

TechPro Academy Project

Application Documentation: Order and Customer Management Web Application

Purpose

The purpose of this application is to serve as the final project for the bootcamp attended by Manos Therapidis. It aims to enhance the knowledge and practical understanding of the concepts covered during the bootcamp. The application provides a simplified system for managing orders and customers.

Goals

1. **Knowledge Enhancement:** Gain hands-on experience by developing a full-stack web application using React, Java, and MySQL.
2. **Application Understanding:** Gain a deeper understanding of various concepts, including frontend development, backend development, API integration, database management, and system architecture.
3. **Practical Application:** Apply the knowledge gained during the bootcamp to build a functional web application that showcases the ability to implement core features and follow best practices.
4. **Demonstrate Proficiency:** Demonstrate proficiency in utilizing frontend technologies, backend technologies, and database management to create a cohesive and functional application.
5. **Project Completion:** Successfully complete the final project within the given timeframe, meeting all the specified requirements and objectives.

Application Overview

The order and customer management web application is designed to simplify the process of managing orders and customers for a business. It provides the following core functionalities:

6. **Customer Management:** Allows users to view, add, edit, and delete customer information, including names and contact details.
7. **Order Management:** Enables users to create new orders, associate them with specific customers, and delete them.
8. **Order Details:** Provides the ability to view and manage order details, including quantities, and Item name.
9. **User Interface:** Offers an intuitive and user-friendly interface for seamless navigation and interaction with the application.
10. **Data Persistence:** Utilizes a MySQL database to store and retrieve customer and order information.

Technology Overview

Frontend Technology: React

- **Component-Based Architecture:** The application follows a component-based architecture, where UI elements are encapsulated into reusable and self-contained components.
- **Virtual DOM:** React utilizes a virtual DOM, which optimizes rendering performance by updating only the necessary components instead of re-rendering the entire page.
- **State Management:** React's state management allows for the dynamic updating and rendering of UI components based on changes in data or user interactions.
- **Routing:** React Router is used for handling client-side routing and navigation within the application.

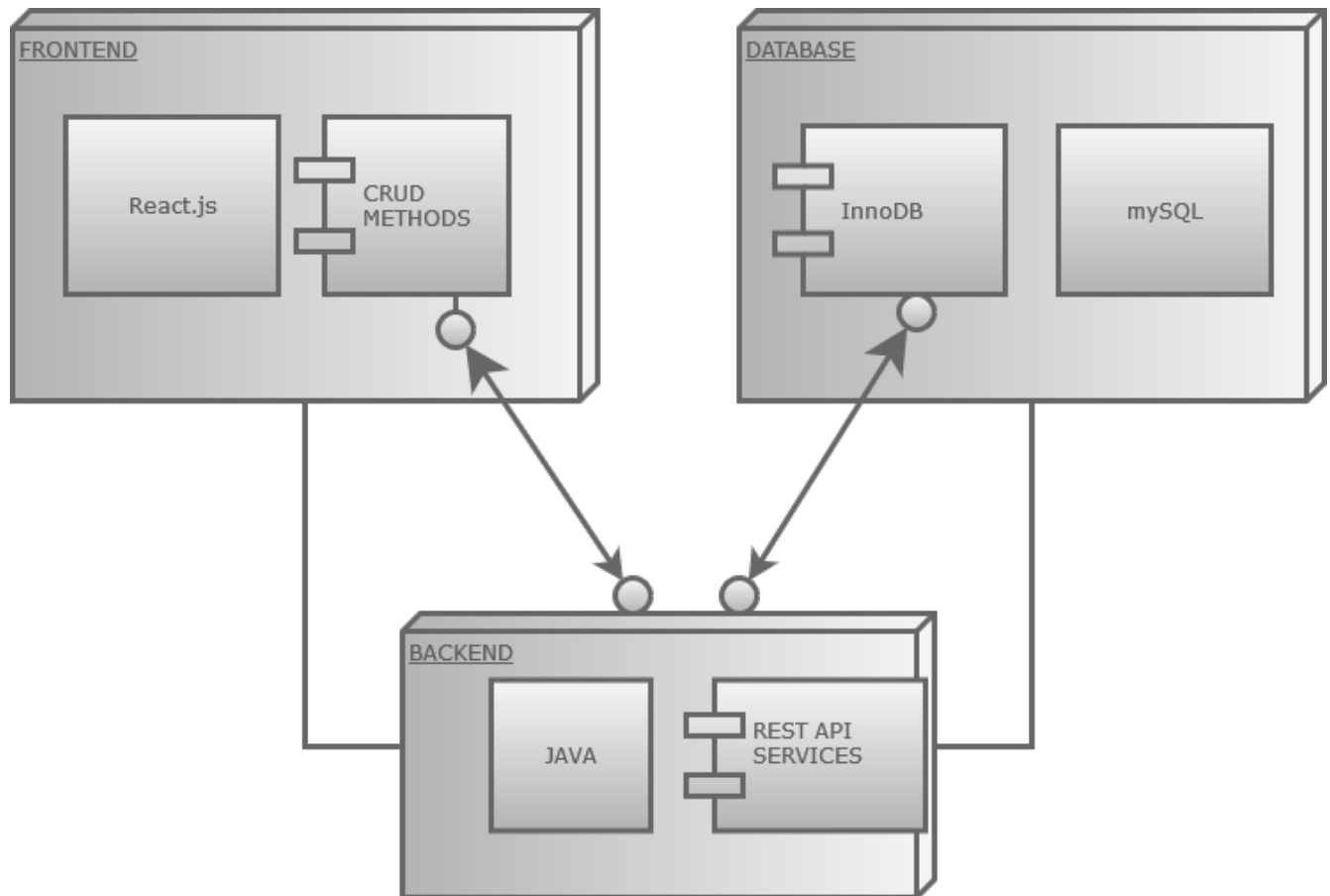
Backend Technology: Java

- **RESTful API:** The backend exposes a RESTful API to facilitate communication between the frontend and the database. It follows standard HTTP methods (GET, POST, PUT, DELETE) to handle CRUD (Create, Read, Update, Delete) operations.
- **Java Servlets:** The application uses Java Servlets, which provide a server-side component model for handling HTTP requests and generating responses.
- **Business Logic:** The backend incorporates business logic to process and validate user requests, interact with the database, and handle various application-specific functionalities.

Database Technology: MySQL

- **Entity-Relationship Model:** The database schema is designed based on an entity-relationship model, defining the relationships between entities such as customers and orders.

Application Diagram



Deployment

Before running the Docker command, you need to install the dependencies for Node.js in the frontend folder.

1. Frontend Setup:

- Navigate to the `"/Tech_Pro_Final_Project/frontend"` directory.
- Run the following command to install the dependencies:
`npm install`

2. Backend Setup:

- The backend already contains the `"project-0.0.1-SNAPSHOT.jar"` file, so there is no need to build the app.

3. Building and Initiating Docker Containers:

You have two options to build and initiate the Docker containers:

Option 1: Using Docker Compose

- In the `"/Tech_Pro_Final_Project"` directory, run the following command:
`docker-compose up -d --build`

Option 2: Using the `boot_system.sh` Script

- In the `"/Tech_Pro_Final_Project"` directory, run the following command:
`./boot_system.sh`
- If you encounter an error running the `"boot_system.sh"` script, run the following command first:
`chmod +x boot_system.sh`

Important Note:

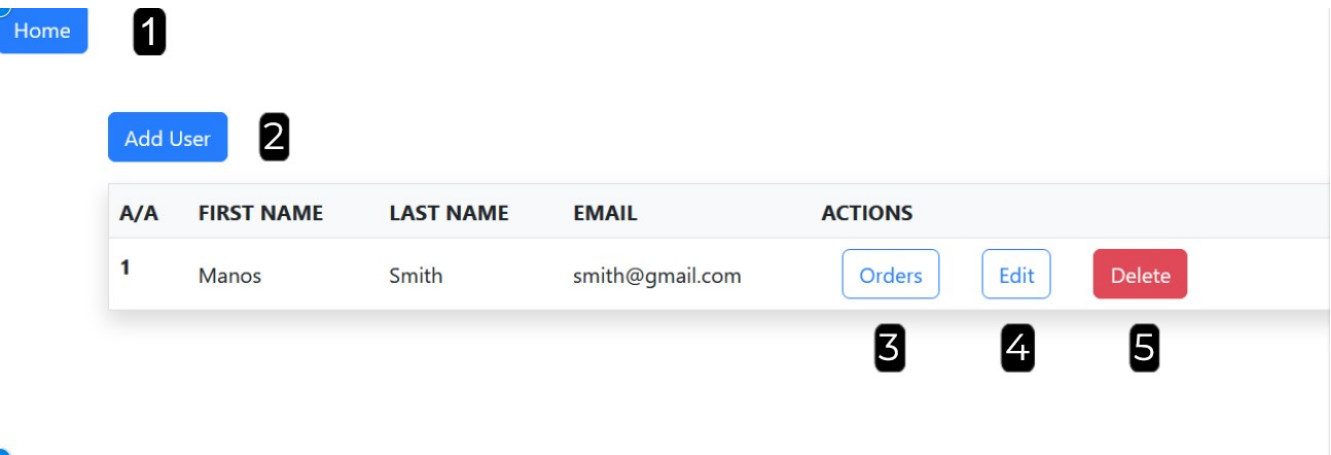
If you are running the application in a virtual machine, make sure to port forward the following ports:

- 1) Port 3000 for the frontend.
- 2) Port 8080 for the backend.

Please ensure that you follow these steps carefully for a successful deployment. Feel free to reach out if you have any further questions or concerns.

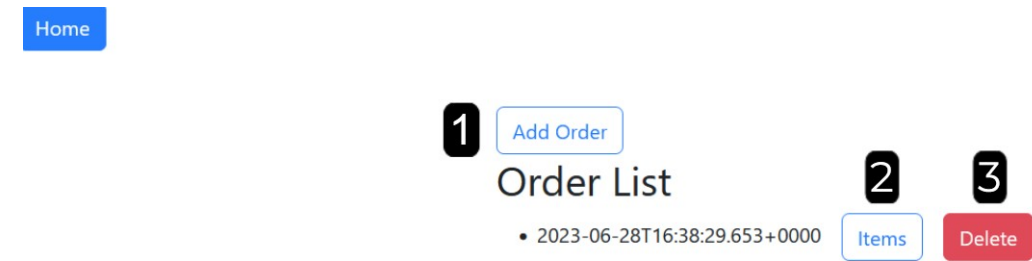
End User Manual

Home page



- 1 . Navigate to the Home page.
- 2 . Navigate to Add User form page
- 3 . Navigate to Order list page.
- 4 . Edit User Information.
- 5 . Delete User.

Order List Page



- 1 . Add new Order.
- 2 . Navigate to Item list page.
- 3 . Delete Order.

Item List page

me

Add Item

1

Edit Order

Item Name

Banana

Quantity

2

Update Order

2

A/A	DATE	ITEM	QUANTITY	ACTIONS		
1	2023-06-28T16:38:29.653+0000	Apple	5	<div>3</div>	<div>Edit</div>	<div>Delete</div> <div>4</div>
2	2023-06-28T16:38:29.653+0000	Banana	2		<div>Edit</div>	<div>Delete</div>

- 1 . Open the popup Add Item form.
- 2 . Add Item or Update the Item.
- 3 . Open the popup Edit Item form.
- 4 . Delete Item.