" 304 -
```

≡ contacts.txt

```
1 Name: jack
2 Email: jack123@gmail.com
3 Phone: 03186746904
4 Message: I want complete home cleaning package is their any discount package available..
5
```

About HomeGuardian

Your all-in-one solution for managing and protecting your home

HomeGuardian is here to help you stay on top of maintenance tasks, get reminders, and ensure everything runs smoothly with ease. We believe that a well-maintained home is a happy home, and our mission is to make home management effortless for you.

Who We Are

We are a team of home maintenance enthusiasts who understand the importance of keeping your home in top shape. Our platform combines technology and practicality to offer a seamless home management experience.

What We Do

Our platform allows you to schedule maintenance tasks, receive timely reminders, and ensure that everything in your home is running smoothly. Whether it's changing filters, checking smoke alarms, or scheduling repairs, HomeGuardian is your go-to solution.

Why Choose Us?

HomeGuardian isn't just a tool; it's a partner in maintaining your home's value and safety. We offer personalized features that adapt to your needs, helping you avoid costly repairs and keep your home in perfect condition.

[Back Now](#)