Alonso Saenz  
Deeba Raine

Jacob Carmicle

Quinn Hall

Vedant Garg

Team #7

Yang CS 351

5/4/2025

## Table of Contents

* [Project Overview (p#)](#_9v3nnz8u93h6)
* [System Architecture (p#)](#_2nljy0e8ysnq)
* [Key Features (p#)](#_dx8pdikqndjp)
* [Future Improvements (p#)](#_txz87bbkvqd0)
* [Challenges & Solutions (p#)](#_jk9u78vnpcsa)
* [Conclusion (p#)](#_q4yj78j23eqv)

## Project Overview

The ManageSoft web application is intended to be used by representatives to view active customers and orders. Using a login system, an admin is able to access the database and execute SQL queries.

The purpose of the project is to create a safe, web-based application for administrators to access so that they can make changes to representative and customer data.

We had three main goals when approaching this project. The first was to create an effective web-based application that provides the user easy traversal of the database. The second was to connect the SQL database to the web-based application. Lastly, we implemented all required SQL queries to be available via the dashboard.

In terms of scope, the project achieves its fundamental requirements. It generates a report for each representative, including the amount of customers assigned to the representative, the average balance of the customers assigned to the representative, and the representative’s first and last name. It also has the capability to generate a report displaying the total quoted price of all the orders from a given customer. Additionally, the user is able to add new representatives, update customer credit limits, and exit the dashboard. However, the program does have some noteworthy limitations. For one, the software is not customer-oriented. It is intended only for administrators to use. Customers do not have any place where they can view their orders. Moreover, the application does not implement the ability to add a new customer.  
  
Jacob & Alonso mainly handled database setup, SQL queries, and CSS. Vedant & Deeba are responsible for the PHP, connecting to the database, and HTML. Quinn is responsible for the README.txt and Final Report.

Programming / Technologies Used

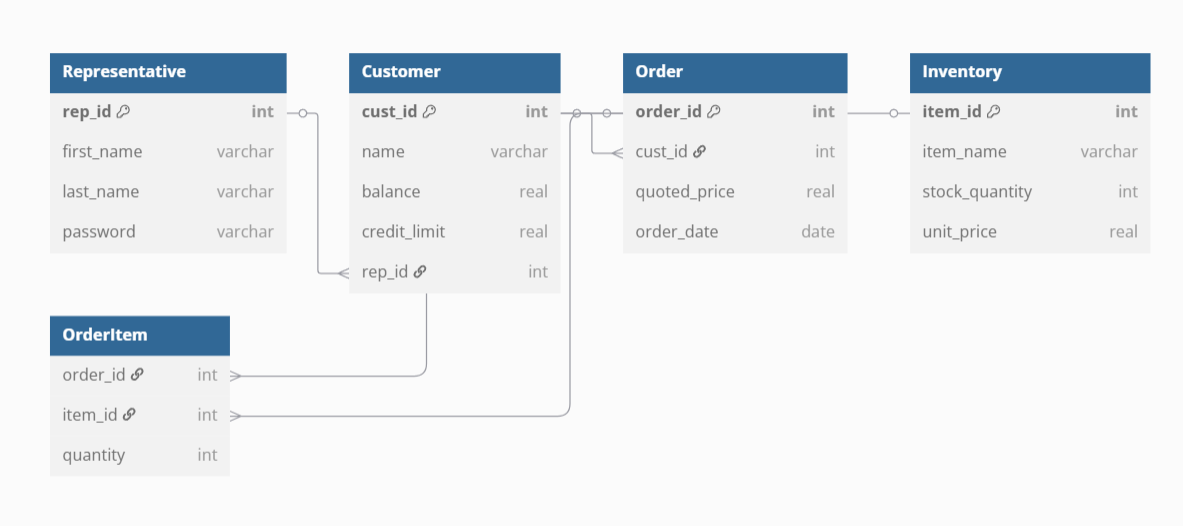
* PHP
* SQL
* CSS
* XAMPP
* MySQL

## System Architecture

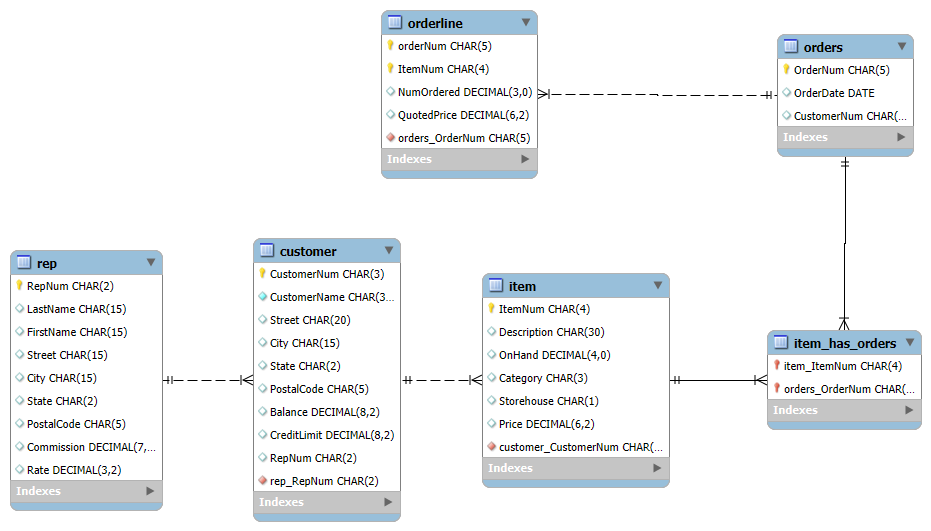
The frontend was handled using HTML5 and CSS3. The backend runs using PHP8 to issue SQL queries to a MySQL database, as shown in ER Diagram 2.

A typical request flow goes somewhat like this:

1. The user opens index.php, enters administrator credentials, and submits them.
2. PHP hashes the password, compares it with a stored digest, and, on success, starts a session with a timeout value.
3. Authenticated users are sent to dashboard.php where they have access to pre-implemented SELECT, UPDATE, and DELETE queries.
4. Each submitted action calls a small block of PHP that executes the corresponding SQL and returns corresponding error/success statements, keeping the entire process seamless.
5. A logout button ends the session and sends the user back to the login screen.



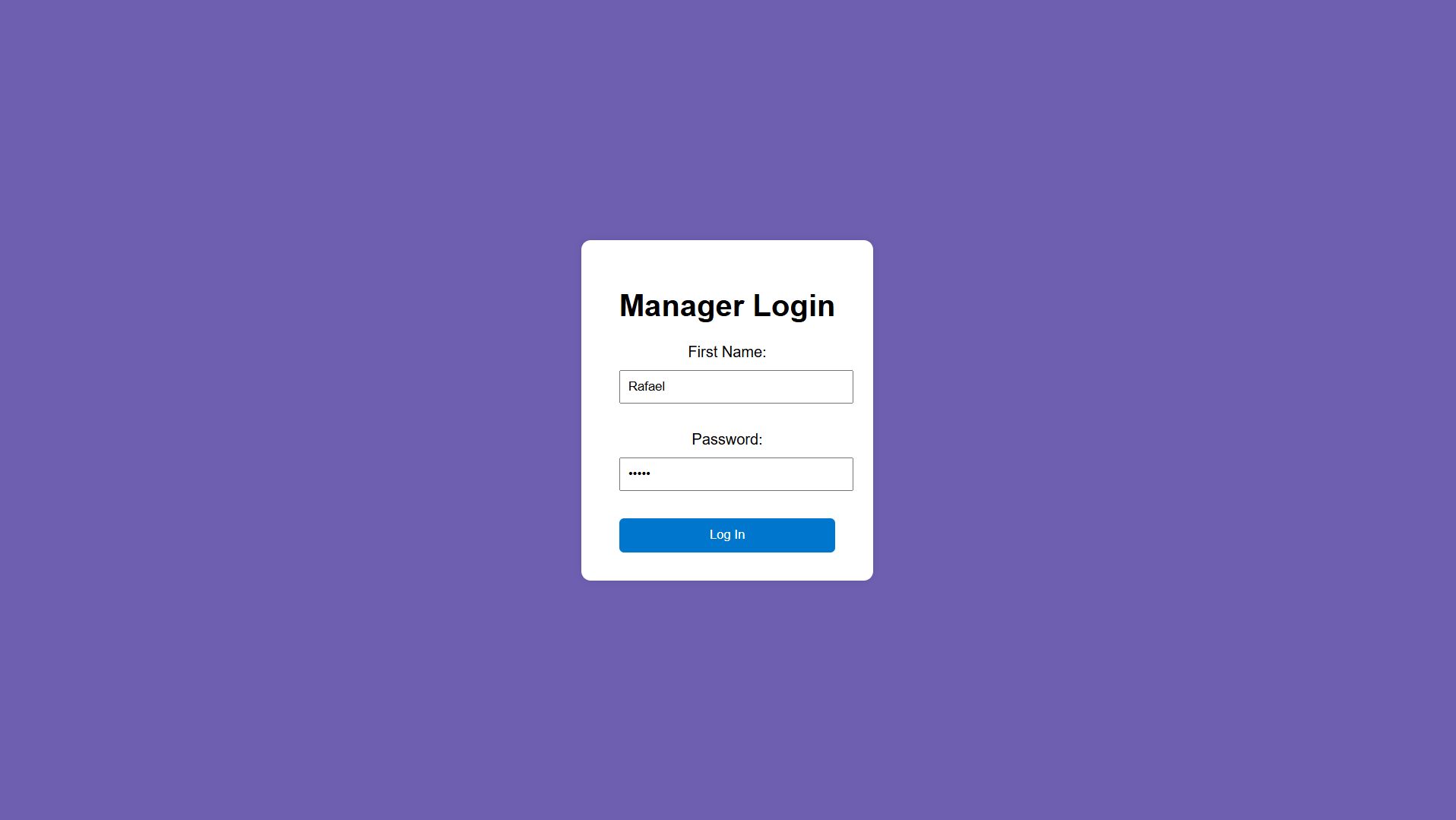
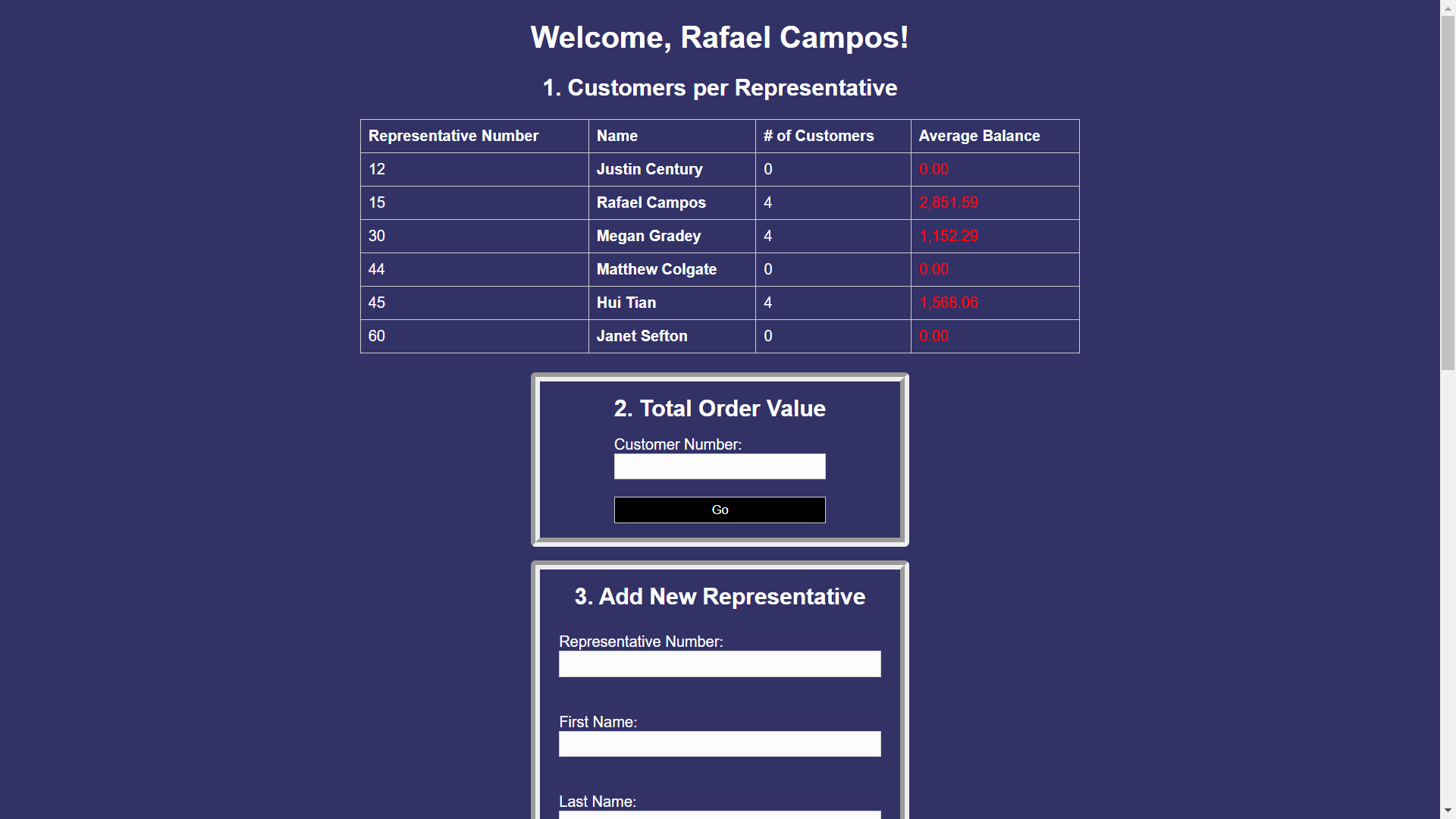
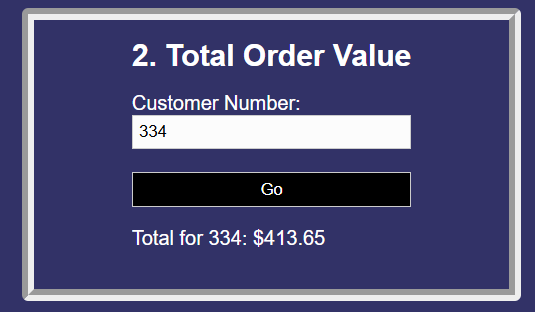
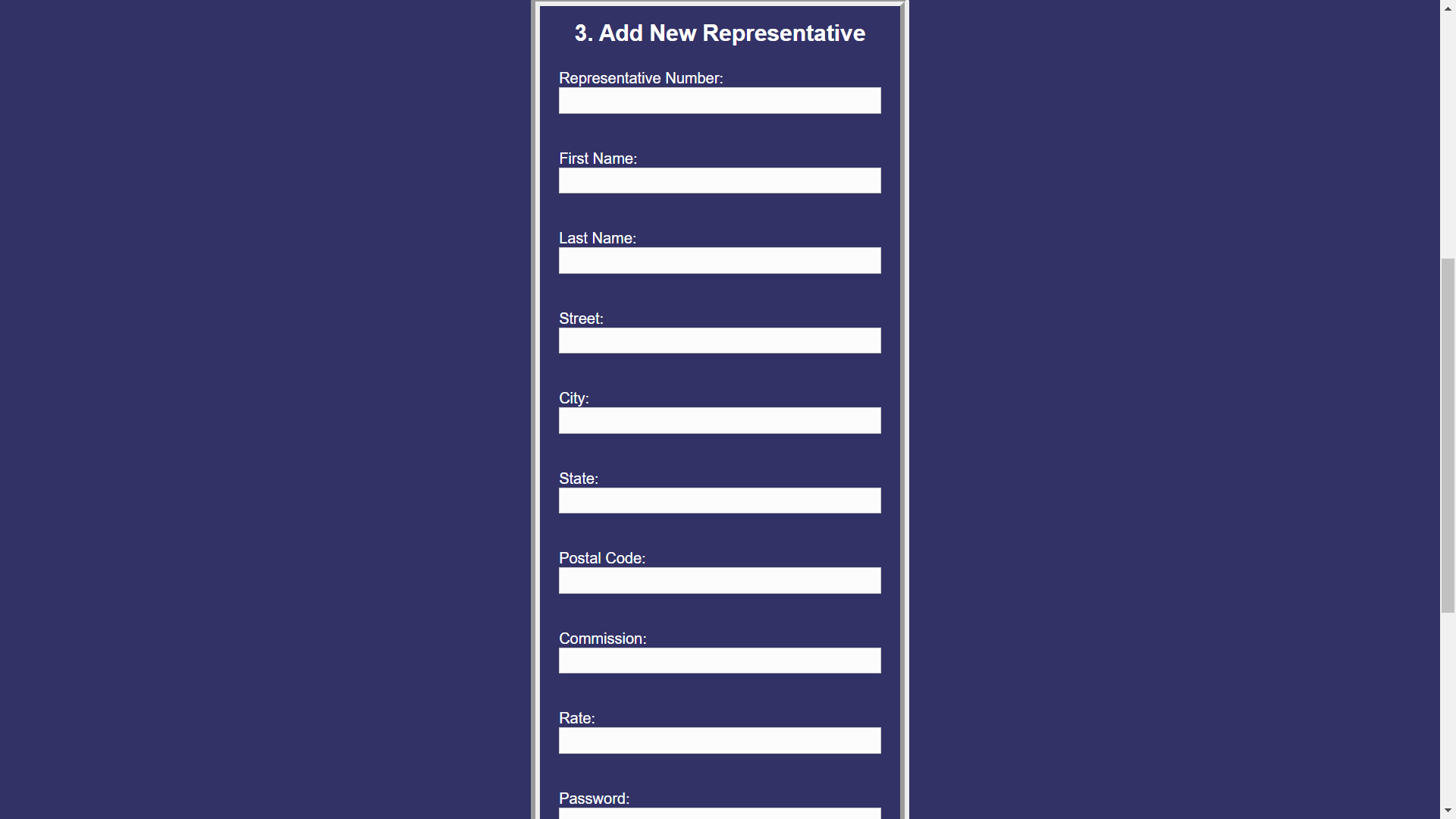
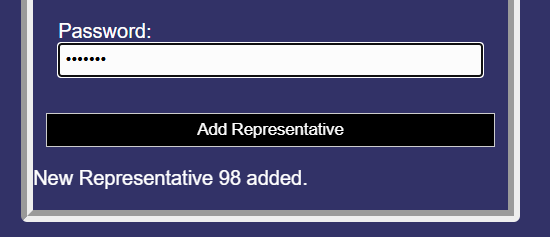
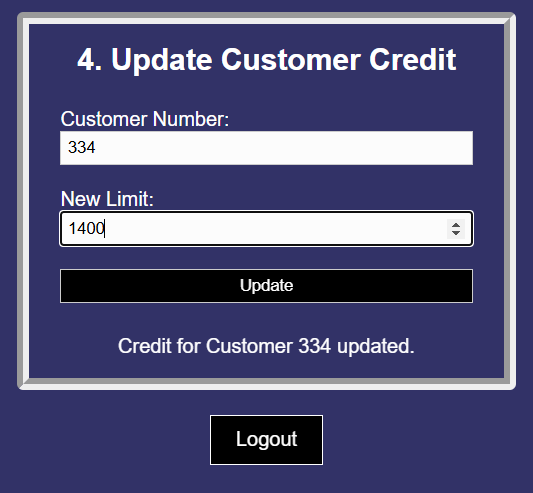
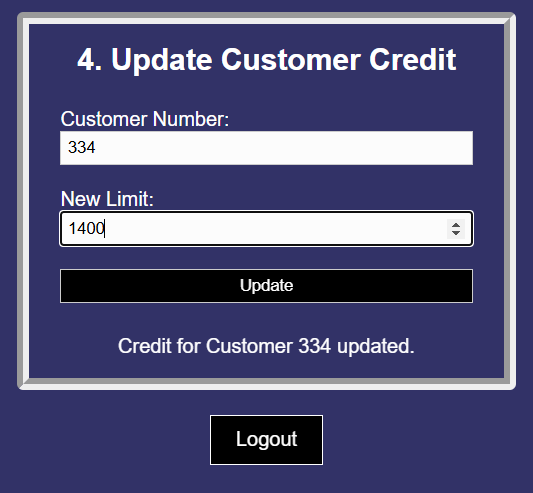
*ER Diagram 1 - Early generation used at the beginning of the project*



*ER Diagram 2 - Official diagram generated in MySQL*

## Key Features

As stated in the Project Overview, the core functionalities are relatively simple:

*   
  Can only log in using administrator credentials.
*   
  Can generate a report for each representative including:
  + The customers assigned to each representative
  + The average balance of all the customers assigned to representative
  + The representative’s first and last name
*   
  Can generate a report that displays the total quoted price of all the orders from a given customer.
*   
    
  Can add a new representative.
*   
  Can update a customer’s credit limit.
*   
  Can exit the dashboard.

These functions were implemented using SQL queries coded into the PHP. The full implementation of these functions are available in the dashboard.php file, but listed below are copies of them with notes to enhance readability.

| -- 1. Customers-per-Representative report  SELECT r.RepNum,  r.FirstName,  r.LastName,  COUNT(c.CustomerNum) AS num\_customers,  AVG(c.Balance) AS avg\_balance  FROM Rep r  LEFT JOIN Customer c ON c.RepNum = r.RepNum  GROUP BY r.RepNum;  ``` :contentReference[oaicite:0]{index=0}:contentReference[oaicite:1]{index=1}  ```sql  -- 2. Total quoted order value for a chosen customer  SELECT SUM(ol.QuotedPrice \* ol.NumOrdered) AS total\_value  FROM OrderLine ol  JOIN Orders o ON o.OrderNum = ol.OrderNum  WHERE o.CustomerNum = :c; -- :c is a bound parameter  ``` :contentReference[oaicite:2]{index=2}:contentReference[oaicite:3]{index=3}  ```sql  -- 3. Add a new sales representative  INSERT INTO Rep  (RepNum, FirstName, LastName, Street, City, State,  PostalCode, Commission, Rate, Password)  VALUES (?,?,?,?,?,?,?,?,?,?); -- values supplied from the POST form  ``` :contentReference[oaicite:4]{index=4}:contentReference[oaicite:5]{index=5}  ```sql  -- 4. Update a customer’s credit limit  UPDATE Customer  SET CreditLimit = :lim -- :lim is a bound parameter  WHERE CustomerNum = :c; -- :c is a bound parameter  ``` :contentReference[oaicite:6]{index=6}:contentReference[oaicite:7]{index=7} |
| --- |

## Future Improvements

Moving forward, the project would likely benefit from having a portal that customers can access. Because the project’s scope is limited to just administrator access, there was really no need to go any farther than that. However, in a real-world application, the company using this web-based application would likely need a way for customers to view and edit their own dashboard.

Additionally, the CSS could be improved to fit more screen dimensions than it can right now. There are a few optimizations that could be made with the design of the website that’d improve readability and accessibility.

## Challenges & Solutions

During this project our group faced a variety of challenges. As a whole, making sure all of us could access, upload, and change our code was integral to our success. To confront this, Vedant created a group project on Github that we could all access. This was a huge help throughout the entire process.

Early on in the project’s life, Alonso accidentally wrote part of the SQL database because of a miscommunication on his part. He was unaware that certain materials were provided to complete the project, and as a result, he did more work than was needed. It was ultimately no problem, though, because no momentum was lost while he was working.  
  
Deeba found that getting everything in the web-based application to connect to the database was particularly challenging. It wasn’t anything she couldn’t overcome, however. It simply took her some time to trudge through and troubleshoot any small issues that arose.

While working on the web-based application, Vedant found working with PHP to be a little troubling. Since it had been a while since he’d utilized the programming language, he spent a lot of time debugging code. A tedious challenge for sure, but it was easily solved through commitment and dedication.

Jacob said the biggest challenge he faced was setting up the CSS in the web-based application. It was hard for him to pinpoint how to change certain aspects of the website. As is expected, he overcame it easily using online resources to supplement his knowledge, and he definitely has something to show for it. The web-based application looks phenomenal.

The biggest challenge Quinn faced was knowing what information to include in the project report. In the project syllabus, it is not clearly stated what needs to be included in the final document, but Quinn was able to work past this by utilizing online resources and consulting his professor. Through these efforts, Quinn was able to compile a comprehensive report of his group’s project.

## Conclusion

In the end, the ManageSoft project has delivered on its primary objectives; it is a secure, web-based dashboard that allows administrators to view and edit representative, customer, and order information without having to write a single line of SQL. The entire process is reliable, repeatable, and functions entirely within a user-friendly interface.

Throughout the process, the team as a whole has deepened their experience with collaborative programming, database to web page integration, and PHP. Many issues that were dealt with throughout the development required a dedicated team member to spend long stretches of time debugging blocks of code. More than anything, this project is a demonstration of patience and persistence. It is a hard lesson to learn, but clearly, it was worth the results.