|  |  |  |
| --- | --- | --- |
| Icon  Description automatically generated | **CMPS 350 Project Phase 2 – UI Design Implementation (10% of the course grade)** | |
| **Group Id:** | | **t6-ahmed-hani-syed** |
| **Group Members:** | | Ahmed Abdelhamid (201907476)  Hani Jassin Jafer (201908619)  Syed Zubair (201708606)  **Emails:** aa1907476@student.qu.edu.qa; hj1908619@student.qu.edu.qa; sz1708606@student.qu.edu.qa; |

**Grading Rubric - In the Functionality column please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Criteria | | % | Functionality\* | Quality of the implementation | Your Grade |
| **1)** Design and implementation of the database schema to manage the data in a MongoDB database (including a schema diagram in the project report). | | 10% | Working (completed 100%) |  |  |
| **2)** Populate the database with the data from the json files. | | 5% | Working (completed 100%) |  |  |
| **3) Implementation:**  - Repository Implementation to read/write data from/to MongoDB (30%)  - Web API implementation (25%)  - Change Web UI to use the Web API (20%) | | **75%** | Working (completed 100%) |  |  |
|  | U1 – Login | 5 | Working (completed 100%) |  |  |
| U2 - YalaPay Dashboard | 10 | Working (completed 100%) |  |  |
| U3 - List/Search/Add/ Update and Delete a Customer:  1. List Customers (5 pts)  2. Search Customers (5 pts)  3. Add Customer (5 pts)  4. Update Customer (5 pts)  5. Delete Customer (5 pts) | 25 | Working (completed 100%) |  |  |
| U4 - List/Search/Add/ Update and Delete an Invoice  1. List Invoices (5 pts)  2. Search Invoices (5 pts)  3. Add Invoice (5 pts)  4. Update Invoice (5 pts)  5. Delete Invoice (5 pts) | 25 | Working (completed 100%) |  |  |
| U5 - List/Search/Add/ Update and Delete payments for an invoice  1. List Payments (5 pts)  2. Search Payments (5 pts)  3. Add Payment (5 pts)  4. Update Payment (5 pts)  5. Delete Payment (5 pts) | 25 | Working (completed 100%) |  |  |
| U6 - Manage Cashing Cheques  1. Add Cheques Deposit (10 pts)  2. List Cheque Deposits (5 pts)  3. Update Cheques Deposit (15 pts) | 30 | Working (completed 100%) |  |  |
| U7 - Invoices Report | 10 | Working (completed 100%) |  |  |
| U8 - Cheques Report | 10 | Working (completed 100%) |  |  |
| **4) Design and Testing Documentation**  **\* Design documentation:**  - Web API services diagram  - Database schema diagram  **\* Testing documentation:** with evidence of working implementation using snapshots illustrating the results of your solution testing.  \* **Discussion of the project contribution** of each team member [-10pts if not done] | | 10% | Working (completed 100%) |  |  |
| **Total** | | 100 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation | | -  100 |  |  |  |

**\* Possible grading for functionality** - ***Working*** (get 70% of the assigned grade), ***Not*** ***working*** (lose 40% of assigned grade and ***Not done*** (get 0). The remaining grade is assigned to the quality of the implementation.

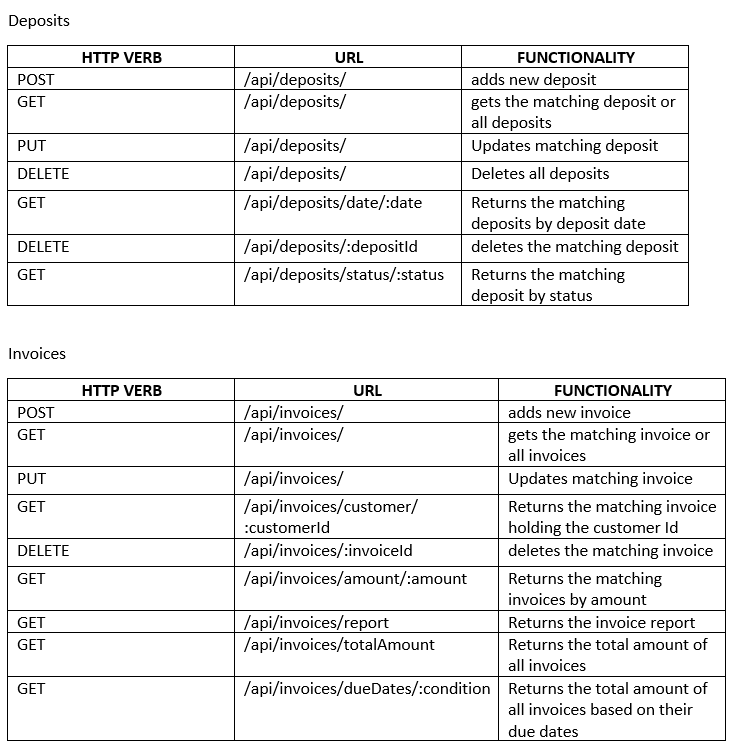
In case your implementation is not working then 40% of the grade will be lost and the remaining 60% will be determined based on of the code quality and how close your solution to the working implementation.

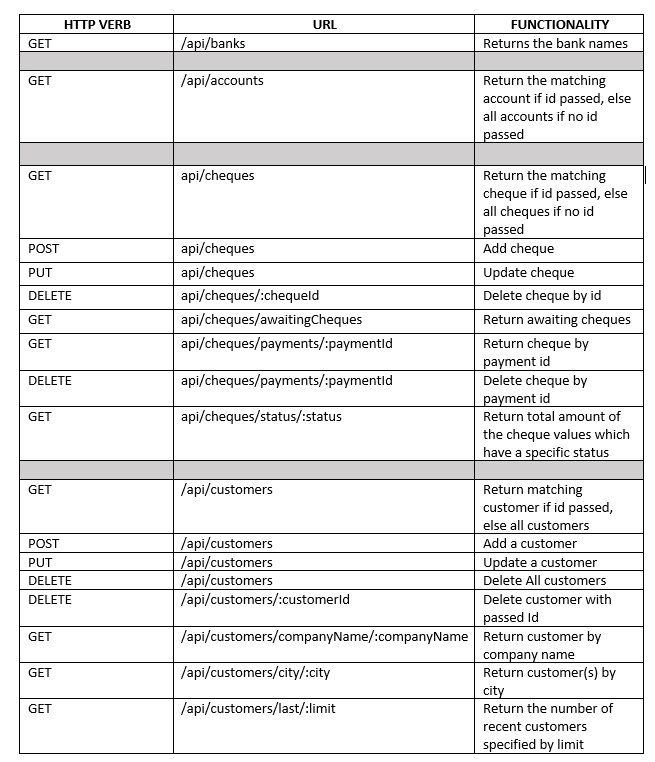
Solution quality also includes meaningful naming of identifiers (according to Android naming conventions), no redundant code, simple and efficient design, clean implementation without unnecessary files/code, use of comments where necessary, proper code formatting and indentation.

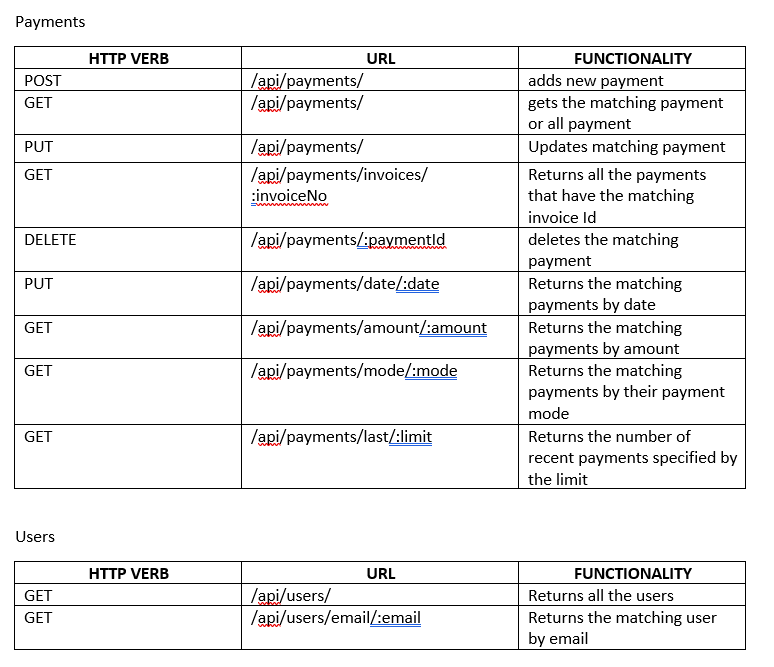
**Marks will be reduced** forcode duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

# Application Design

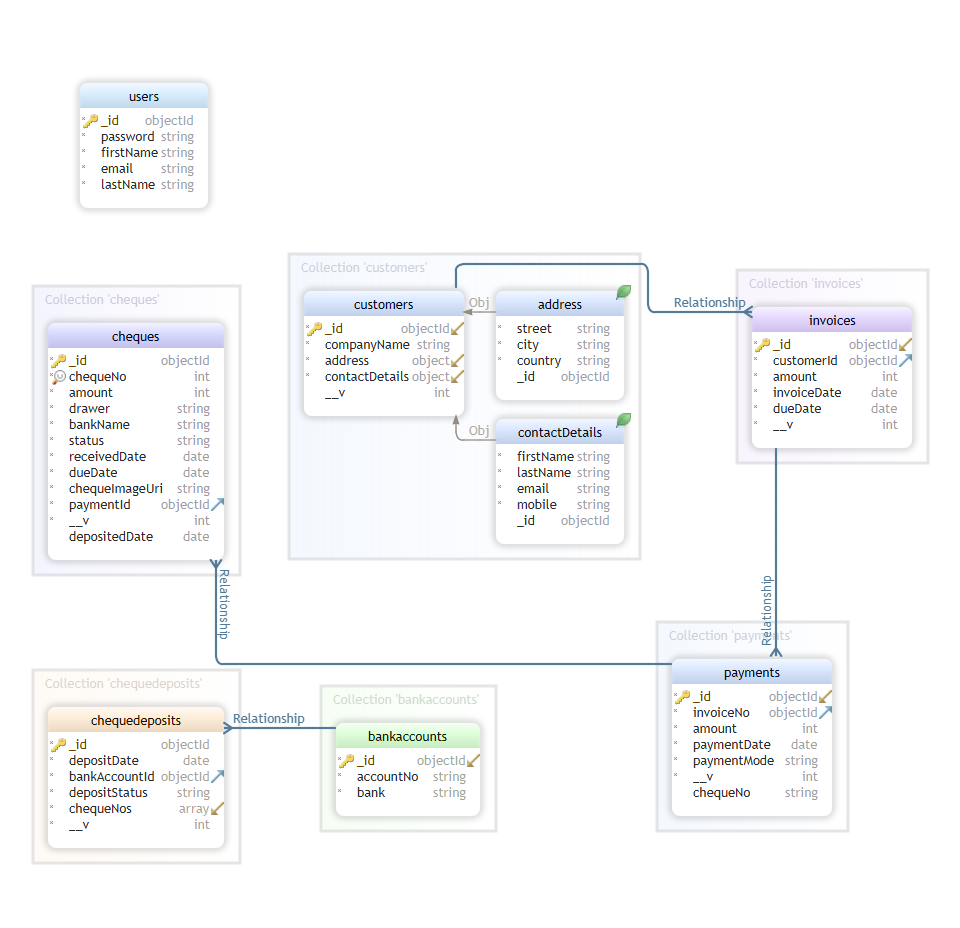
# Web API services diagram





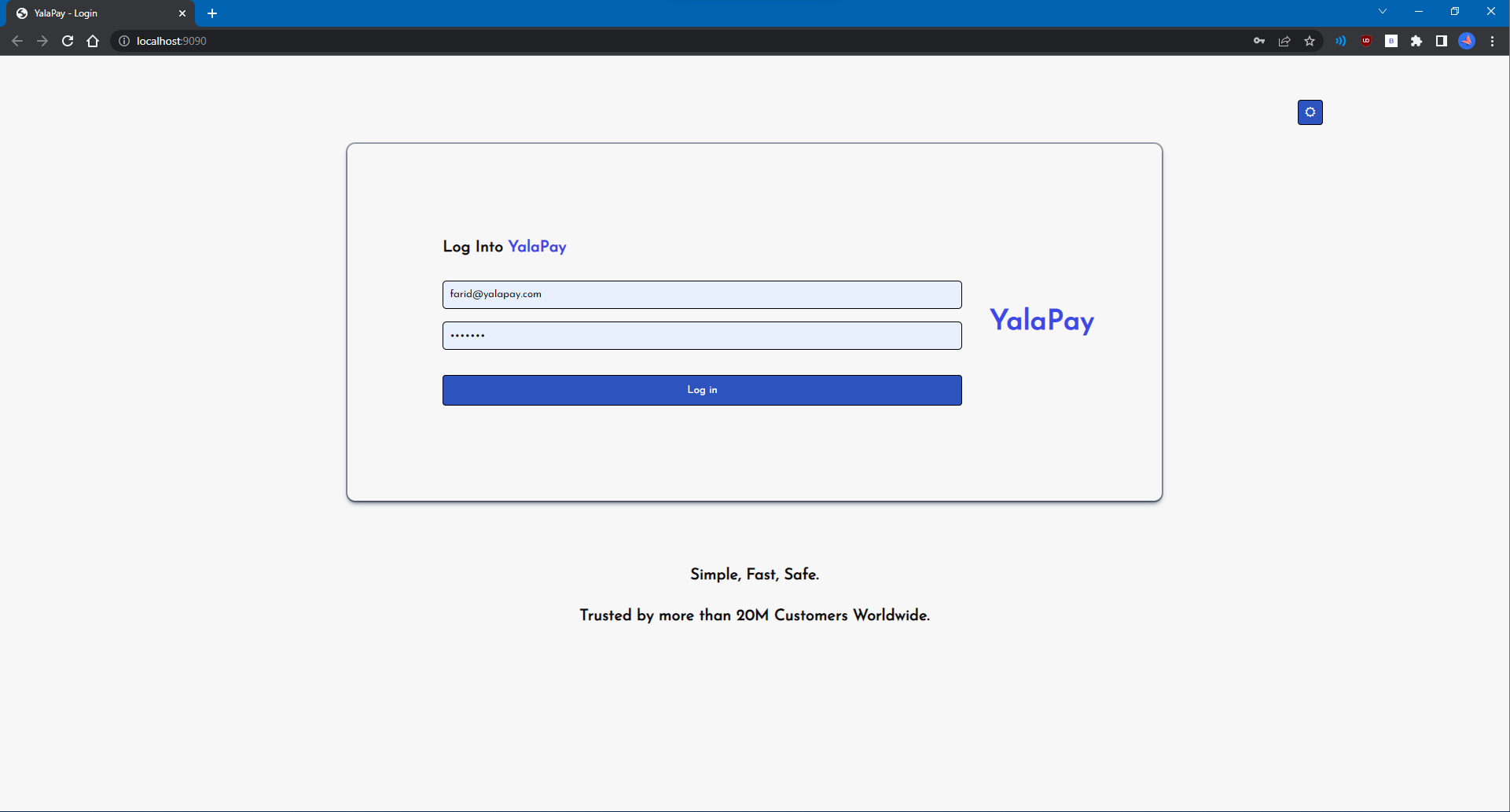


# Database schema diagram

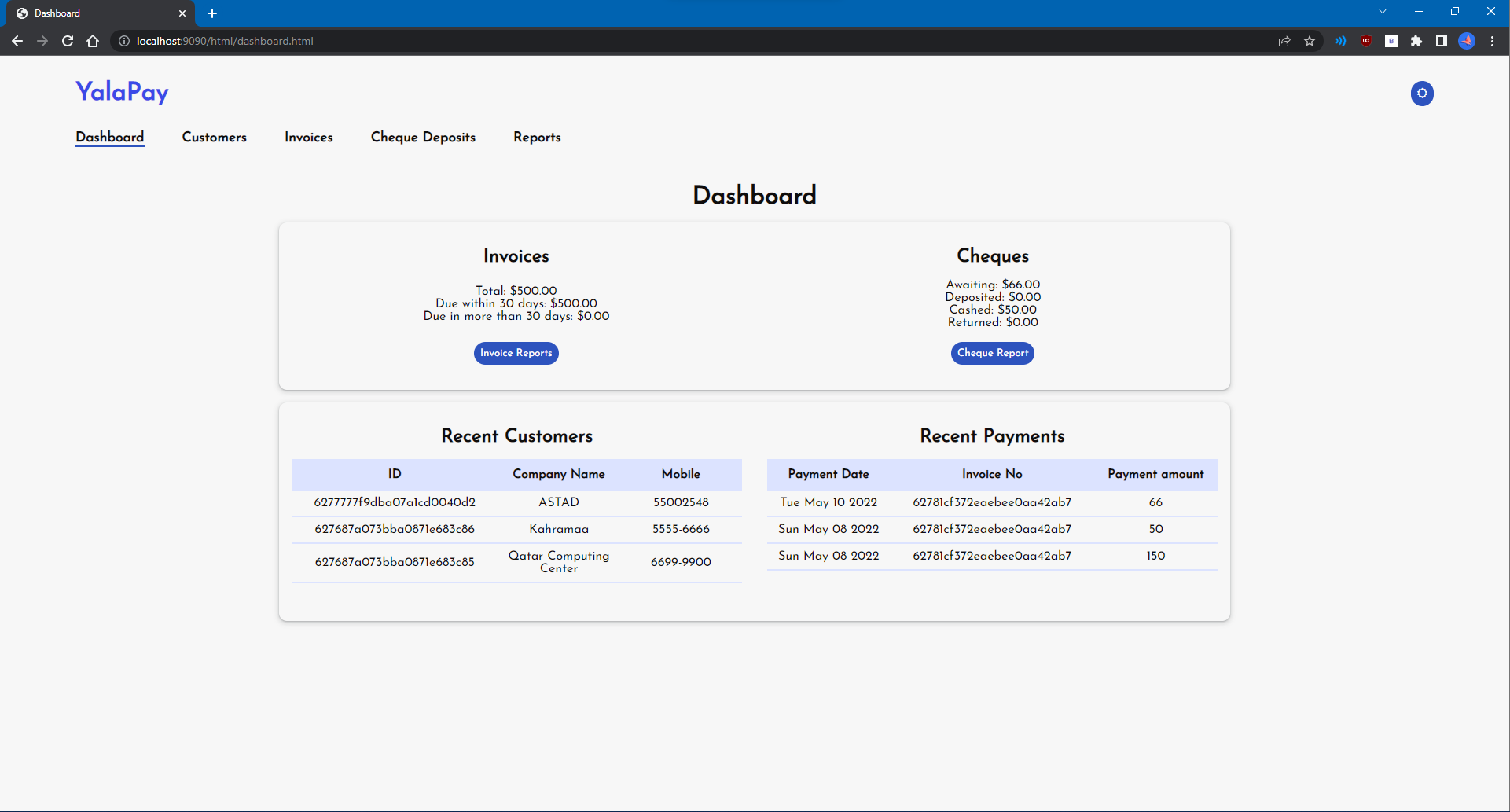


# Testing

# U1 – Login

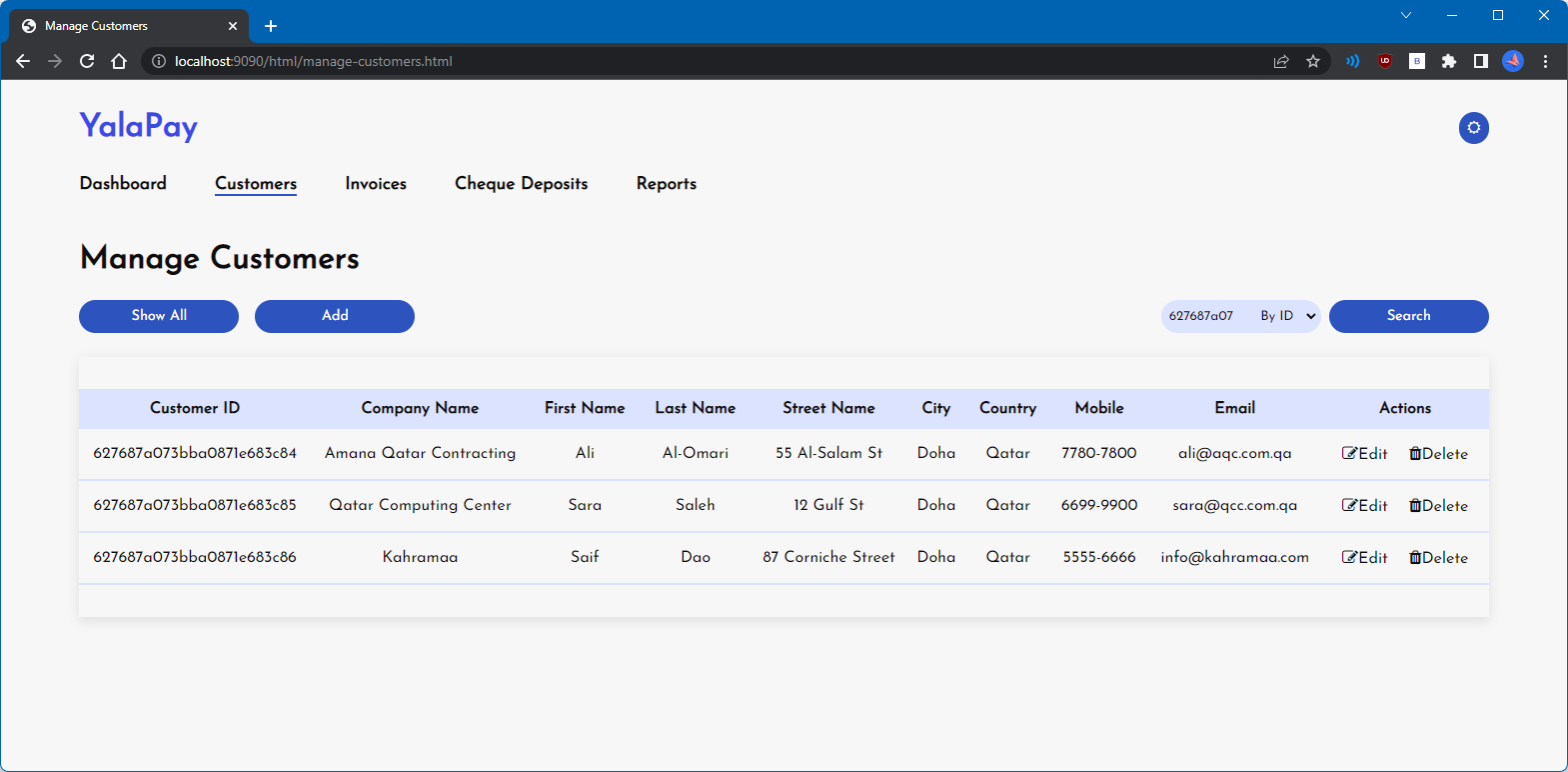


# U2 - YalaPay Dashboard

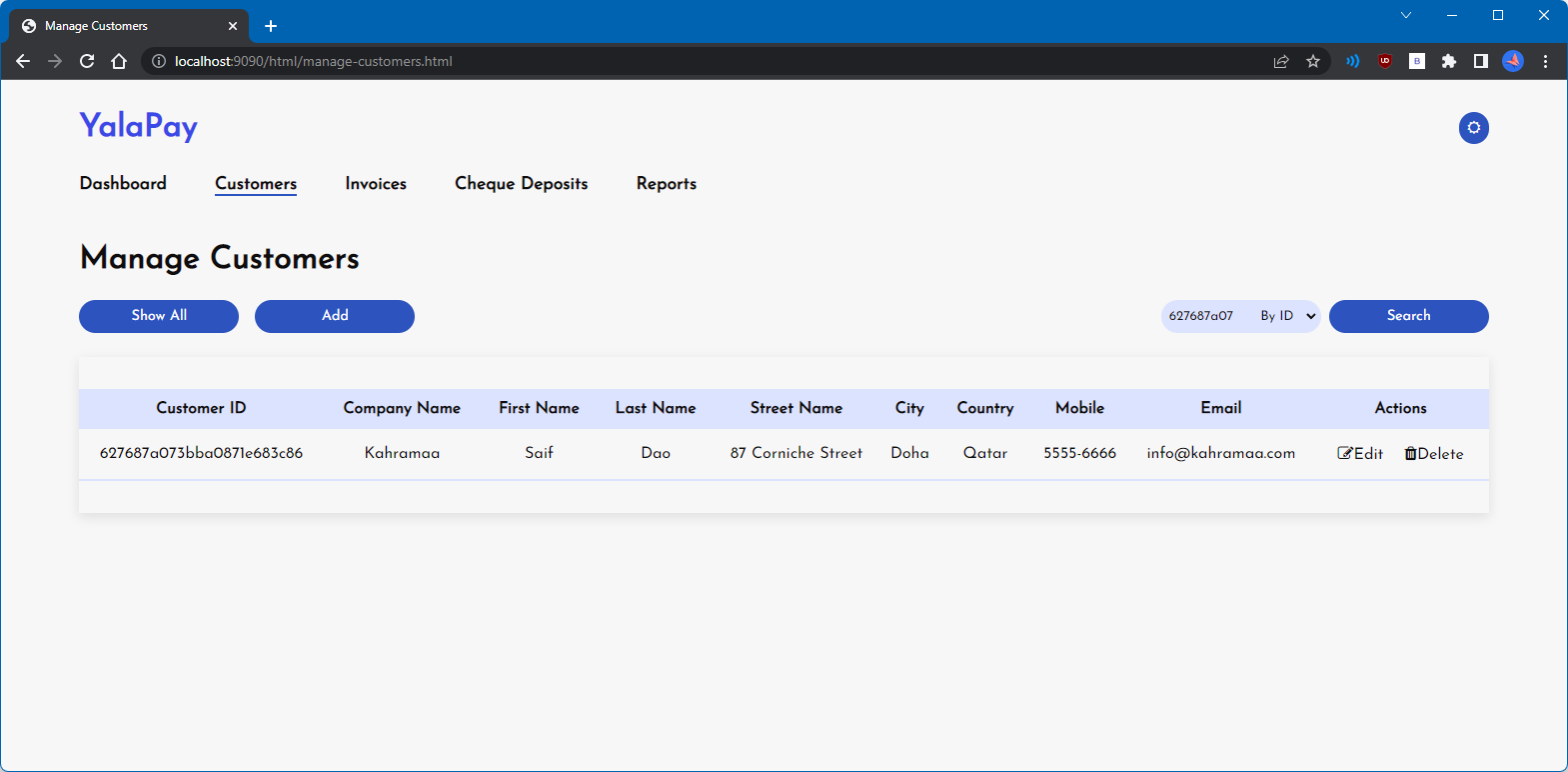


# U3 - List/Search/Add/ Update and Delete a Customer

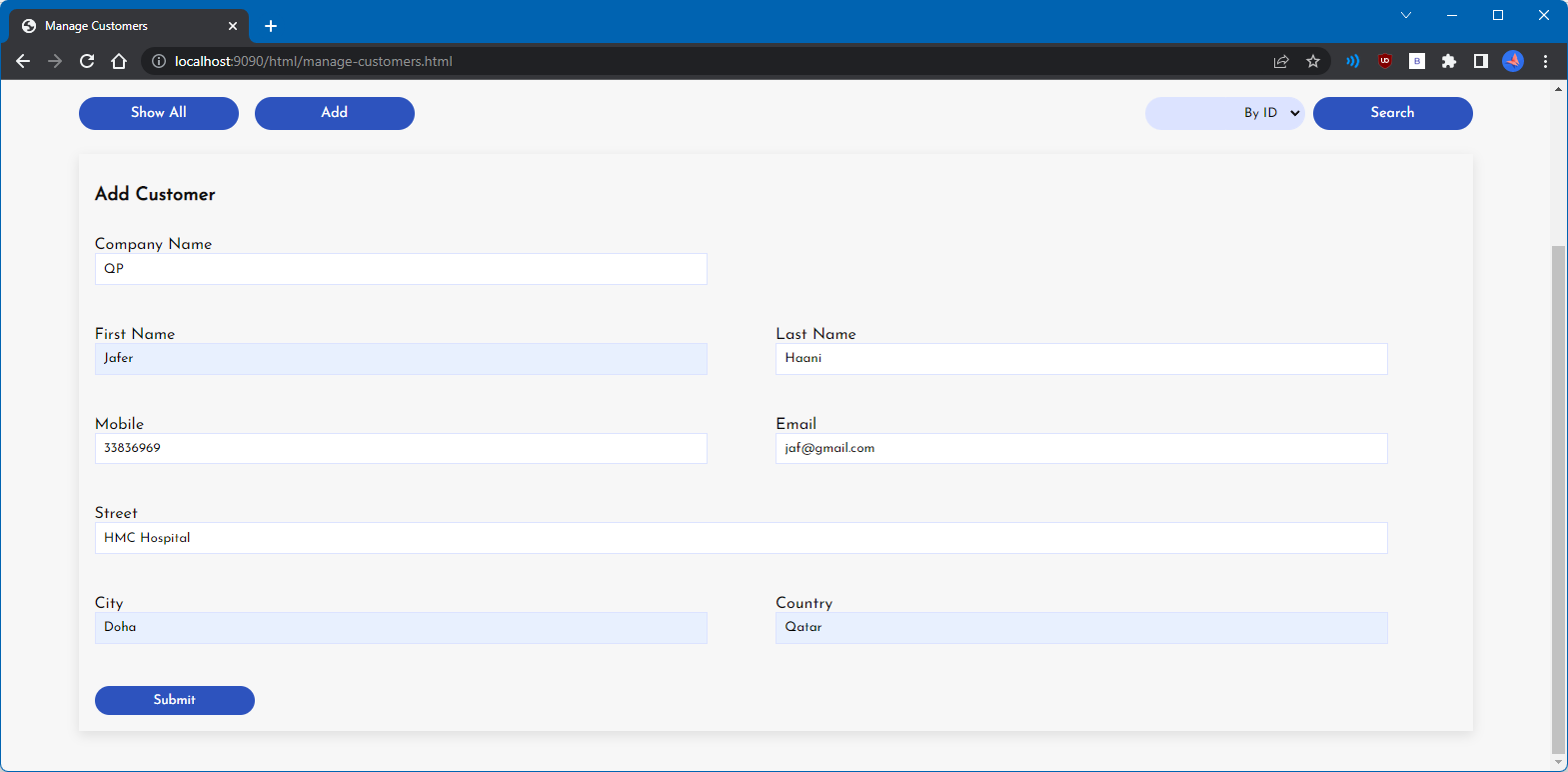
# List Customers

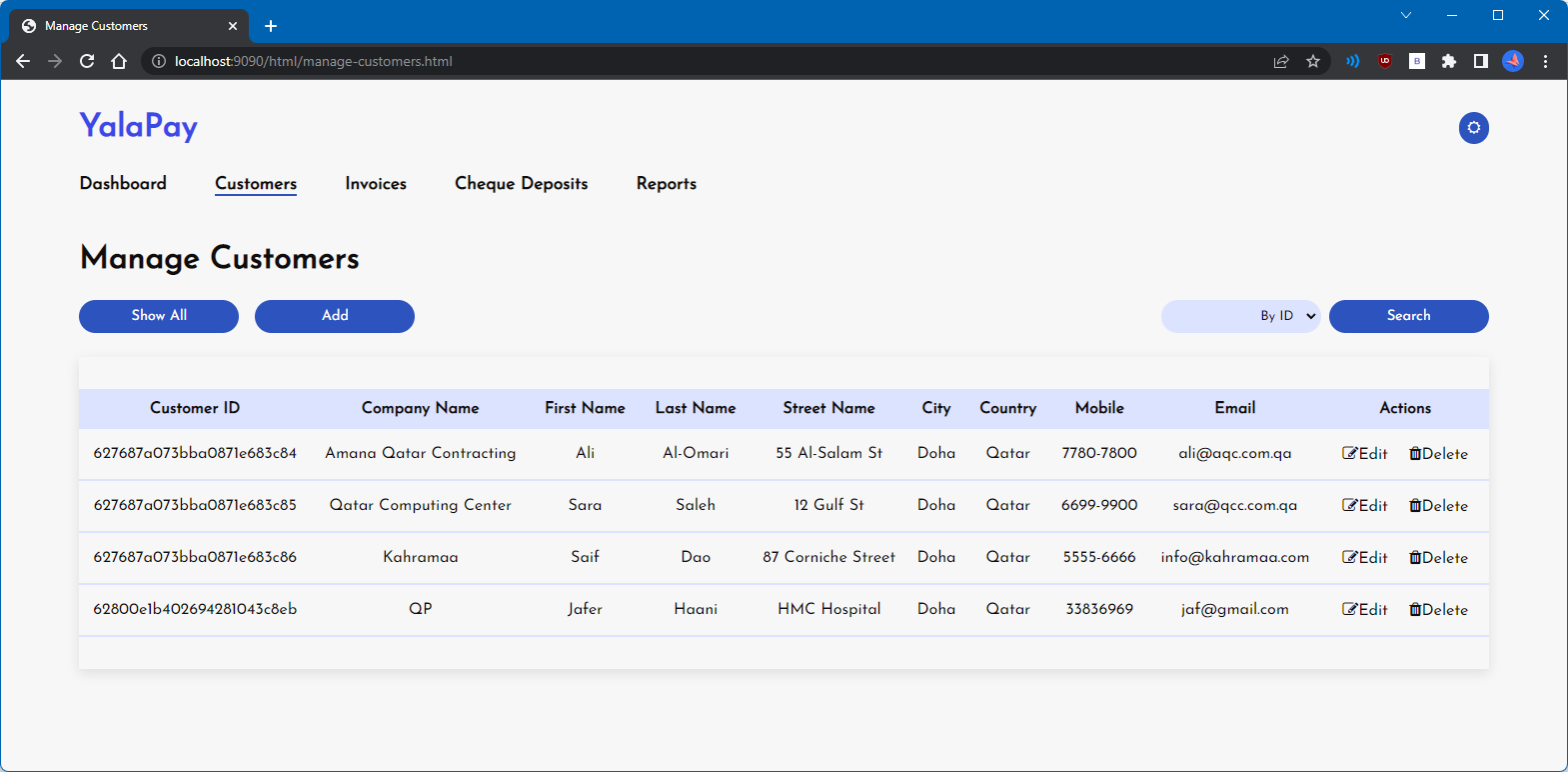


# Search Customers

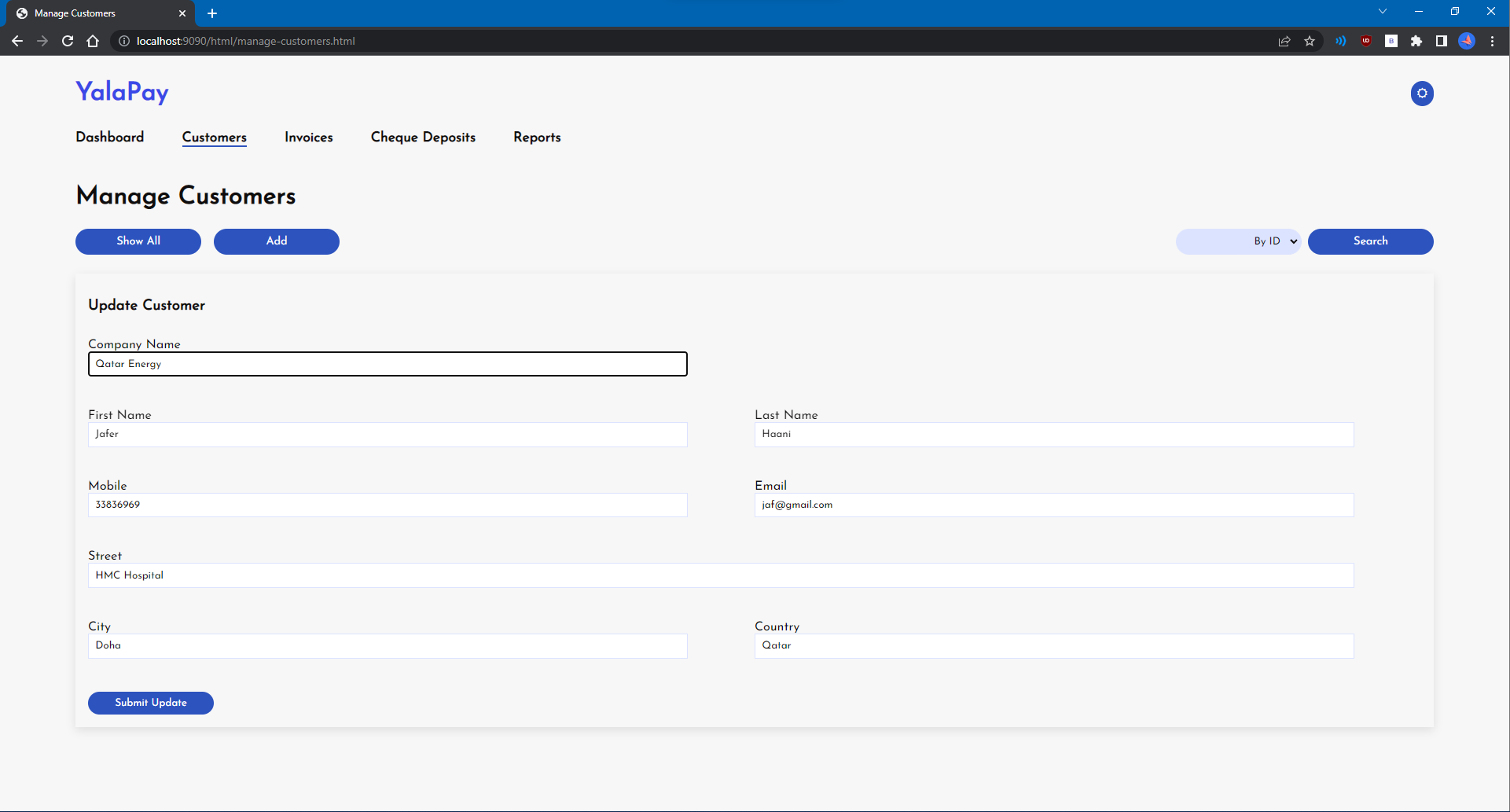


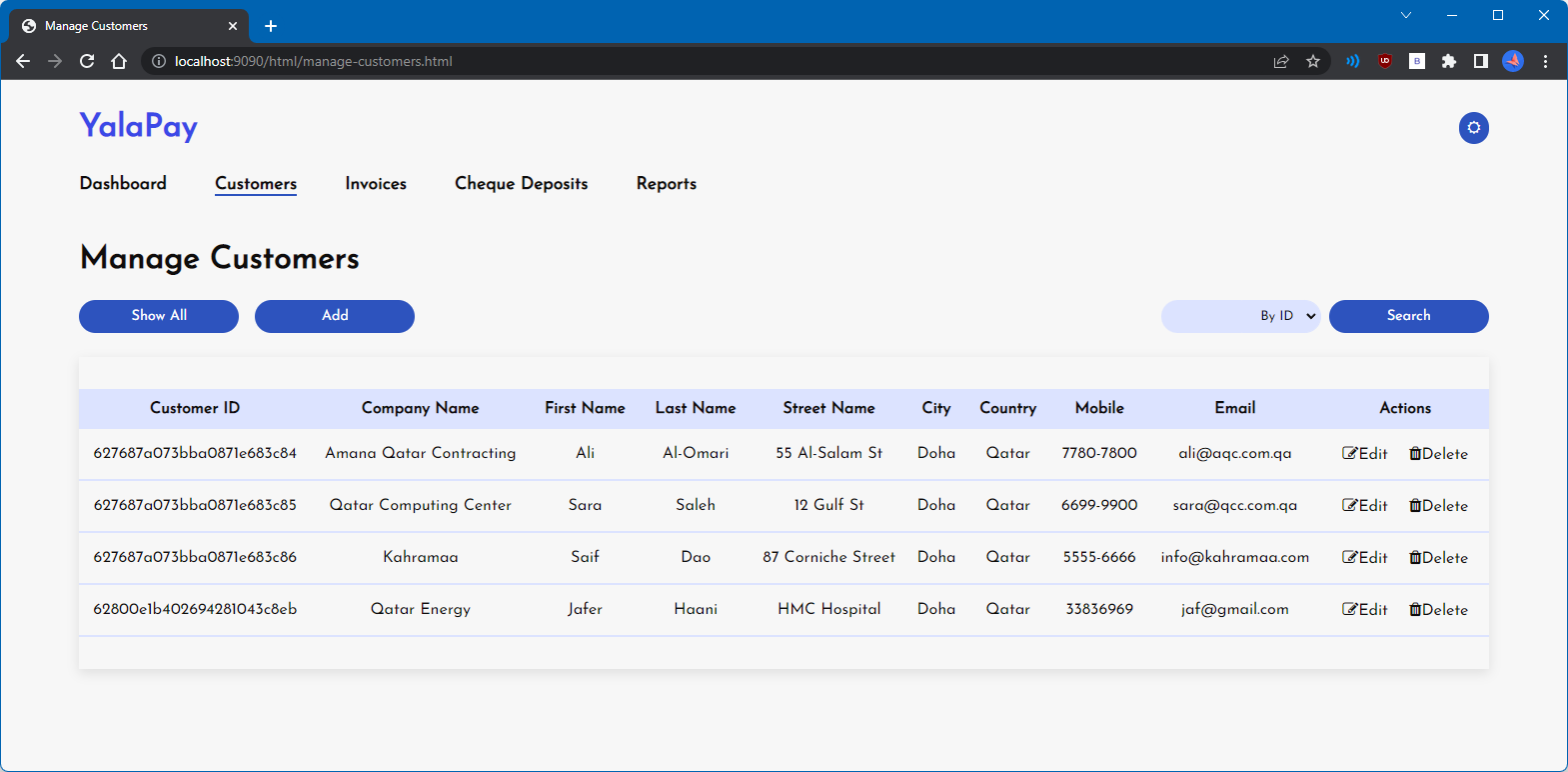
# Add Customer



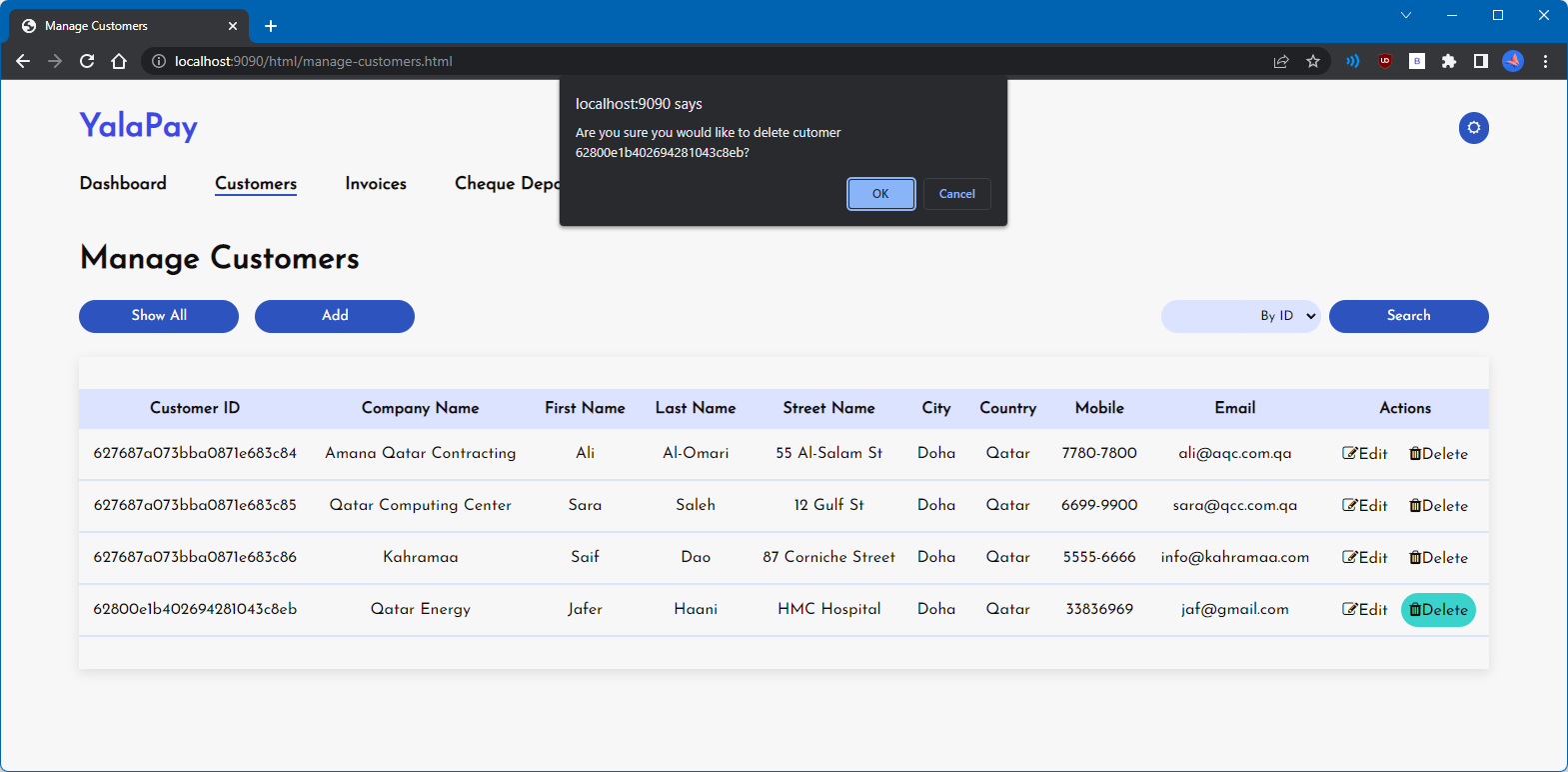


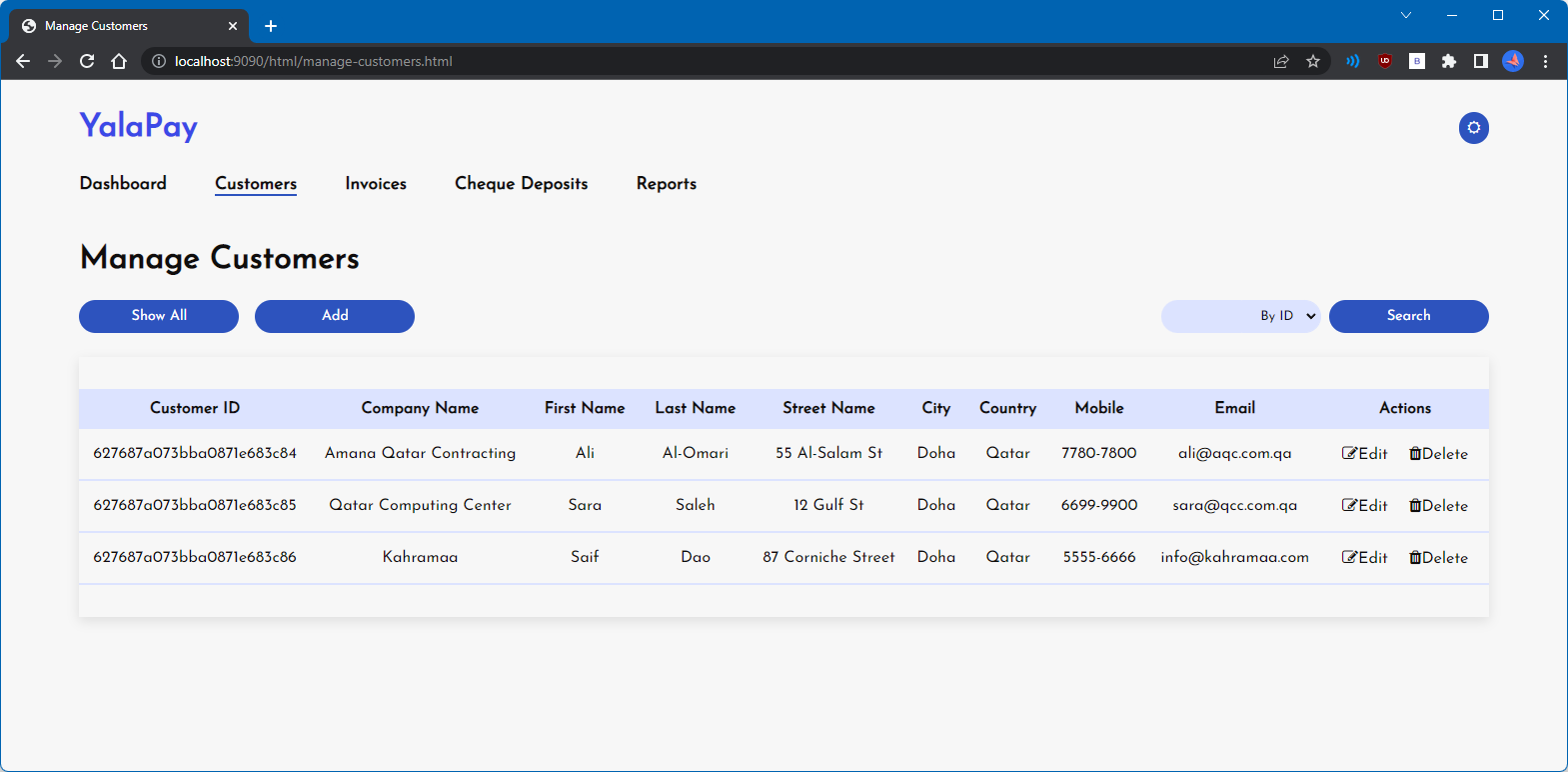
# Update Customer





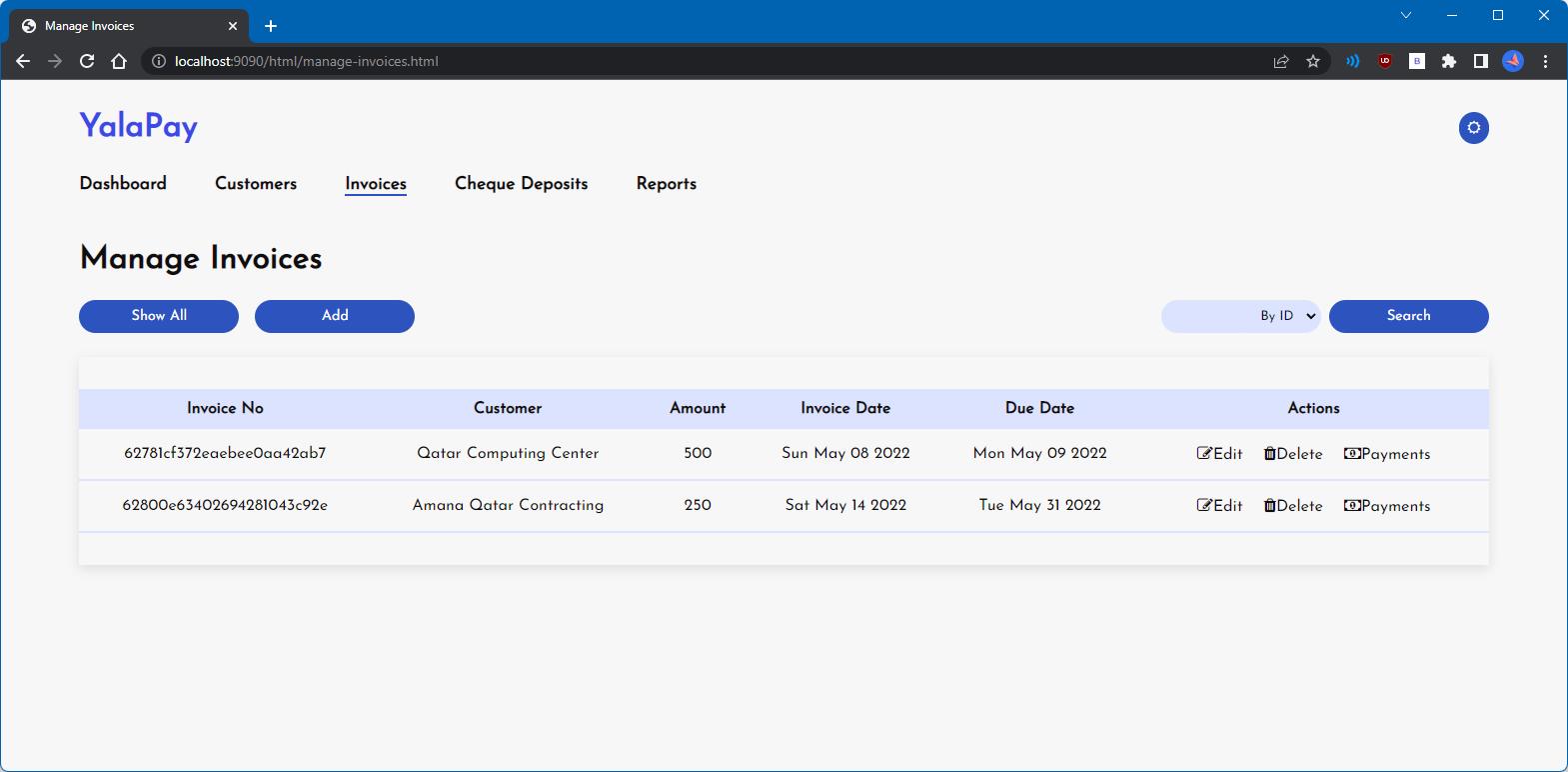
# Delete Customer



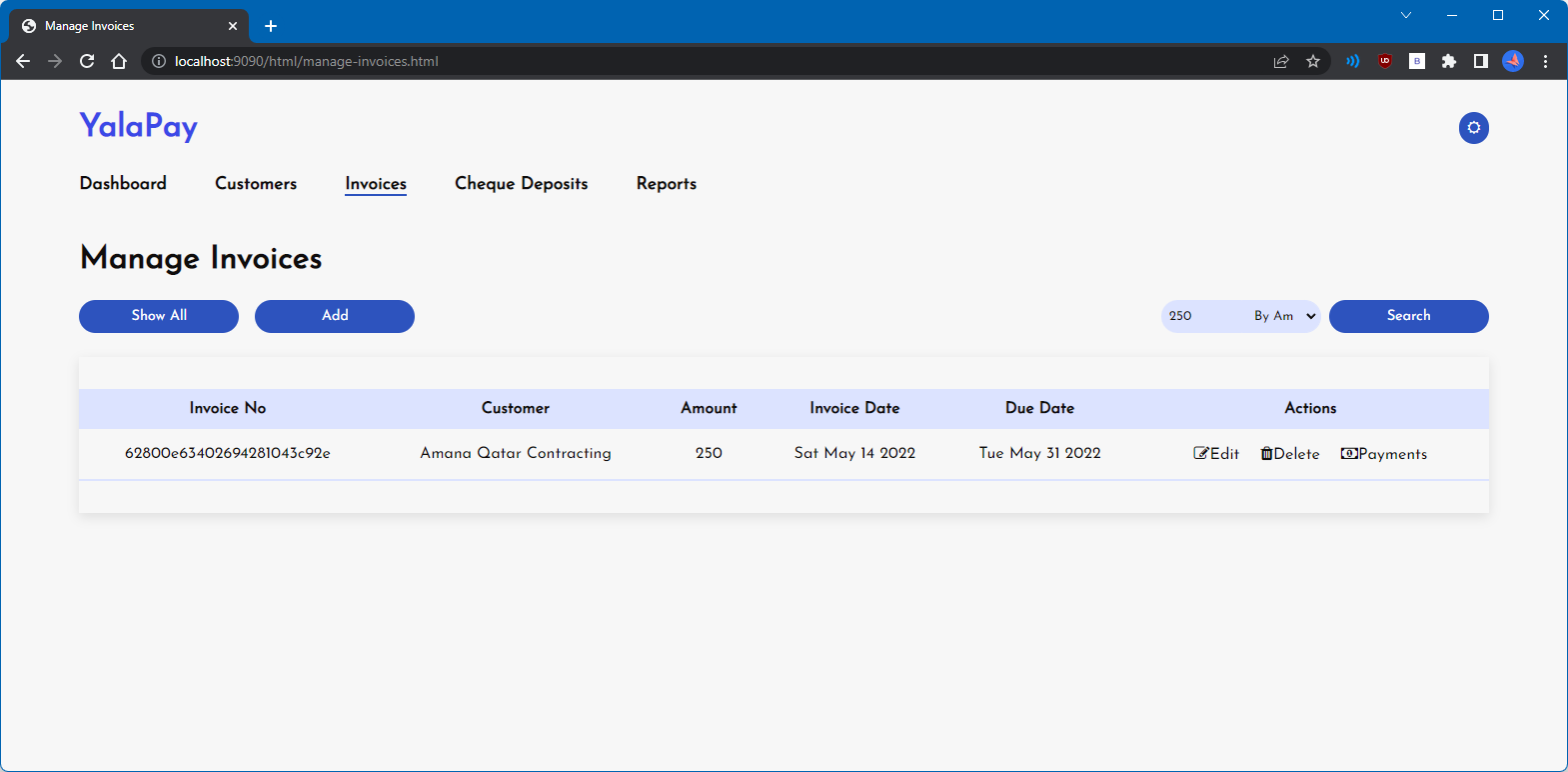


# U4 - List/Search/Add/ Update and Delete an Invoice

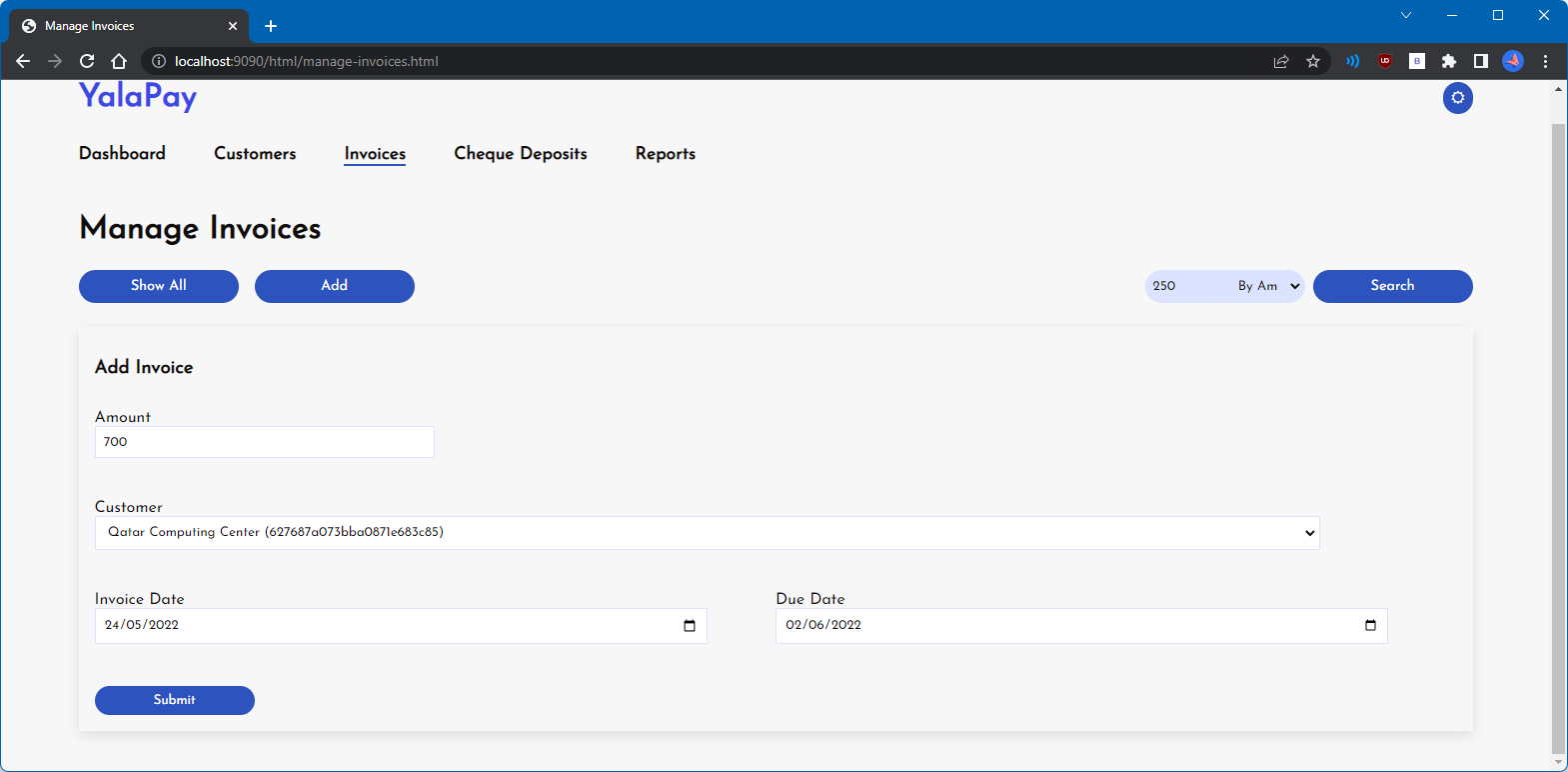
# List Invoices

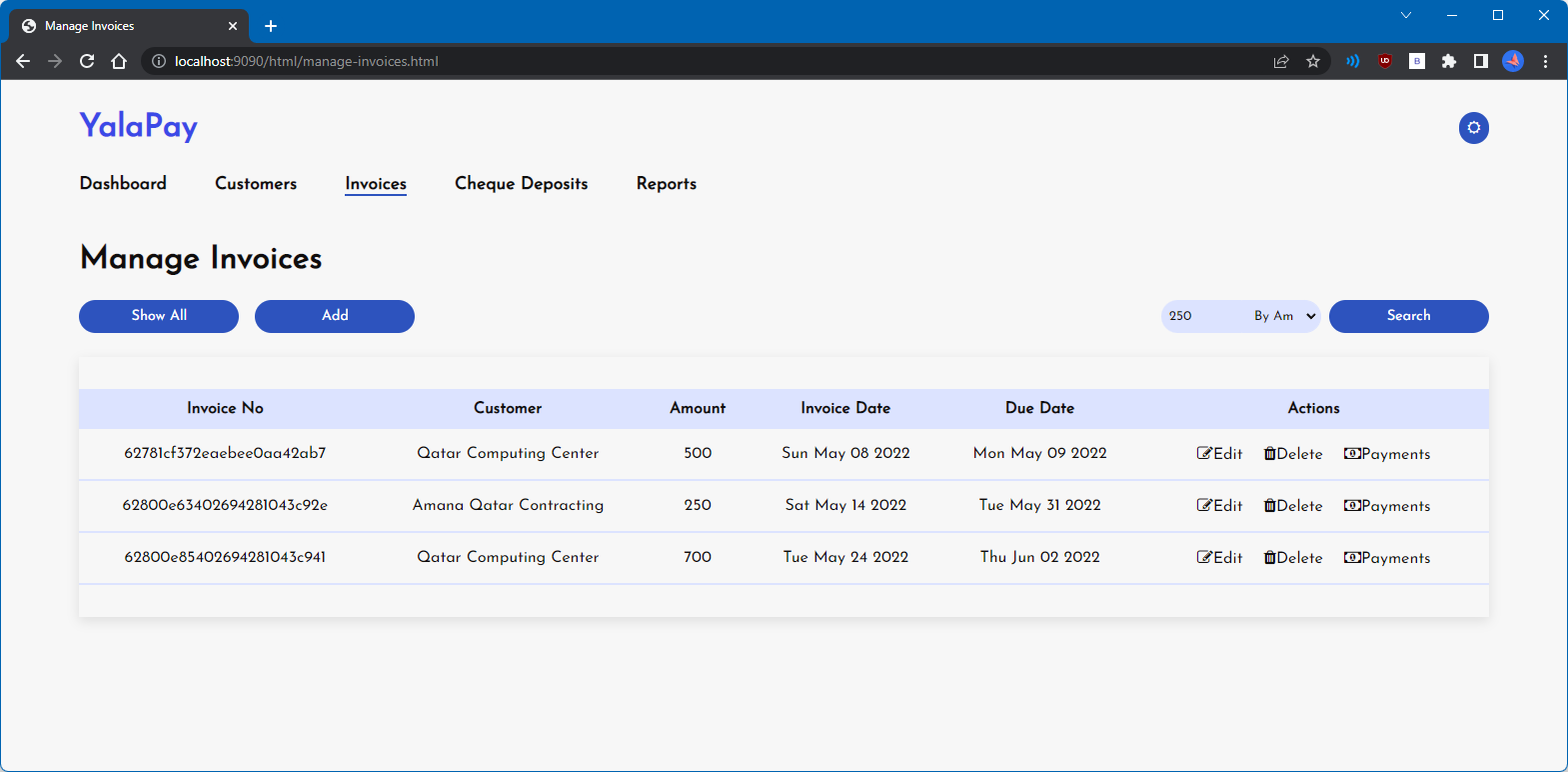


# Search Invoices

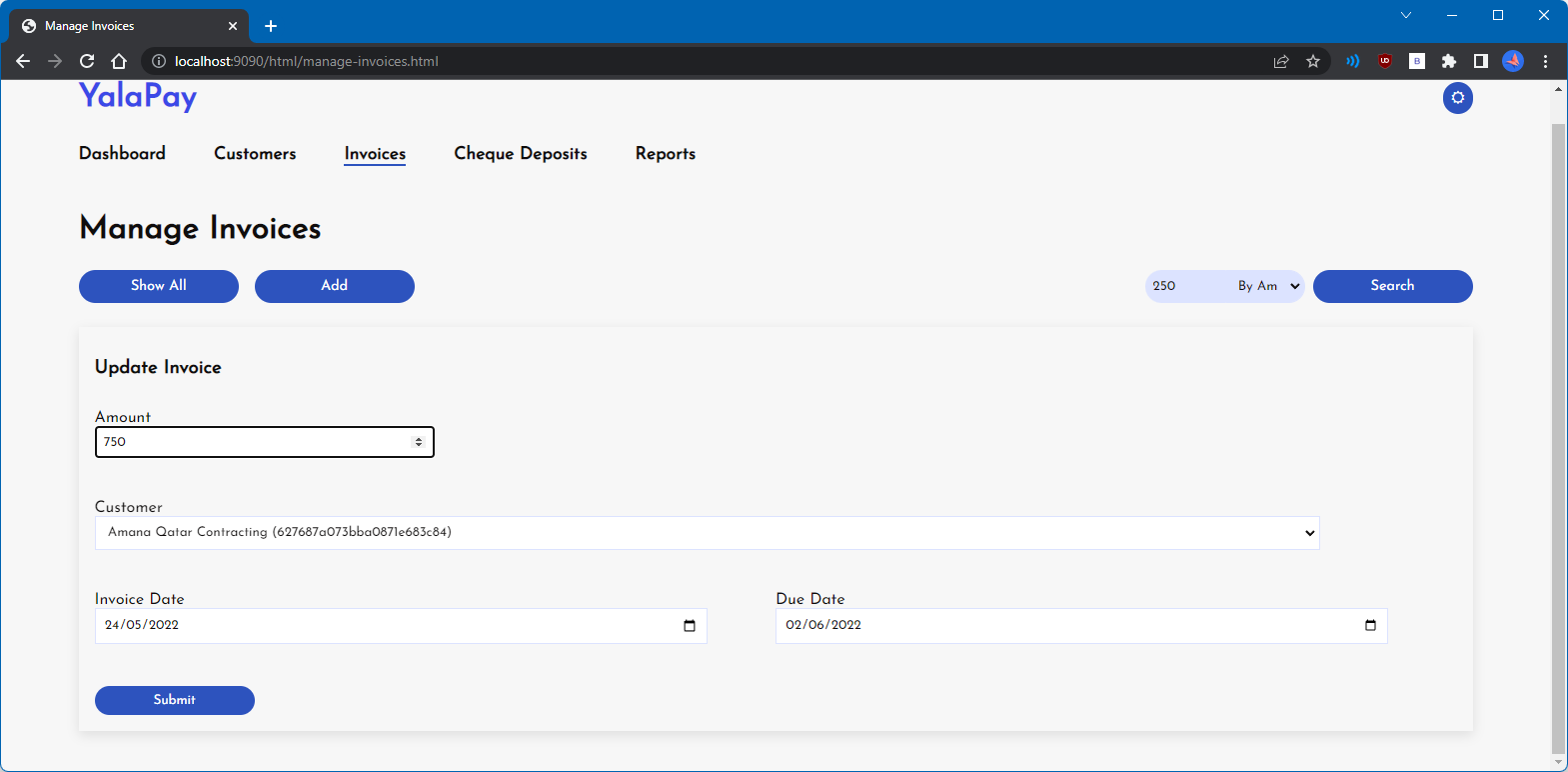


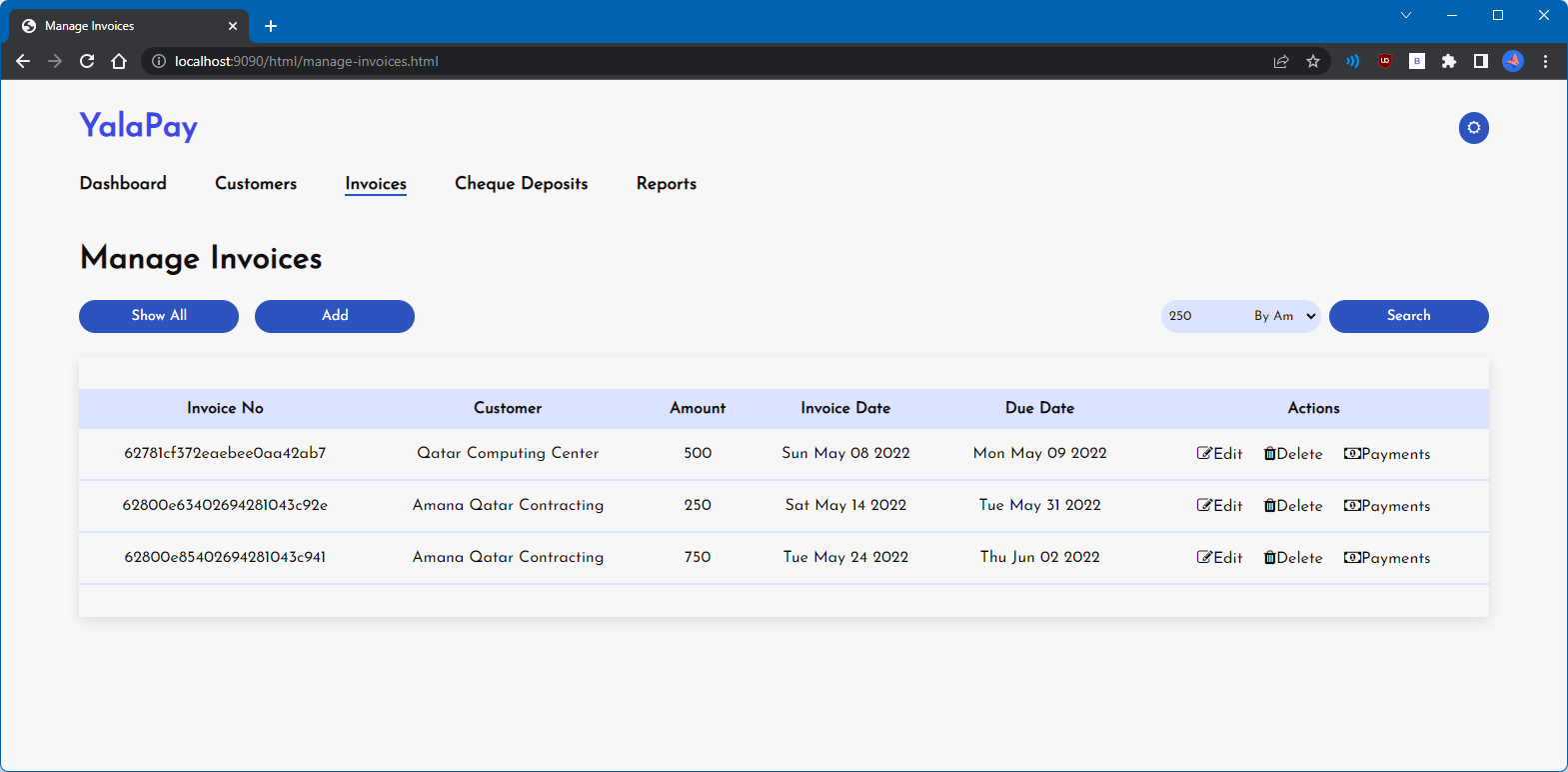
# Add Invoice



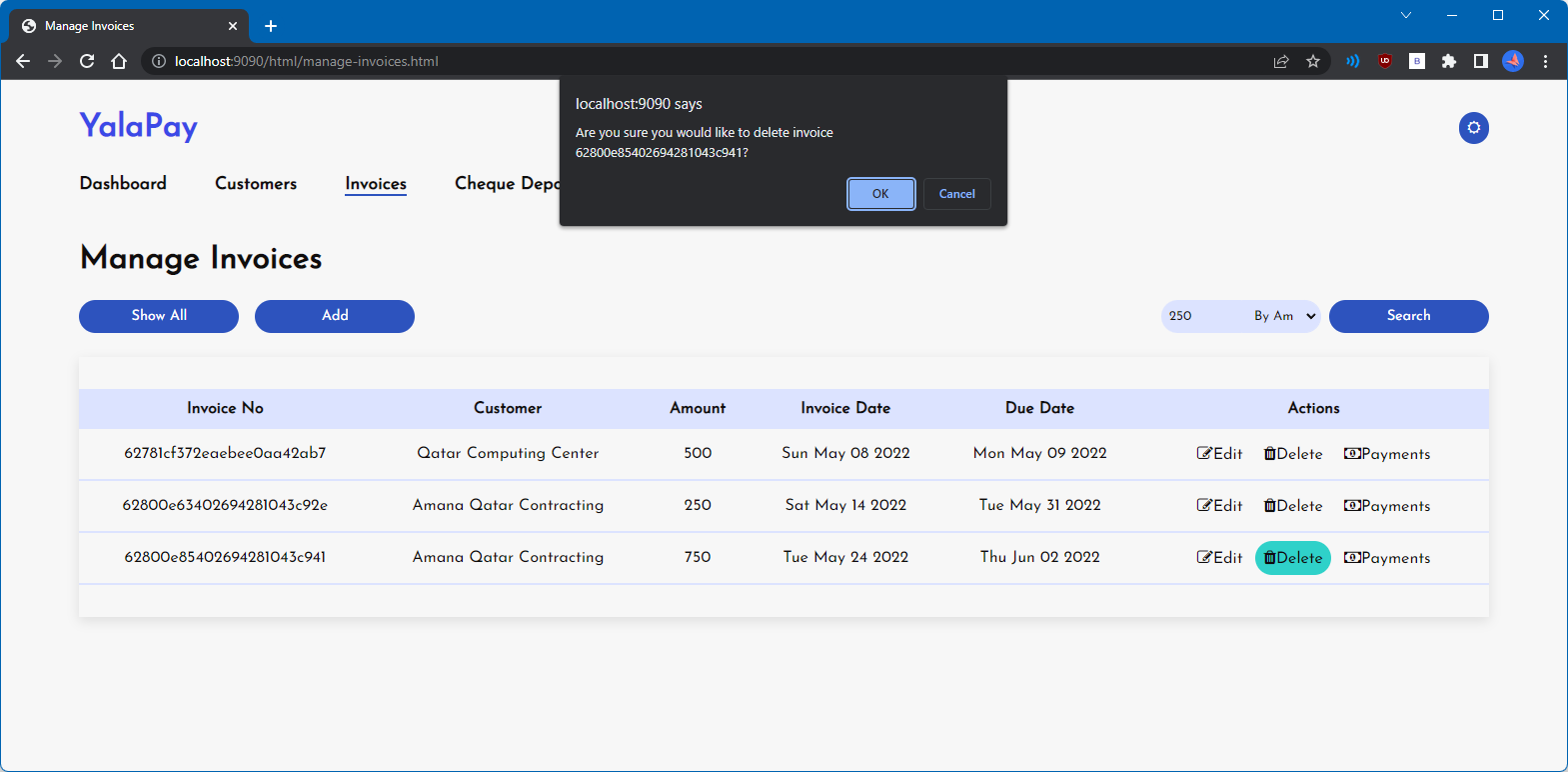


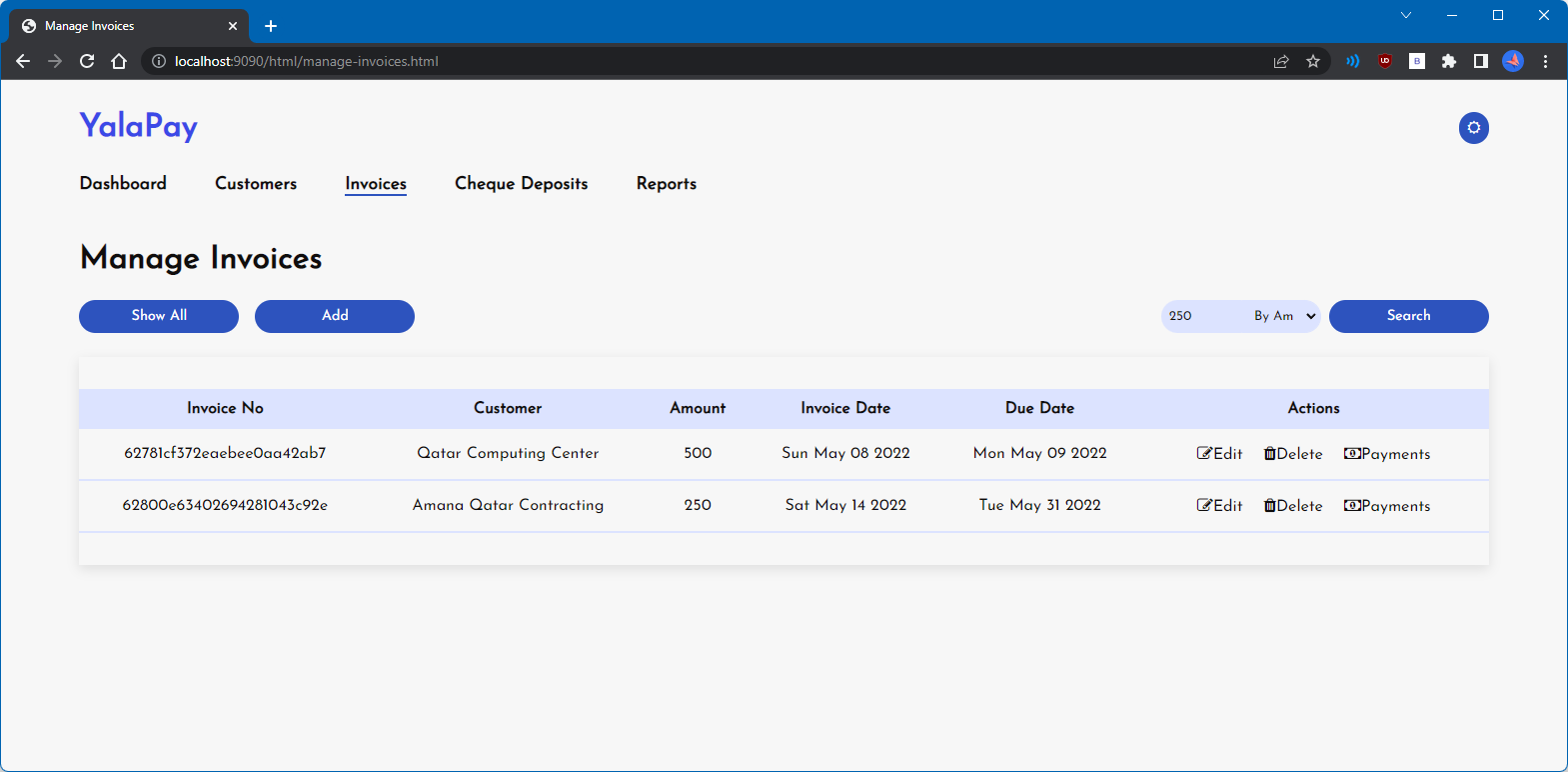
# Update Invoice





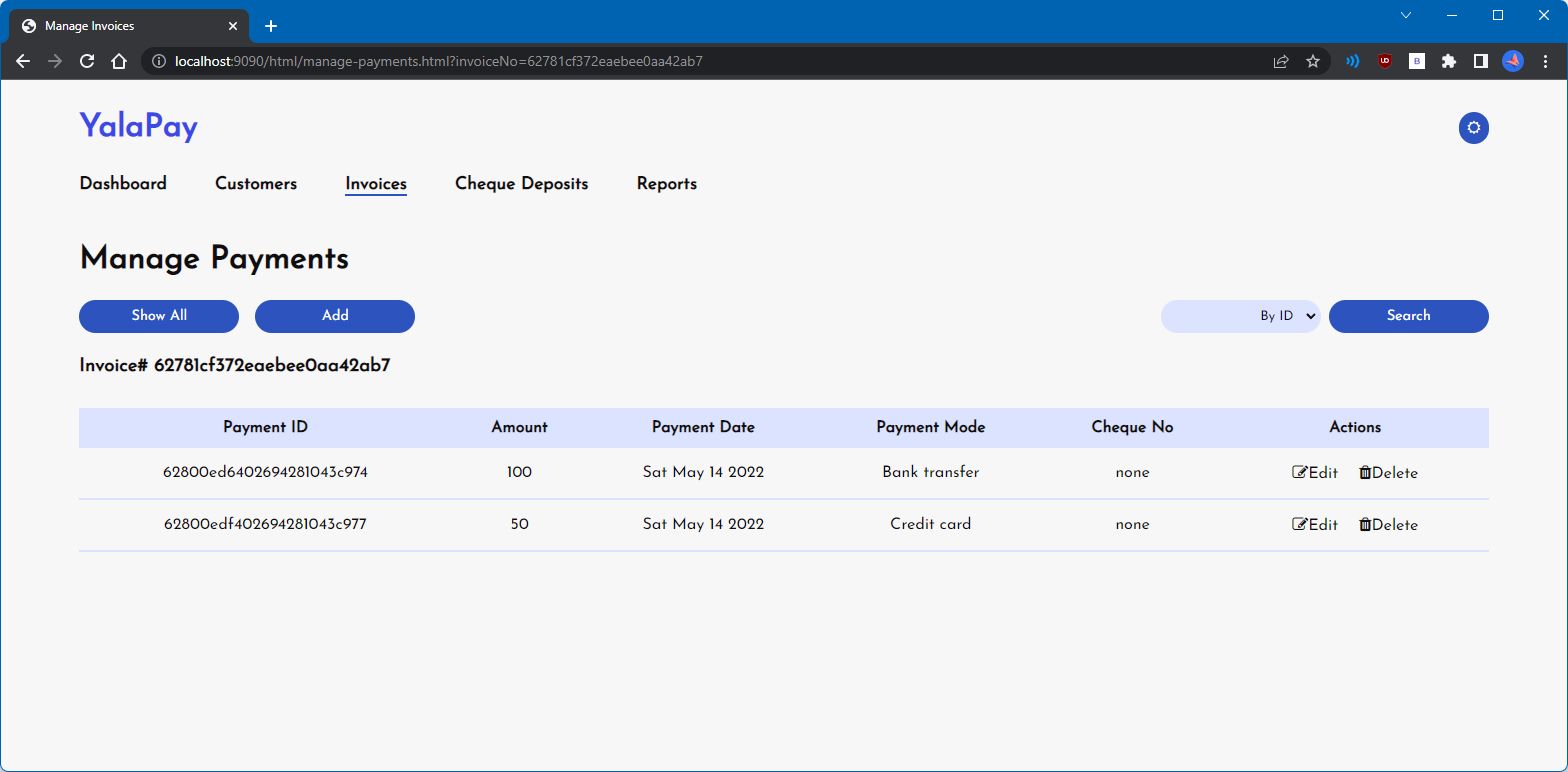
# Delete Invoice



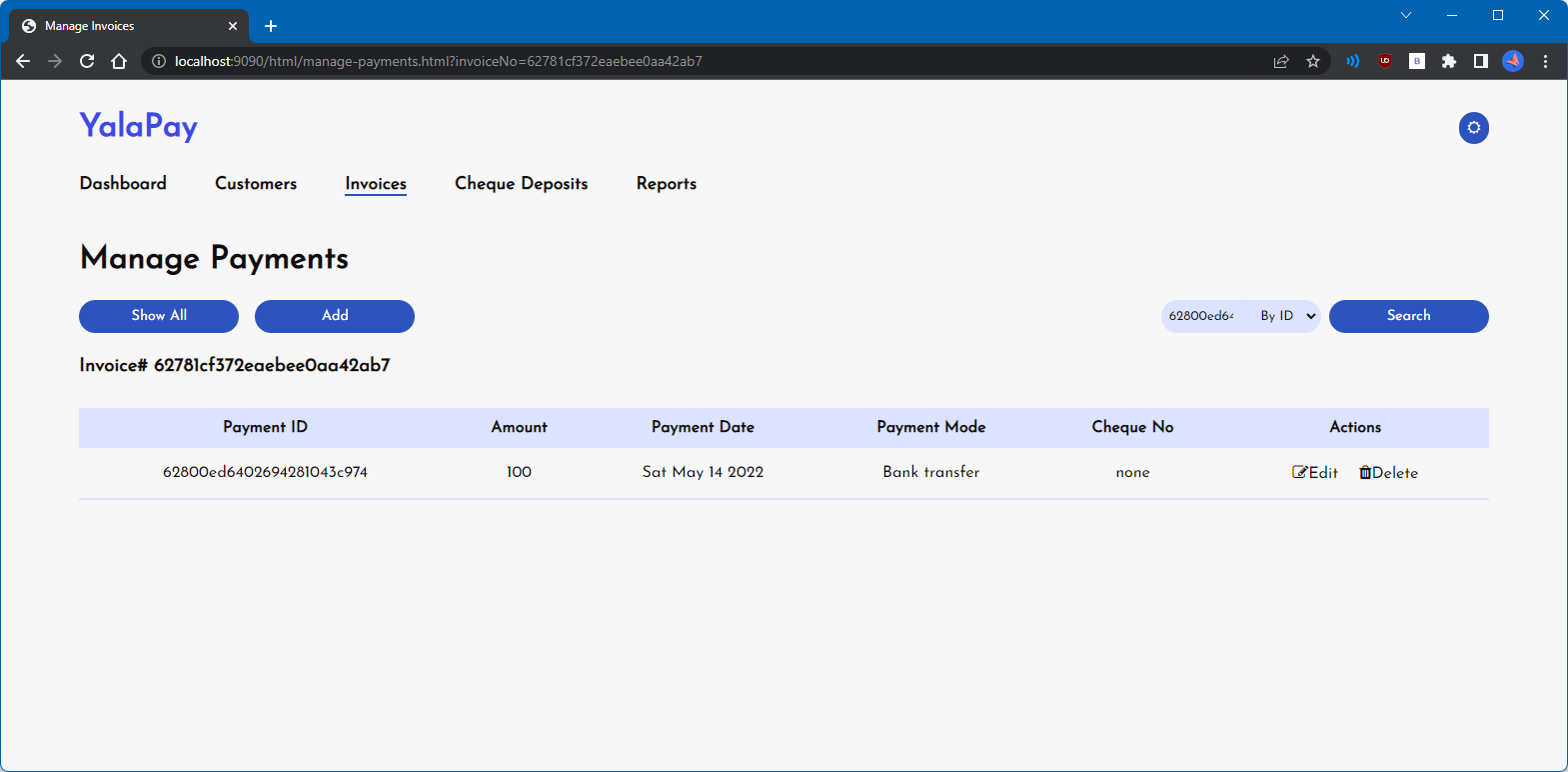


# U5 - List/Search/Add/ Update and Delete payments for an invoice

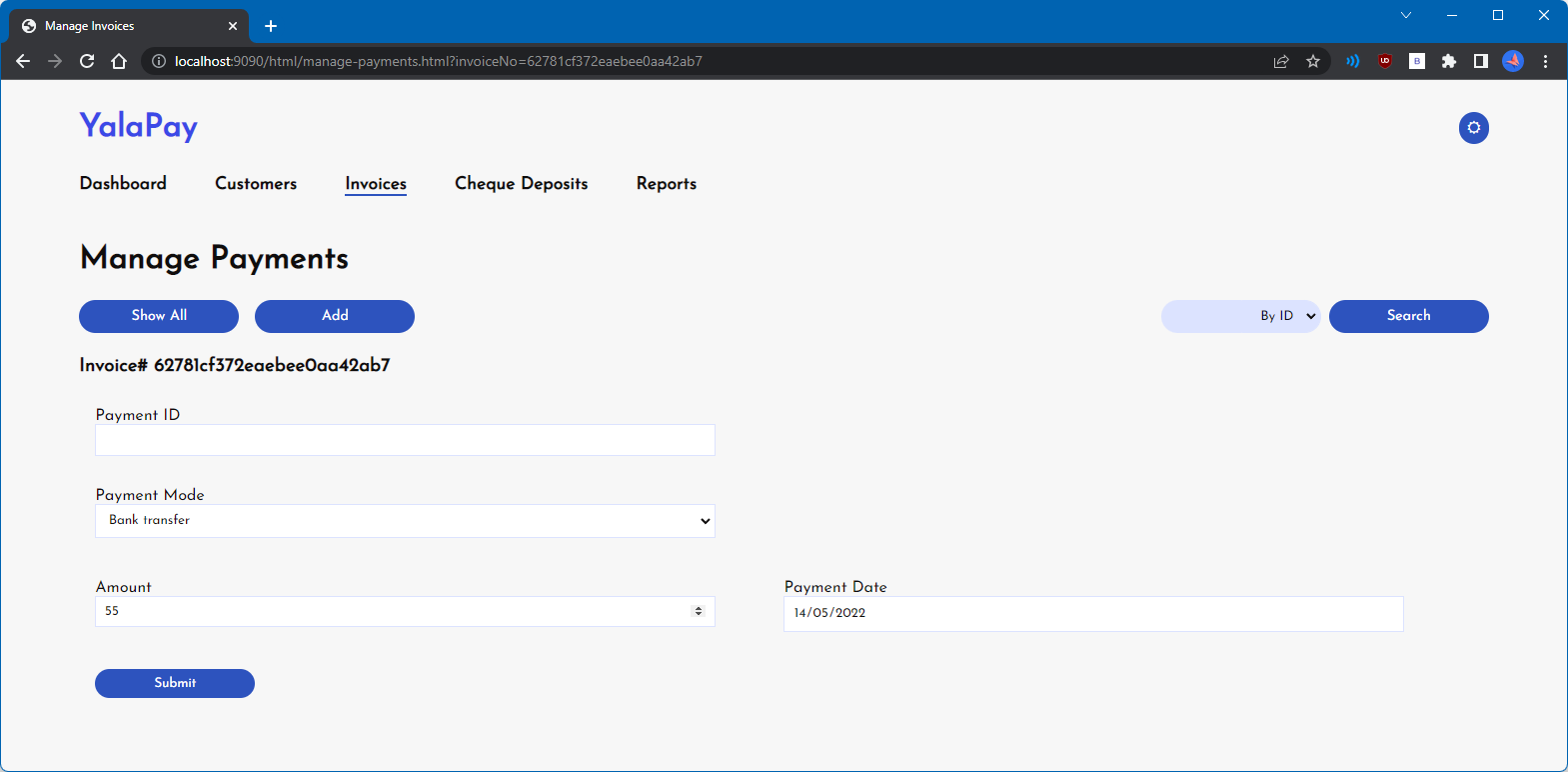
# List Payments

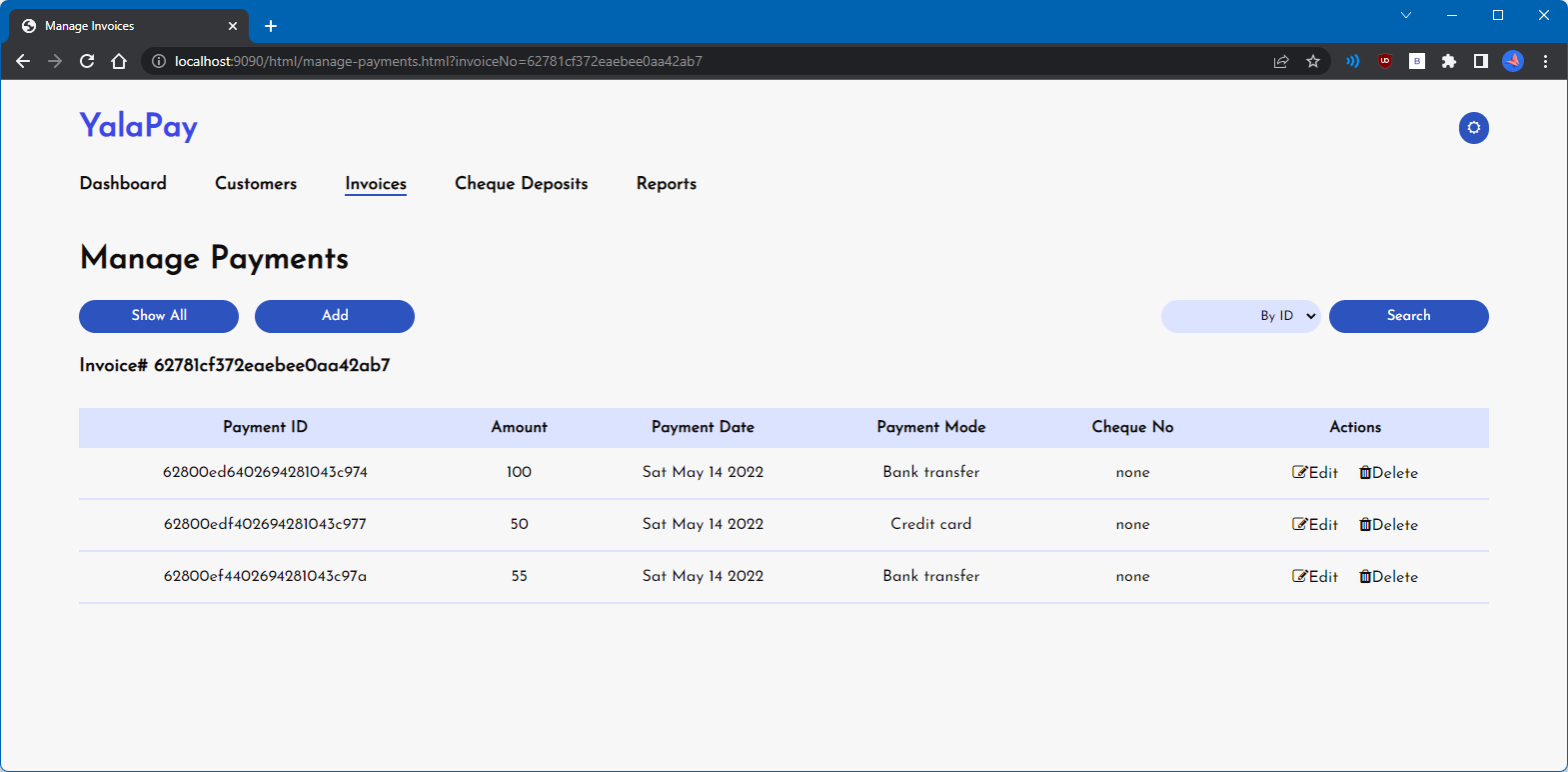


# Search Payments

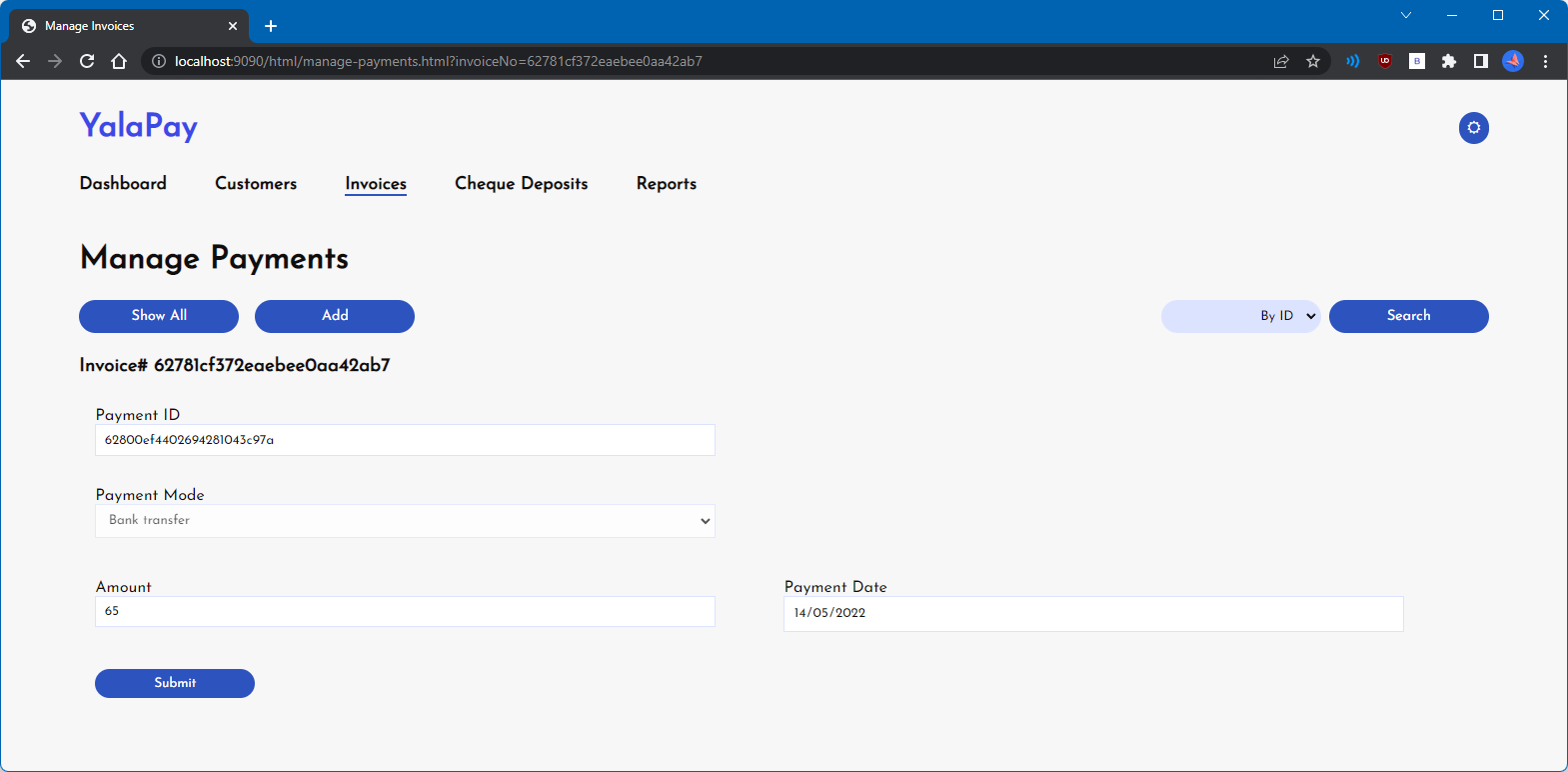


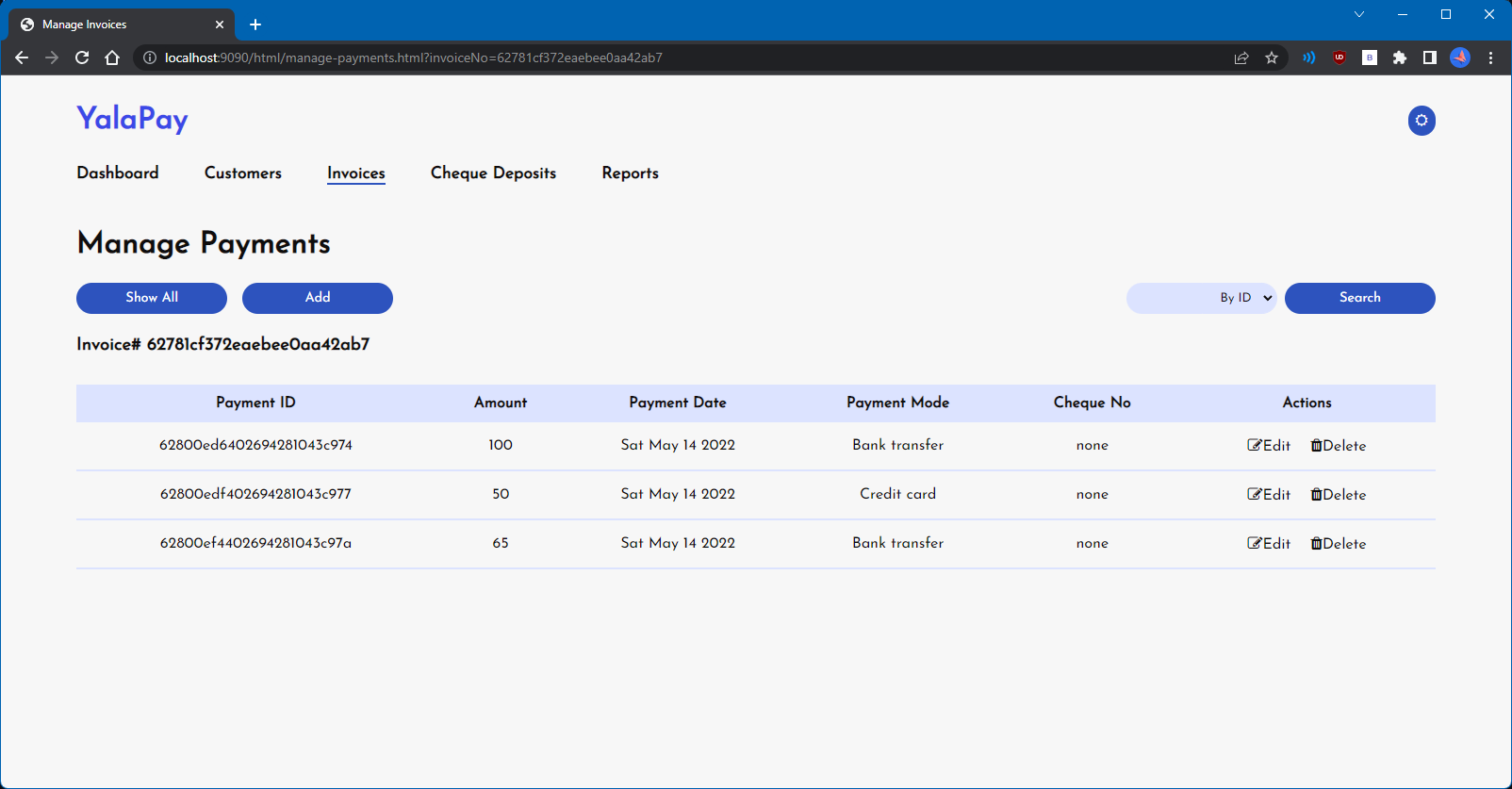
# Add Payment



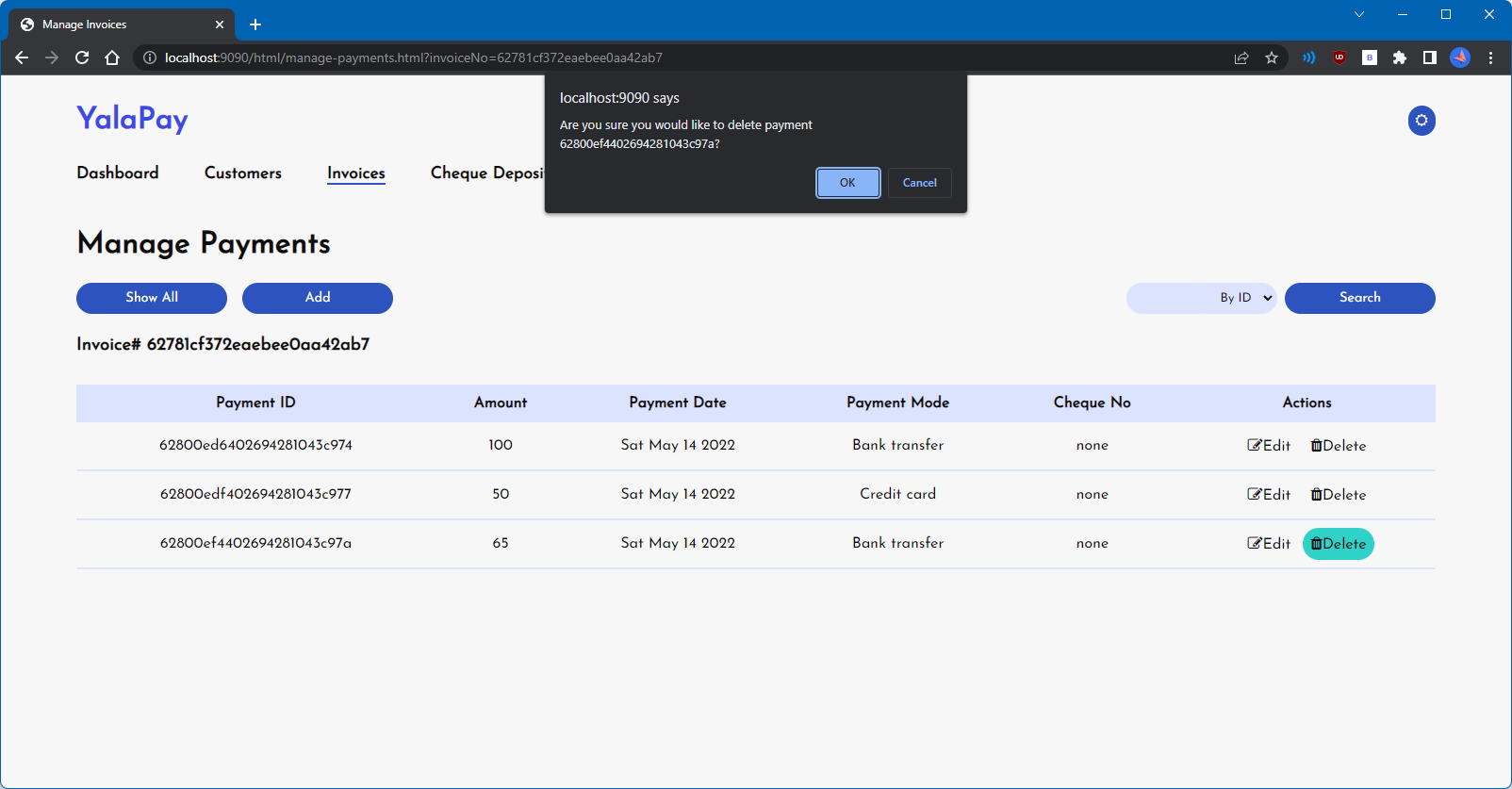


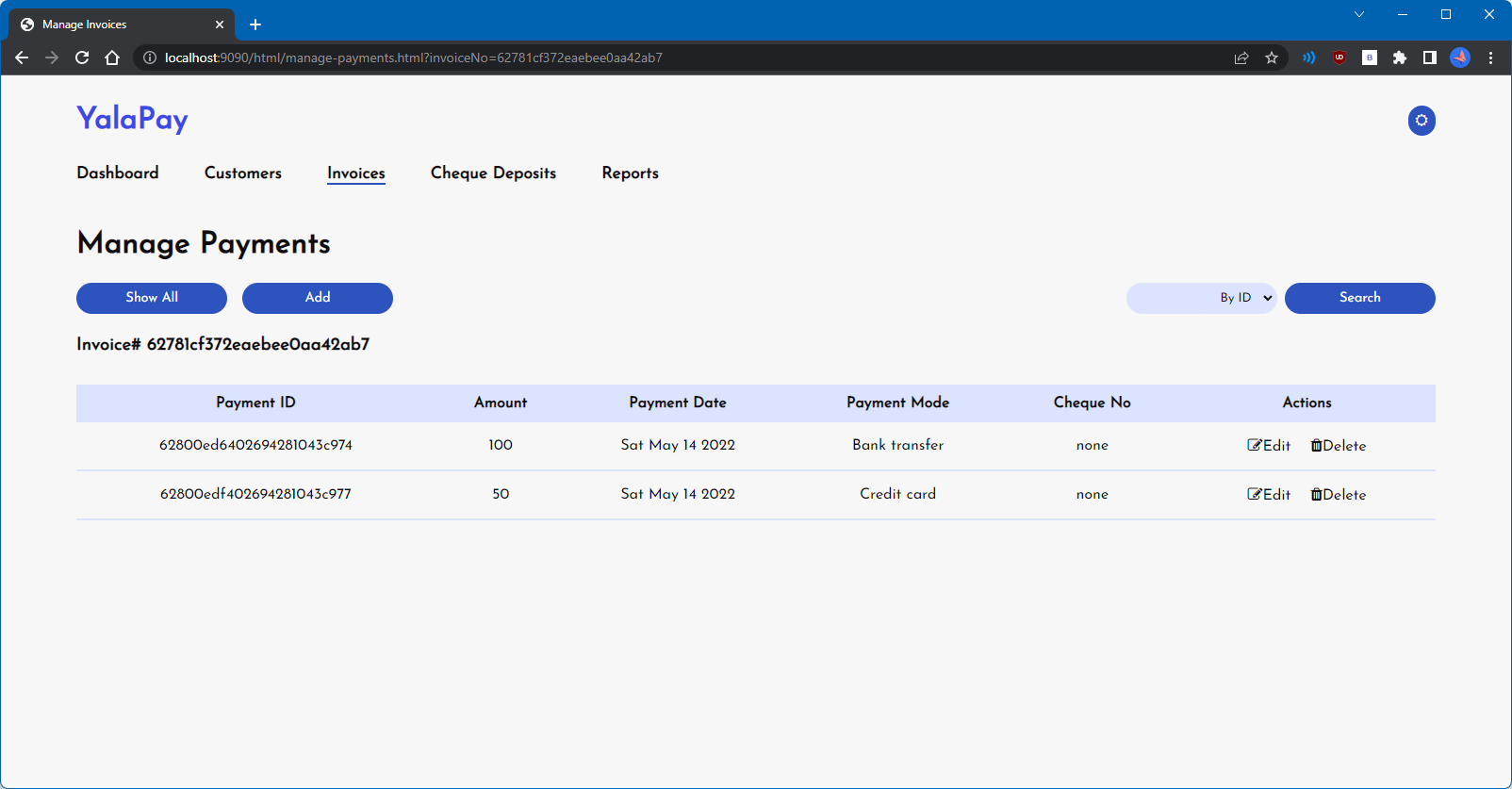
# Update Payment





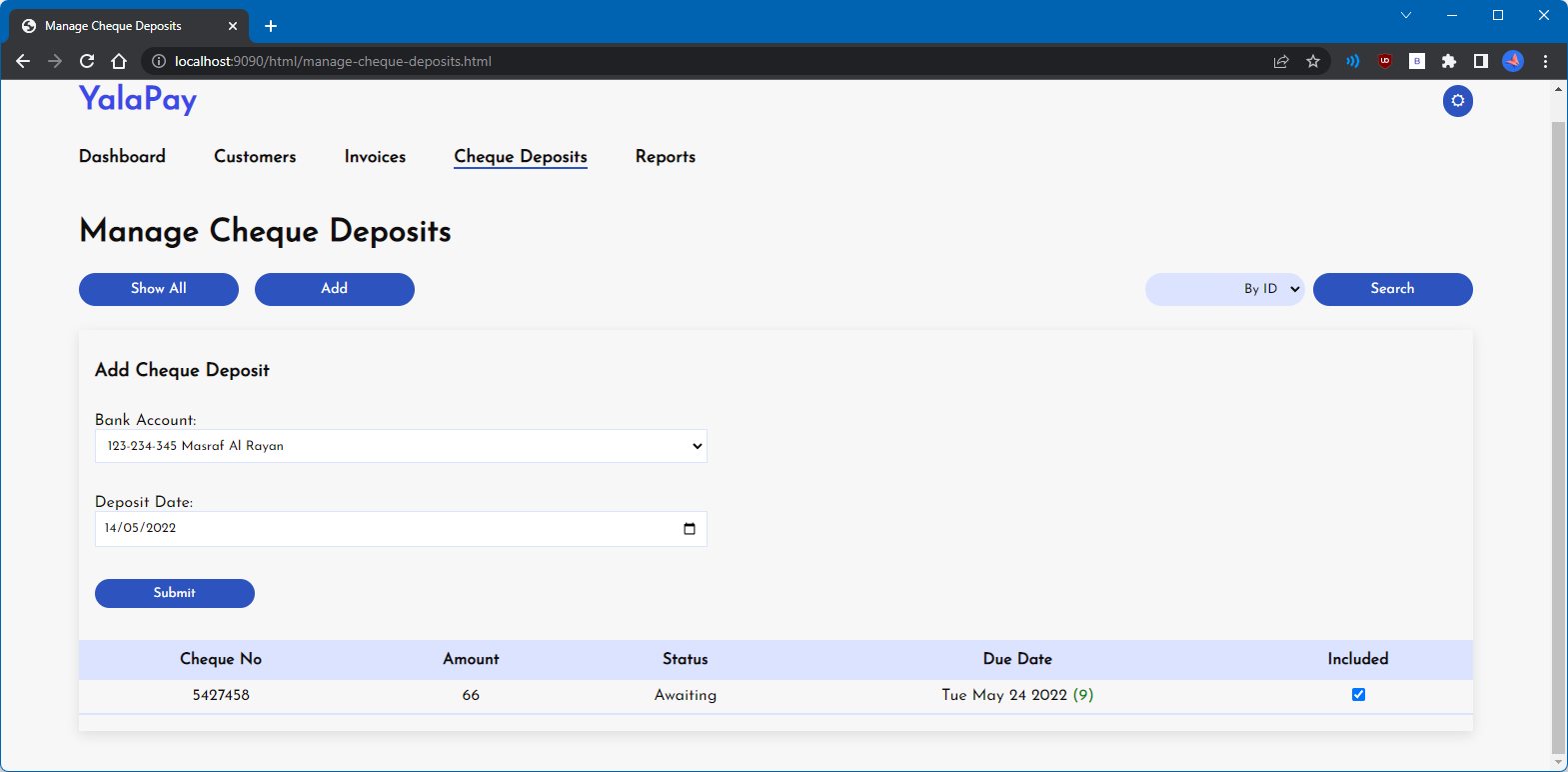
# Delete Payment

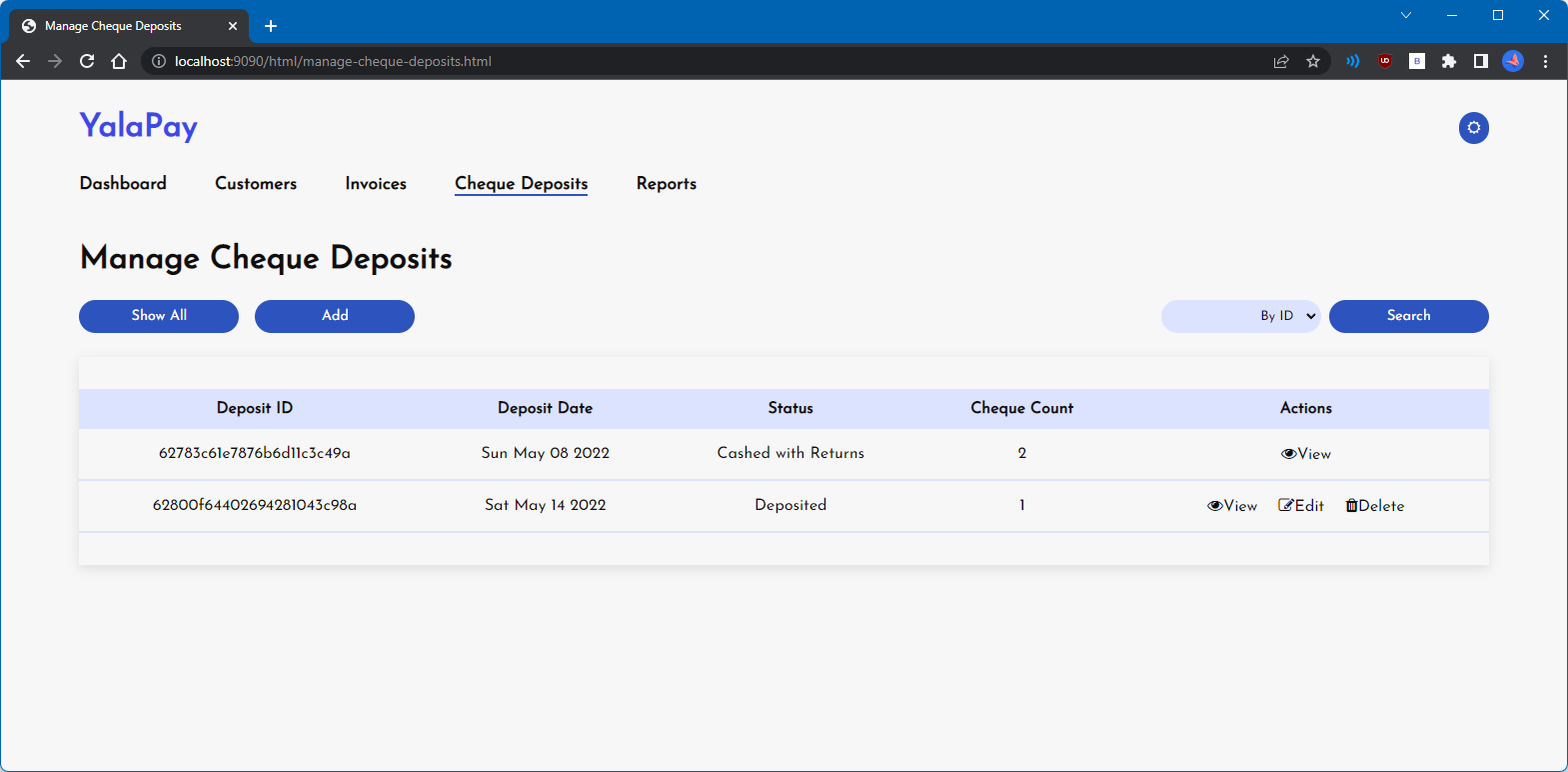




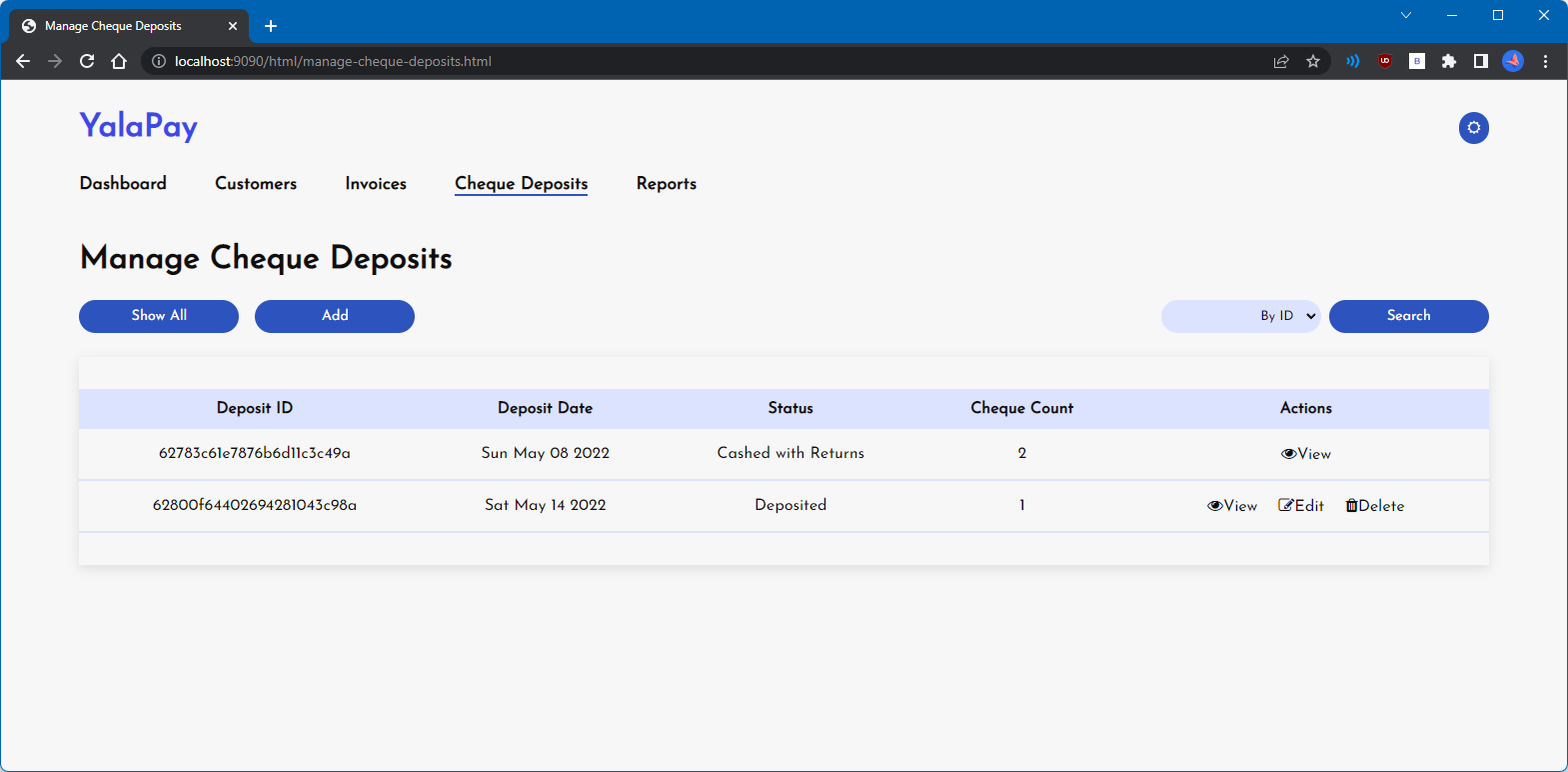
# U6 - Manage Cashing Cheques

# Add Cheques Deposit

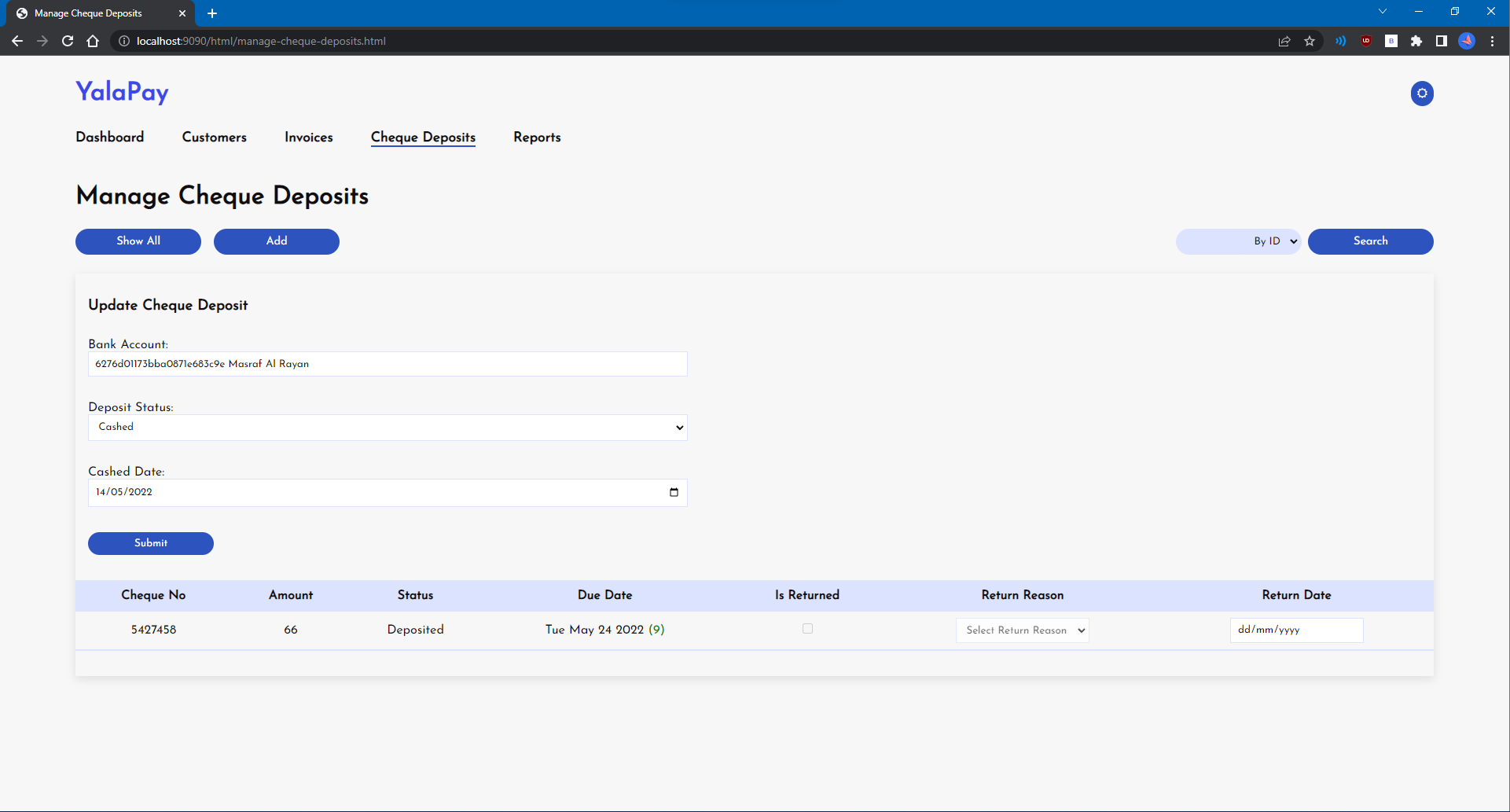


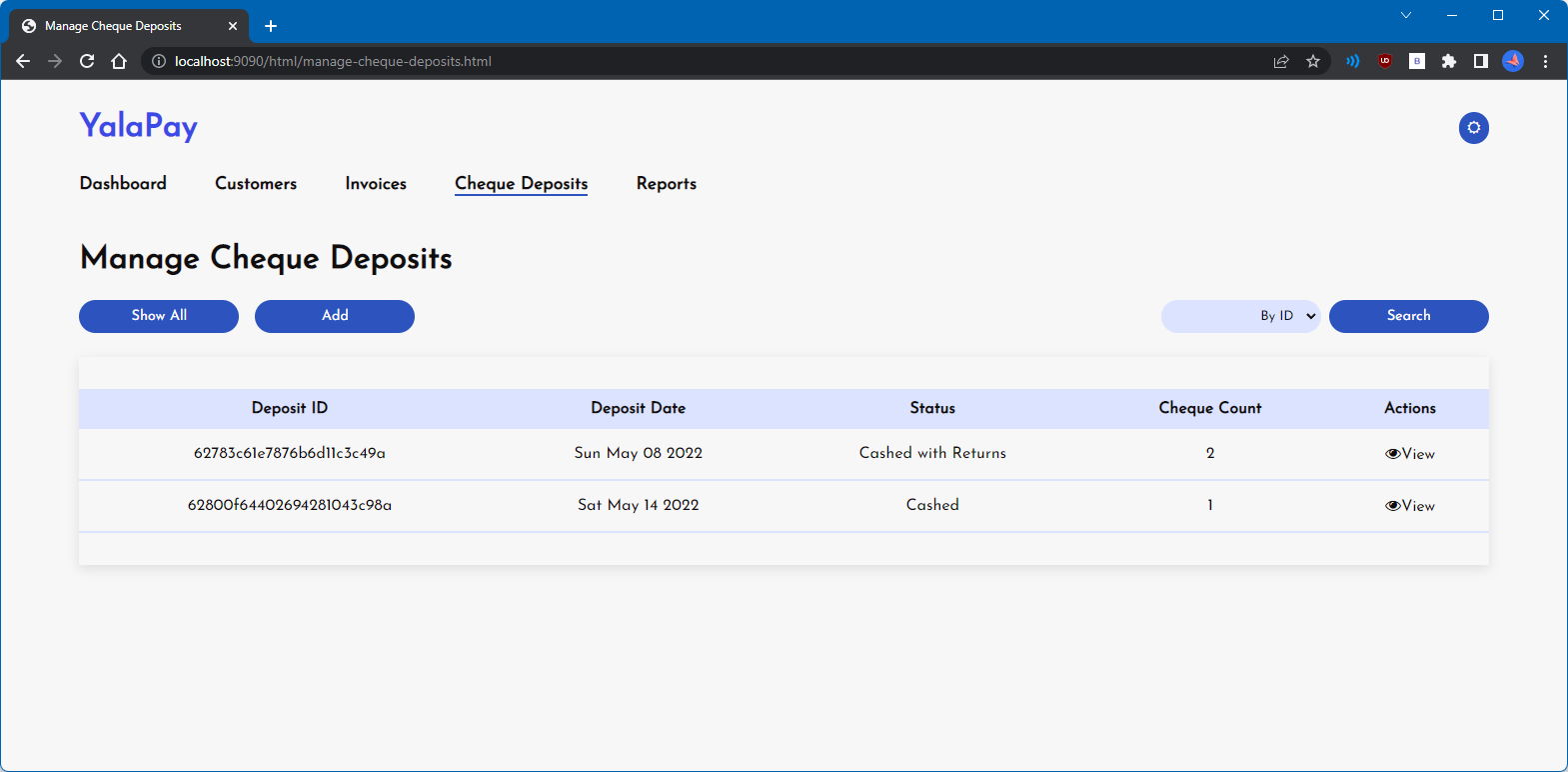


# List Cheque Deposits

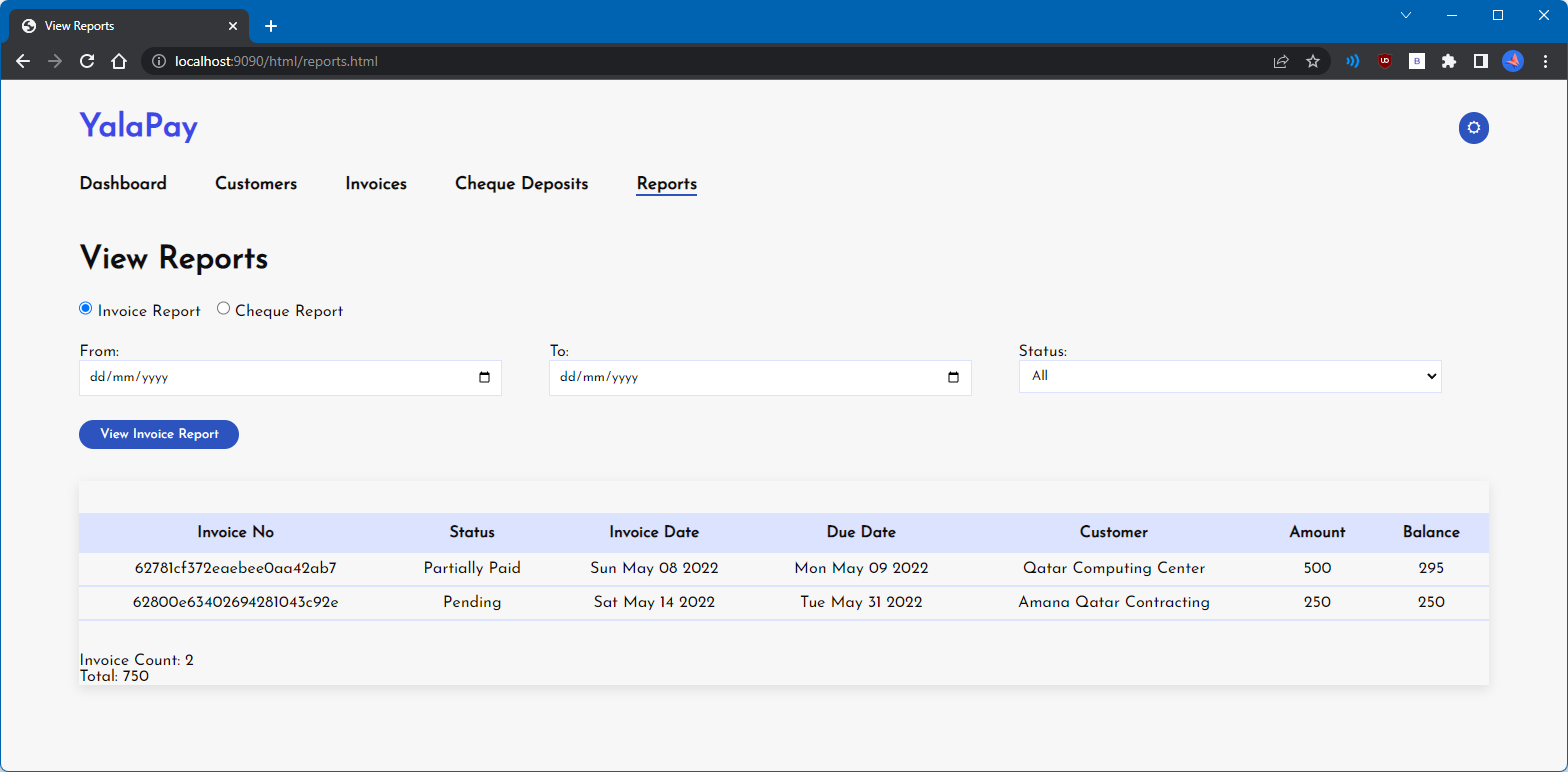


# Update Cheques Deposit

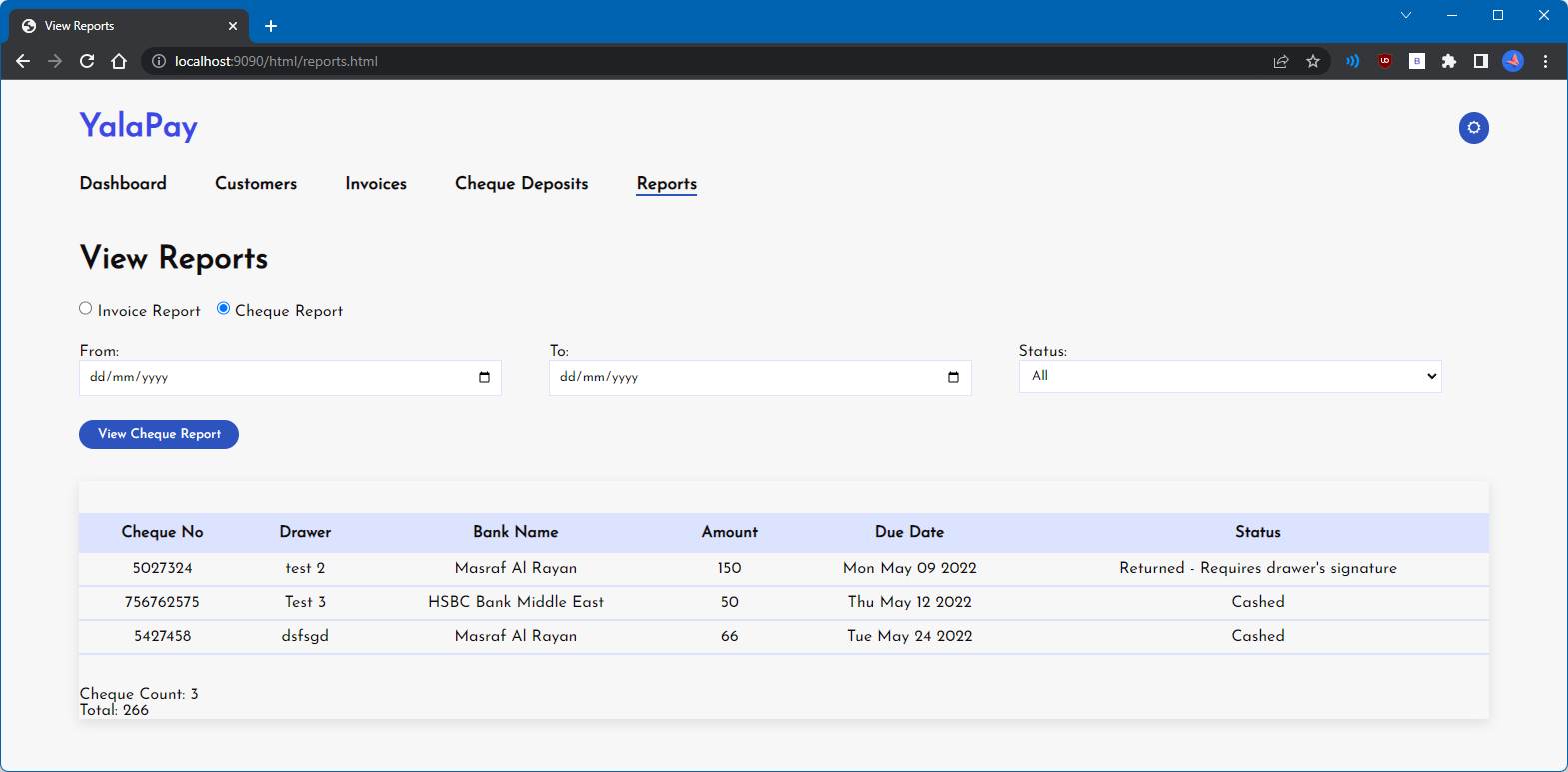




# U7 - Invoices Report



# U8 - Cheques Report



# Summary of team member contributions

* + Ahmed Abdelhamid

Update relevant areas of code to use MongoDB database for Login, Dashboard, Manage Payment

* + Hani Jafer

Update relevant areas of code to use MongoDB database for Manage Invoice, Cheques-Deposits (update), Reports

* + Syed Zubair

Update relevant areas of code to use MongoDB database for Manage Customers, Cheques-Deposits