Project Description:

The project involves the implementation of two tables: one for customers and another for orders. The customers' table contains details such as a generated ID and name, and additional customizable fields like email addresses and contact numbers. The orders table is linked to the customers table through a Customer ID, which acts as a primary key. This project can be utilized as a customer and order database for a retail service or a similar application.

Web App Functionality:

The web app facilitates the management of the relationship between customers and orders. It provides administrative access for editing and making necessary changes. Access controls are implemented, allowing the administrator to specify which users can access specific tables. This functionality proves useful in scenarios where efficient management of customer and order data is paramount.

Need for Two Tables:

The web application requires at least two tables to establish a relational database. This relationship is akin to a login and password page, connecting multiple tables through a primary key attribute. In this case, the two tables (customers and orders) are interconnected by the Customer ID, serving as a key component for maintaining data integrity and coherence. Having two tables allows for a more structured and efficient representation of the relationship between customers and their associated orders.

Database and Access Method Decisions:

The decision to choose the database was based on the assignment's criteria, emphasizing CRUD activities and the use of two tables. The choice of a customers and orders database was intentional, aiming for simplicity and relevance to everyday online experiences. This structure enables a straightforward demonstration of CRUD operations. The access method was designed to align with the project's requirements, allowing the administrator to control and define access privileges. This decision was made to ensure secure and tailored access to the tables based on specific user roles. shopping or selling. This made me choose the database idea overall.

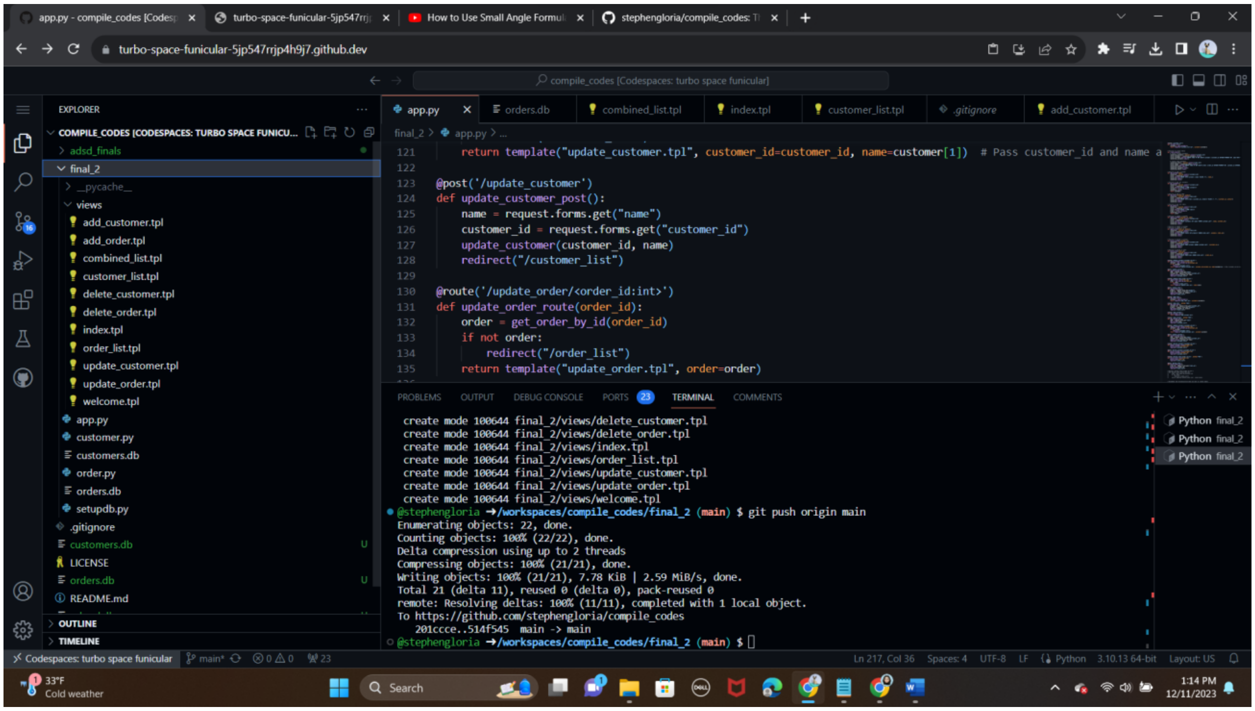
Provide your source code for the project, clearly indicating the location of the code – 3

- The source code for the project is from:
    - https://github.com/gregdelozier/advanced-database
    - https://github.com/gregdelozier/advanced-database/tree/main/beta

- Provide a link to the repository containing the code:
    - **The GITHUB LINK**: https://github.com/stephengloria/compile_codes

- Following is the embedded screenshots for the web application running and performing CRUD activities.

This is a screenshot of the code running and to show the connecting of app.py



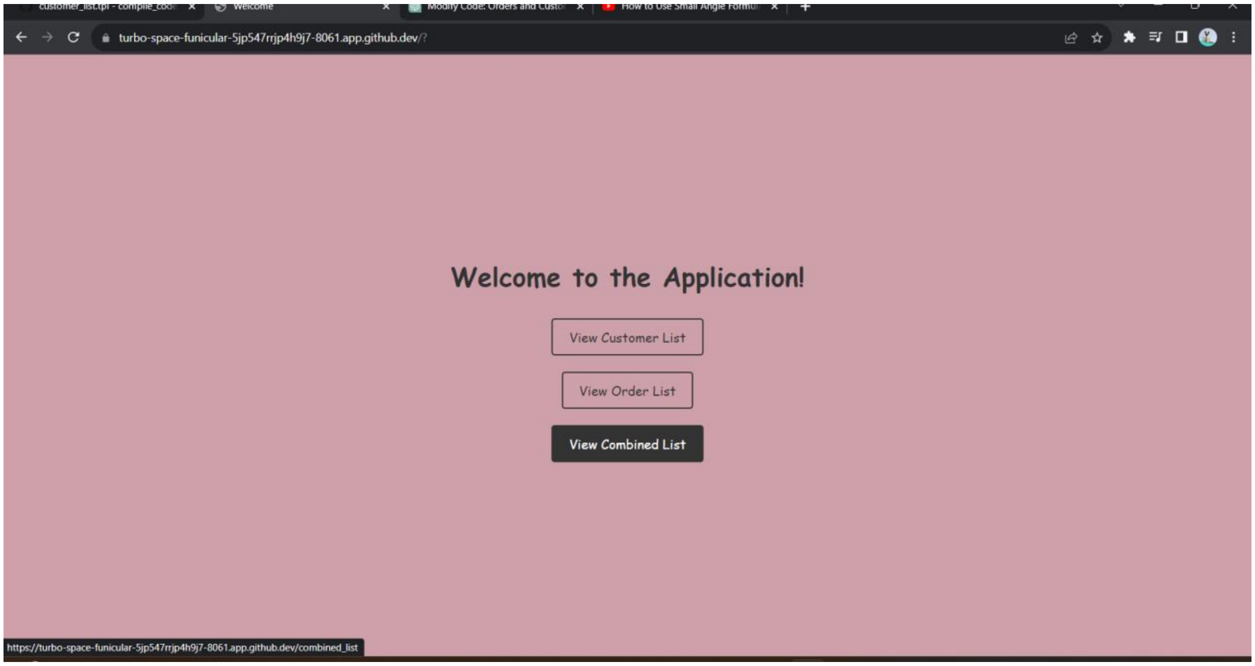
```python
    return template("update_customer.tpl", customer_id=customer_id, name=customer[1])  # Pass customer_id and name a

@post('/update_customer')
def update_customer_post():
    name = request.forms.get("name")
    customer_id = request.forms.get("customer_id")
    update_customer(customer_id, name)
    redirect("/customer_list")

@route('/update_order/<order_id:int>')
def update_order_route(order_id):
    order = get_order_by_id(order_id)
    if not order:
        redirect("/order_list")
    return template("update_order.tpl", order=order)
```

```
 create mode 100644 final_2/views/delete_customer.tpl
 create mode 100644 final_2/views/delete_order.tpl
 create mode 100644 final_2/views/index.tpl
 create mode 100644 final_2/views/order_list.tpl
 create mode 100644 final_2/views/update_customer.tpl
 create mode 100644 final_2/views/update_order.tpl
 create mode 100644 final_2/views/welcome.tpl
@stephengloria ➜/workspaces/compile_codes/final_2 (main) $ git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 2 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (21/21), 7.78 KiB | 2.59 MiB/s, done.
Total 21 (delta 11), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (11/11), completed with 1 local object.
To https://github.com/stephengloria/compile_codes
   201ccce..514f545  main -> main
@stephengloria ➜/workspaces/compile_codes/final_2 (main) $ 
```

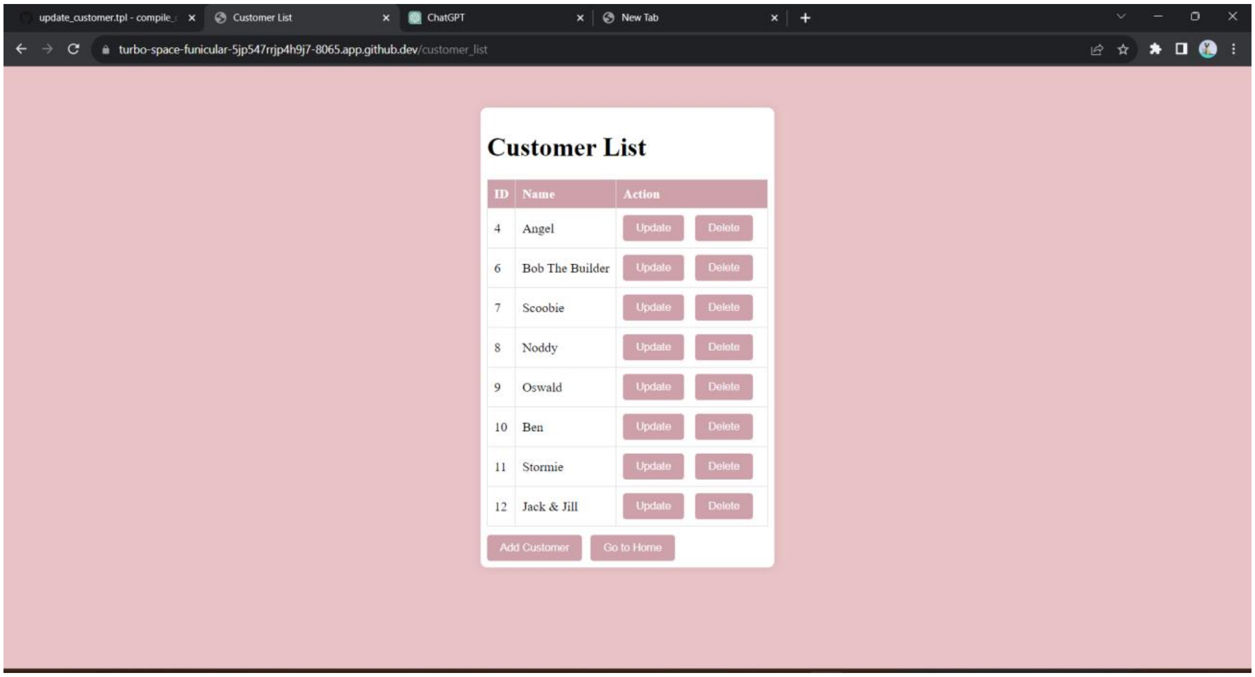
The Welcome Page


customer_list.tpl - compile_code
Welcome
Modify Code: Orders and Custo
How to Use Small Angle Formul
turbo-space-funicular-5jp547rrjp4h9j7-8061.app.github.dev/?

Welcome to the Application!

View Customer List
View Order List
View Combined List
https://turbo-space-funicular-5jp547rrjp4h9j7-8061.app.github.dev/combined_list

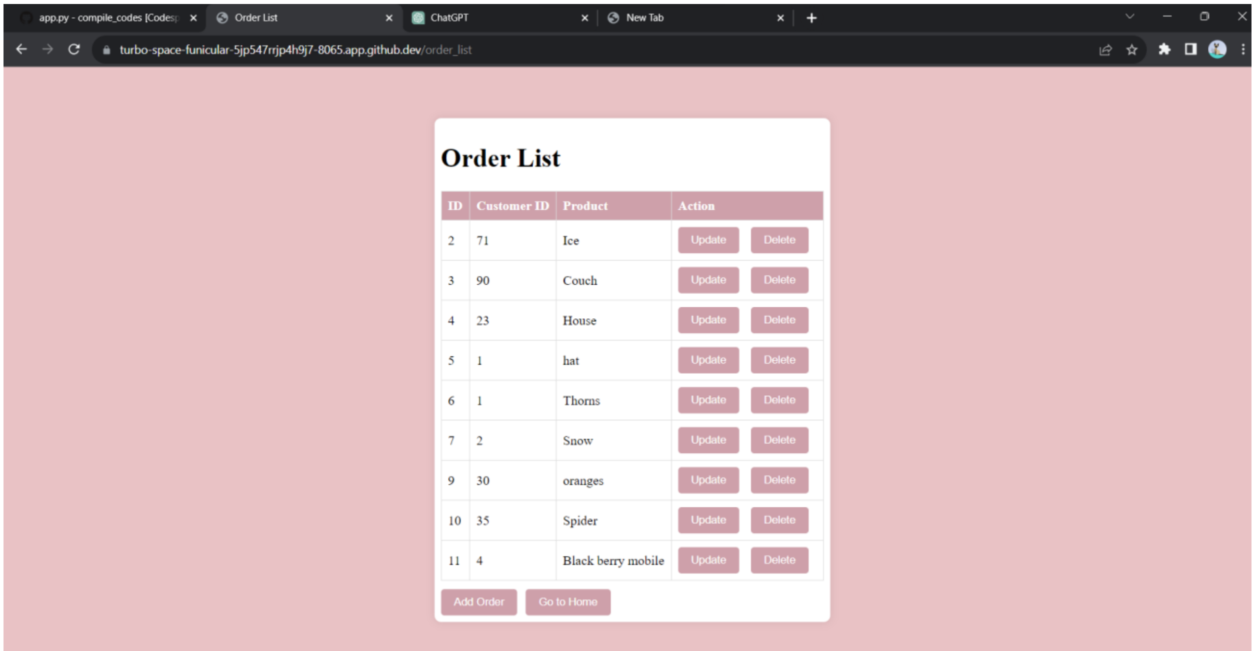
CUSTOMERS LIST


update_customer.tpl - compile_
Customer List
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/customer_list

Customer List

ID	Name	Action
4	Angel	Update Delete
6	Bob The Builder	Update Delete
7	Scoobie	Update Delete
8	Noddy	Update Delete
9	Oswald	Update Delete
10	Ben	Update Delete
11	Stormie	Update Delete
12	Jack & Jill	Update Delete

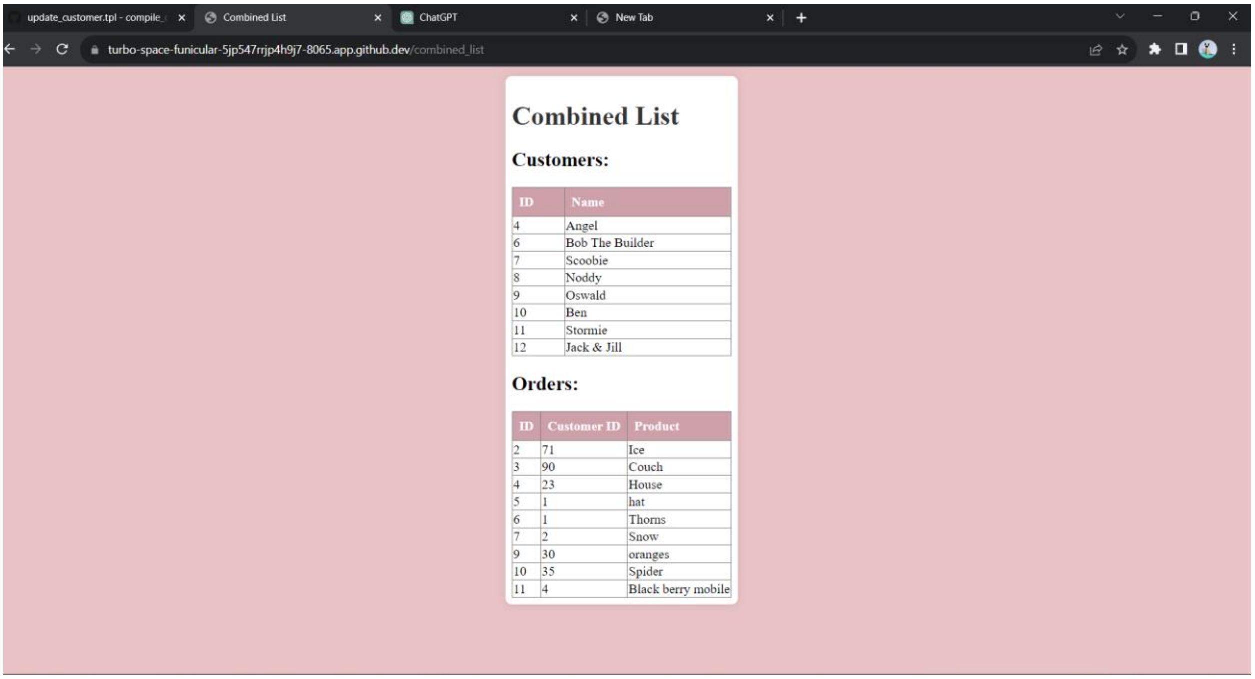
ORDER LIST-


app.py - compile_codes [Codesp
Order List
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/order_list

Order List

ID	Customer ID	Product	Action
2	71	Ice	Update Delete
3	90	Couch	Update Delete
4	23	House	Update Delete
5	1	hat	Update Delete
6	1	Thorns	Update Delete
7	2	Snow	Update Delete
9	30	oranges	Update Delete
10	35	Spider	Update Delete
11	4	Black berry mobile	Update Delete

COMBINED LIST- SHOWING THE 2 TABLES


update_customer.tpl - compile_
Combined List
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/combined_list

Combined List

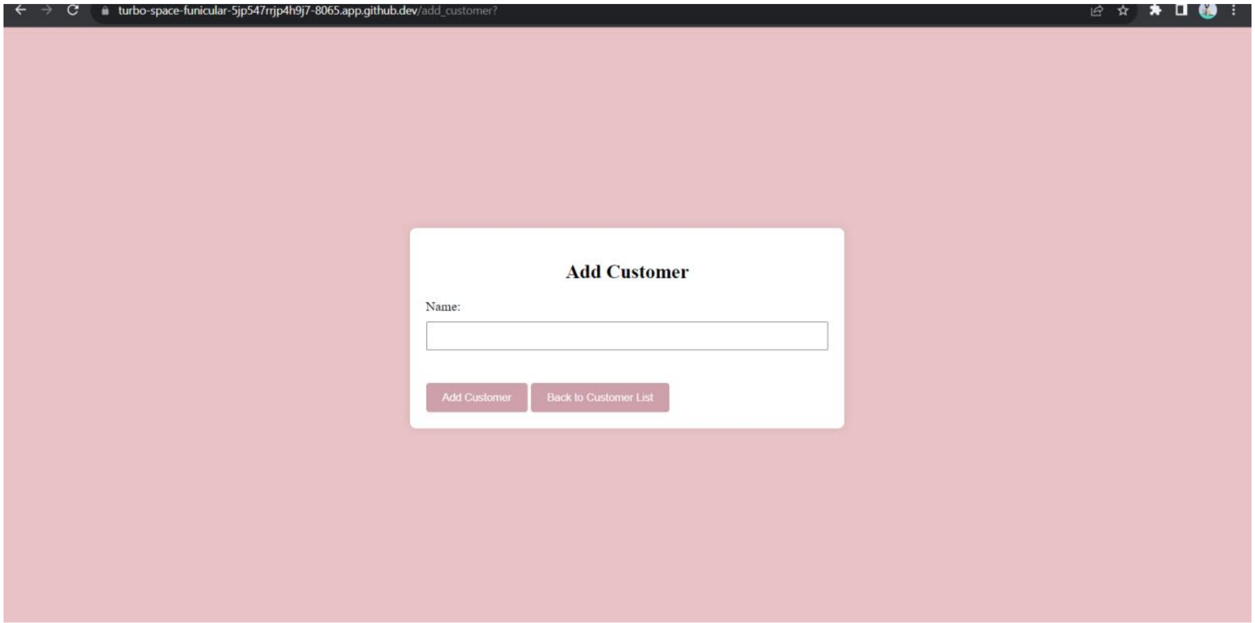
Customers:

ID	Name
4	Angel
6	Bob The Builder
7	Scoobie
8	Noddy
9	Oswald
10	Ben
11	Stormie
12	Jack & Jill

Orders:

ID	Customer ID	Product
2	71	Ice
3	90	Couch
4	23	House
5	1	hat
6	1	Thorns
7	2	Snow
9	30	oranges
10	35	Spider
11	4	Black berry mobile

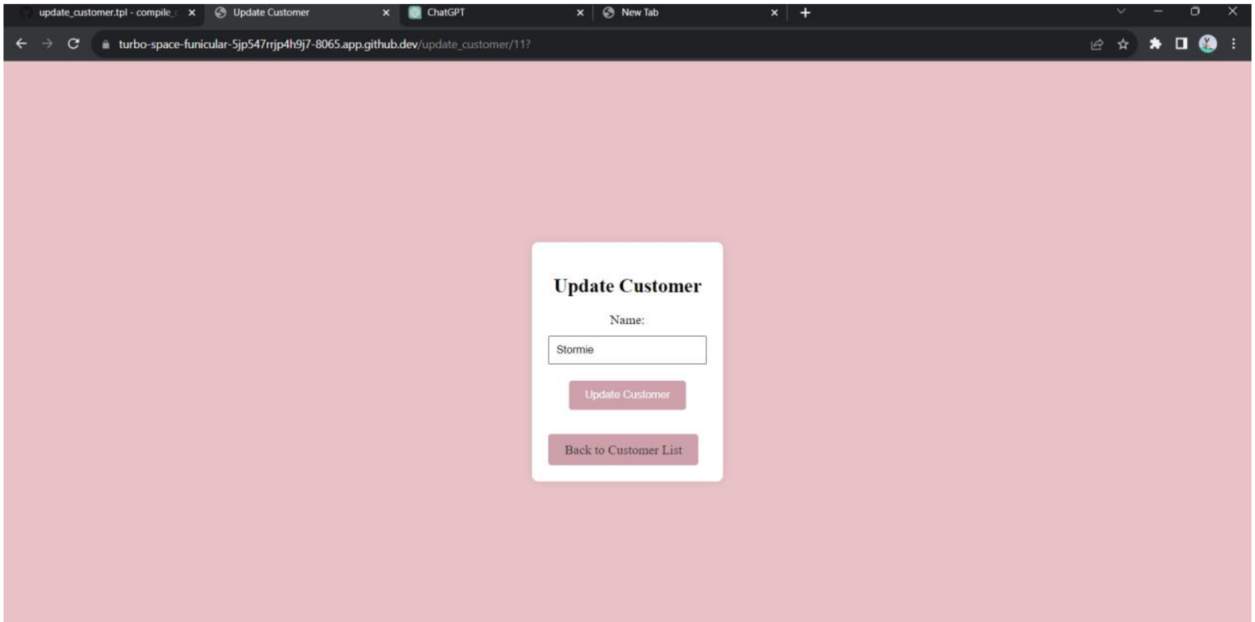
1(a) Add customer


turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/add_customer?
Add Customer
Back to Customer List

Add Customer

Name:

2(b) Update the customer

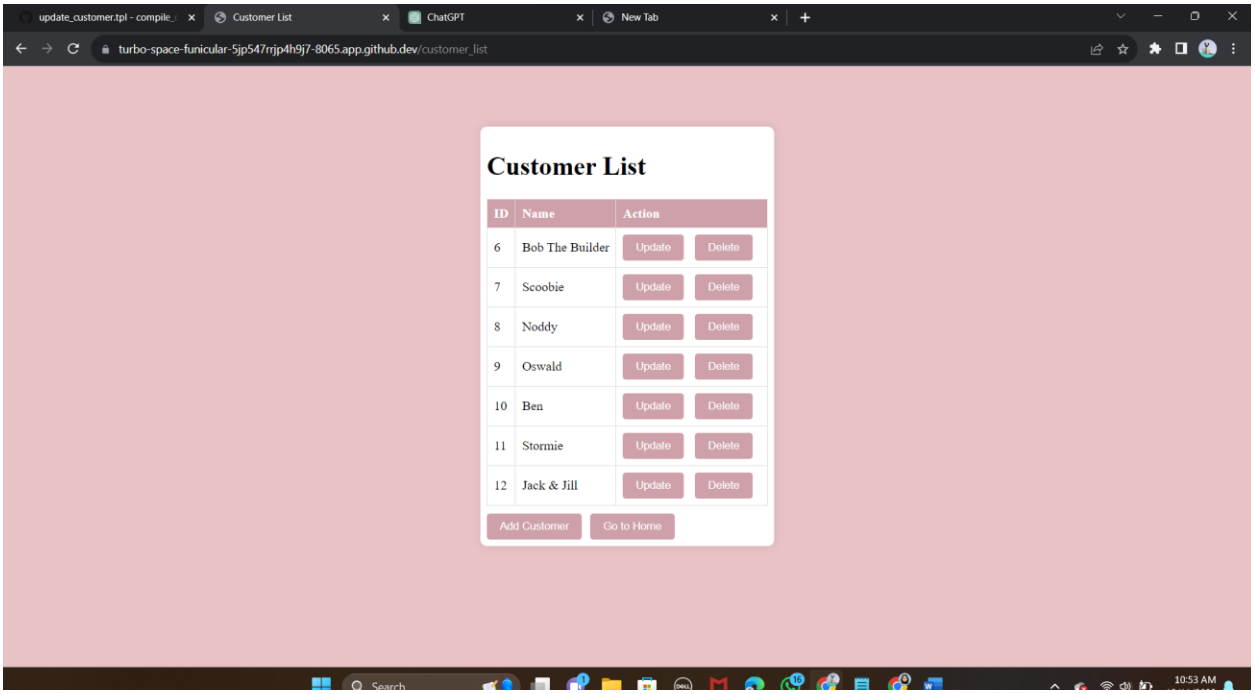

update_customer.tpl - compile_
Update Customer
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/update_customer/11?

Update Customer

Name: Stormie

Update Customer
Back to Customer List

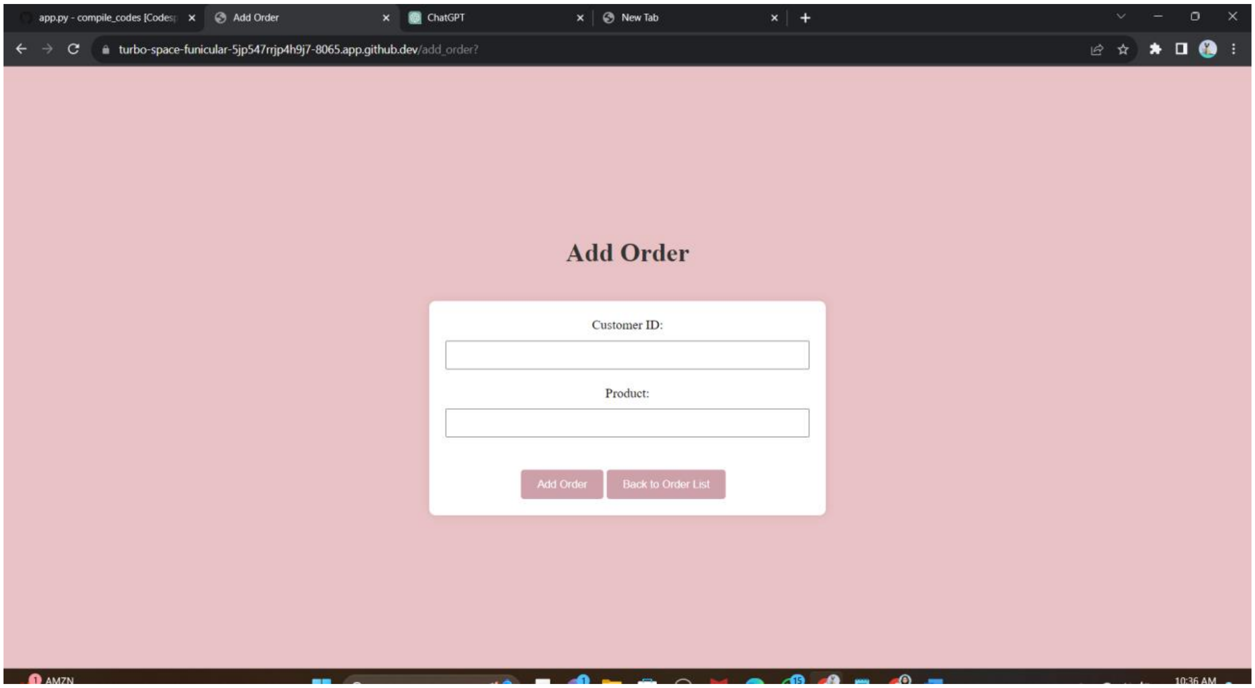
2 (c) Delete Customer – Deleted the customer Angel


update_customer.tpl - compile_
Customer List
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/customer_list
Search
10:53 AM

Customer List

ID	Name	Action	
6	Bob The Builder	Update	Delete
7	Scoobie	Update	Delete
8	Noddy	Update	Delete
9	Oswald	Update	Delete
10	Ben	Update	Delete
11	Stormie	Update	Delete
12	Jack & Jill	Update	Delete

2(a) Add Order

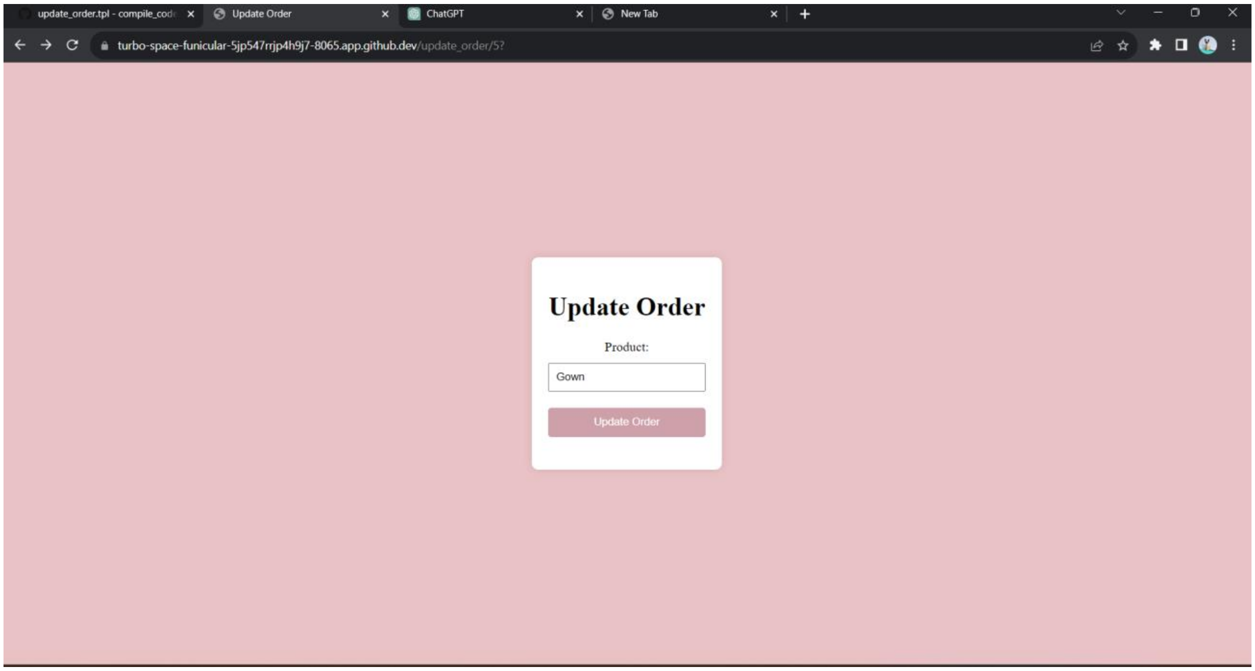

app.py - compile_codes [Codesp
Add Order
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/add_order?
Add Order
Back to Order List
AMZN
10:36 AM

Add Order

Customer ID:

Product:

2 (b) Update Order


update_order.tpl - compile_code
Update Order
ChatGPT
New Tab
turbo-space-funicular-5jp547rrjp4h9j7-8065.app.github.dev/update_order/5?
Update Order

Update Order

Product: Gown