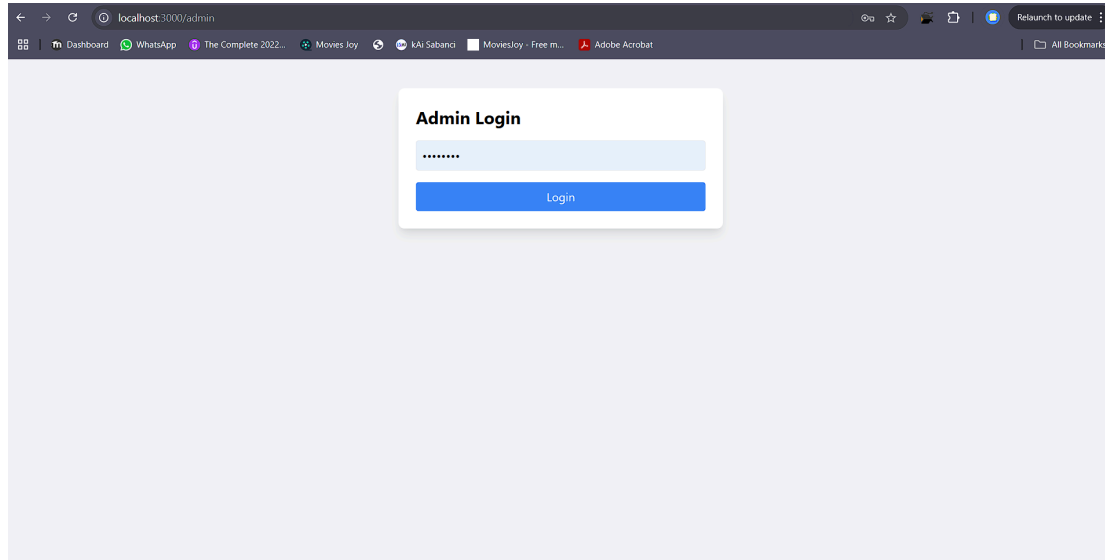


Coffee Ordering System Phase 3

(Emre Kaan Usta - Bahar Abit - Hüseyin Eren Yıldız - Ayşe Efdal Erdem)

(30728 - 28933 - 31047 - 31082)



1. Admin Login Page

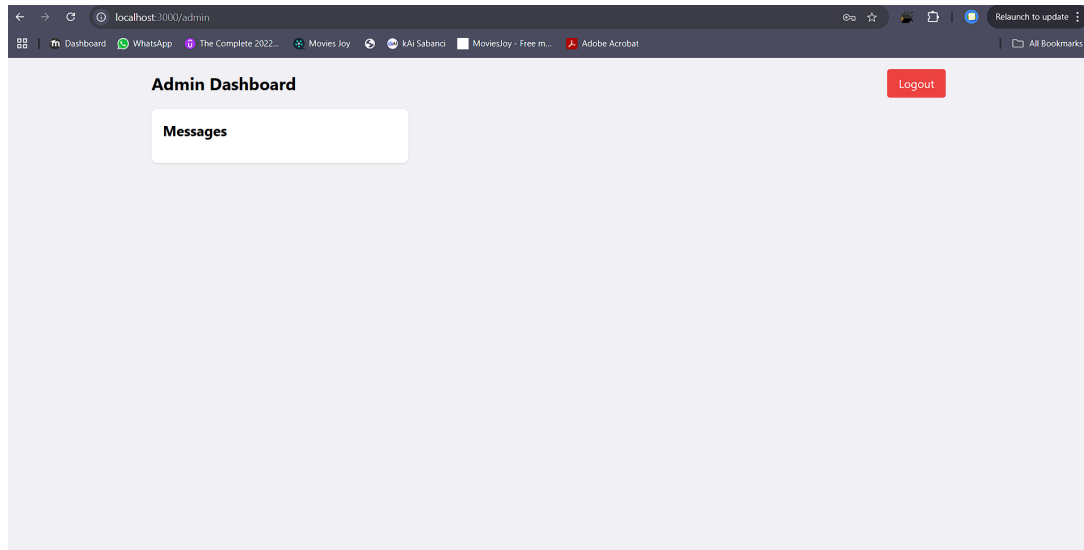
The first screenshot illustrates the admin login page. This page acts as a secure gateway for administrators to access the support system.

Features:

- **Login Field:** A password input field for the admin to authenticate their access.
- **Login Button:** A button to submit the entered credentials and proceed to the admin dashboard.
- **Design:** A clean and intuitive interface with central alignment to ensure ease of use.

Purpose:

- To ensure that only authorized personnel can access user support messages.
- The admin login page prevents unauthorized access to sensitive data by implementing Firebase Authentication.



2. Admin Dashboard

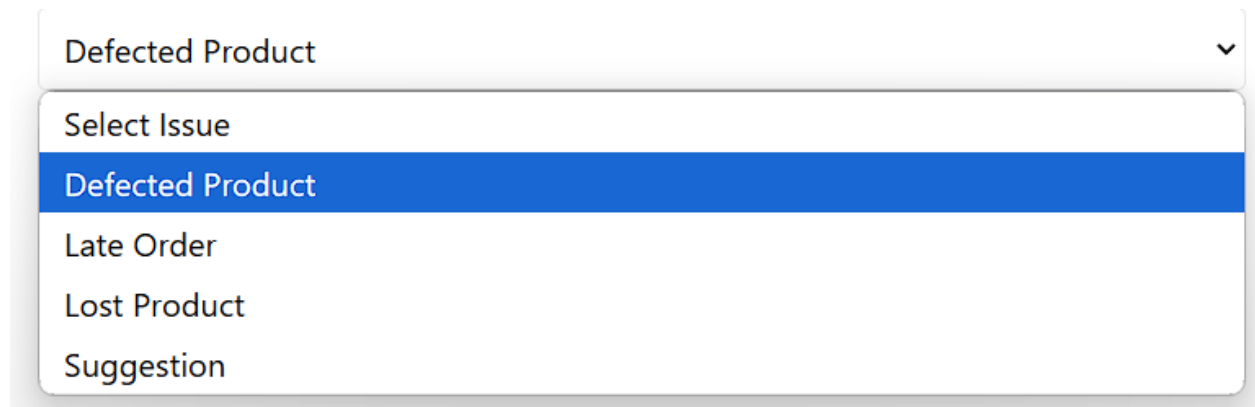
The second screenshot shows the admin dashboard, which is accessible after successful login. This dashboard is the control center for managing user support messages.

Features:

- **Messages Section:** Displays real-time support messages from users or guests. This section will list all messages, allowing the admin to view and respond as needed.
- **Logout Button:** Allows the admin to securely exit the session.
- **Real-Time Updates:** Integrated with Firebase Realtime Database to ensure that messages are displayed and updated in real-time.
- **Purpose:**
 - To provide administrators with an organized view of user support messages.
 - To enable efficient communication between users and the admin.

Implementation Notes:

1. **Firebase Integration:**
 - Firebase SDK is used for both authentication and database management.
 - Firebase rules are configured to restrict user access to their own messages while allowing the admin to view all messages.
2. **Real-Time Communication:**
 - Messages sent by users are instantly visible to the admin through Firebase's real-time synchronization.
3. **Security:**
 - Authentication ensures that only authorized users (admin) can log in.
 - Firebase database rules prevent unauthorized access to sensitive data.



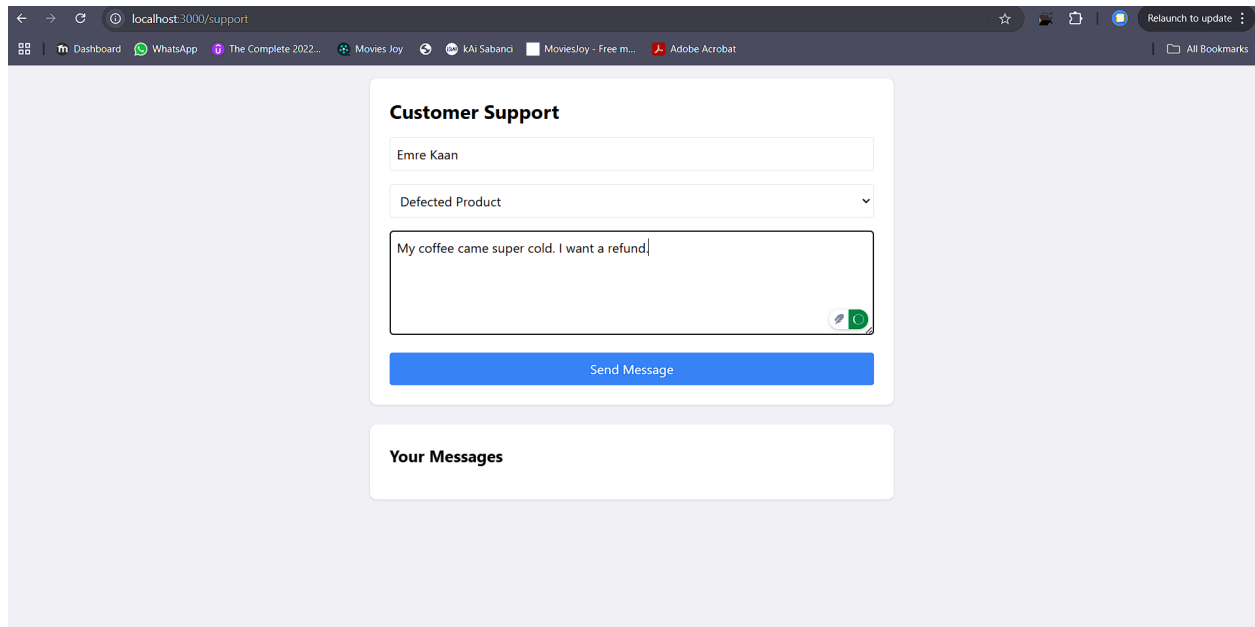
The provided screenshot showcases the dropdown menu for selecting the issue type on the user support page.

Features of the Dropdown:

1. **Purpose:**
 - Allows users or guests to specify the type of issue they are facing, ensuring that the admin can categorize and prioritize their support messages effectively.
2. **Options Available:**
 - **Select Issue** (default option): Guides the user to choose a relevant category.
 - **Defected Product**: For reporting damaged or defective products.
 - **Late Order**: For complaints about delayed deliveries.
 - **Lost Product**: For issues related to missing or undelivered items.
 - **Suggestion**: For users to provide feedback or recommendations.
3. **Design:**
 - A simple and user-friendly dropdown design to ensure accessibility and ease of use.
 - Dropdown options are clearly labeled to avoid confusion.

Implementation Notes:

1. **Backend Integration:**
 - The selected issue type is stored in Firebase Realtime Database or Firestore, along with the user's message and other details.
 - This categorization helps the admin quickly identify and respond to specific types of issues.
2. **Validation:**
 - Users are required to select an issue type before submitting their message, preventing incomplete or ambiguous submissions.
3. **Real-Time Updates:**
 - The issue type is instantly visible to the admin for prompt response.



The screenshot above displays the customer support page, designed for users or guests to communicate their issues directly to the admin.

Features of the Customer Support Page:

1. **User Input Fields:**

- **Name Field:** Allows users to input their name for personalized support.
- **Issue Dropdown:** Users can select the type of issue from predefined categories (e.g., Defected Product, Late Order).
- **Message Textbox:** Provides space for users to describe their issue or request in detail.

2. **Send Message Button:**

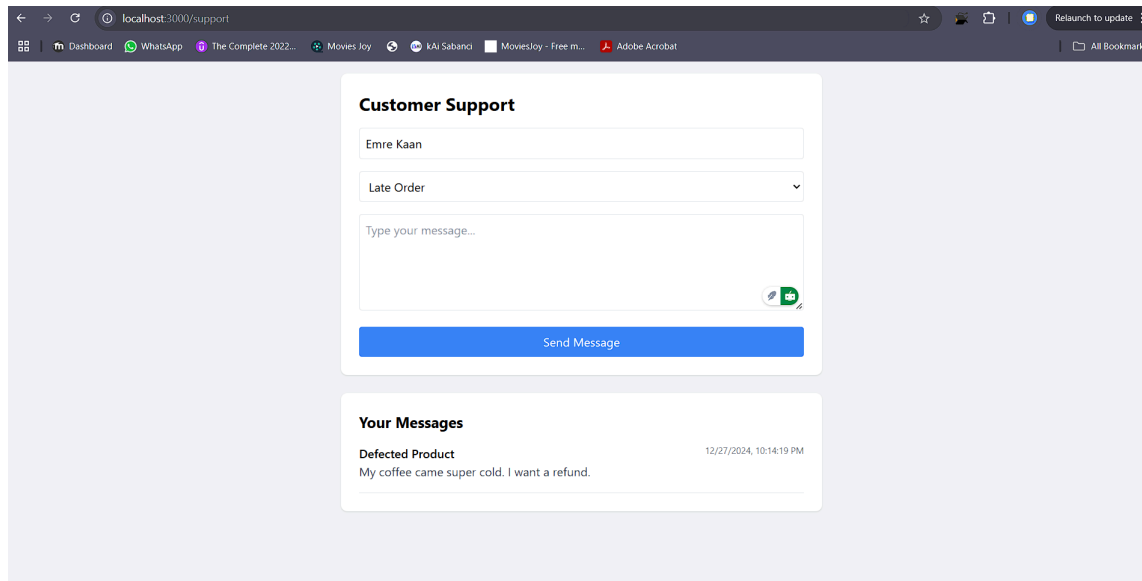
- Submits the user's message along with their name and selected issue to the Firebase database.

3. **Message History:**

- Displays the user's past messages in the "Your Messages" section, ensuring users can track their communication history.

Implementation Notes:

- **Real-Time Integration:** Messages are immediately synced with Firebase and visible to the admin for timely responses.
- **Validation:** Ensures that all required fields (name, issue type, message) are filled before submission.
- **User Experience:** Designed to be simple and user-friendly, with clear input sections and submission feedback.



The screenshot above demonstrates the customer support page with a previously submitted message displayed in the "Your Messages" section.

Features of the Customer Support Page:

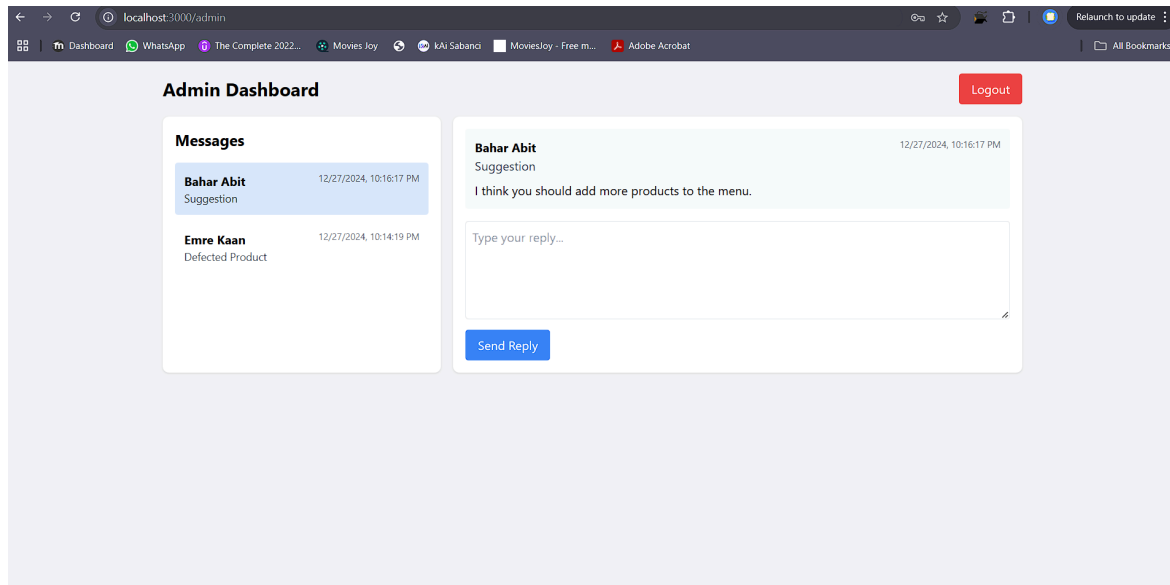
1. Input Fields:
 - Name Field: Allows the user to input their name for personalized communication.
 - Issue Dropdown: Provides predefined issue types (e.g., Defected Product, Late Order) for better categorization.
 - Message Textbox: Enables users to type their concerns or feedback.
2. Message Submission:
 - Clicking the Send Message button sends the user's details and message to Firebase in real time.
3. Message History:
 - Displays previously submitted messages under "Your Messages," including the issue type, message content, and timestamp.

Real-Time Updates:

- Submitted messages instantly appear in "Your Messages" and are visible to the admin for quick action.

Implementation Notes:

- Backend Integration: Firebase Realtime Database is used to store and retrieve messages.
- User Feedback: Clear and concise display of submitted messages enhances user experience.
- Timestamp: Each message is logged with a date and time for tracking purposes.



The screenshot above illustrates the admin dashboard, where all user messages are displayed and managed.

Features of the Admin Dashboard:

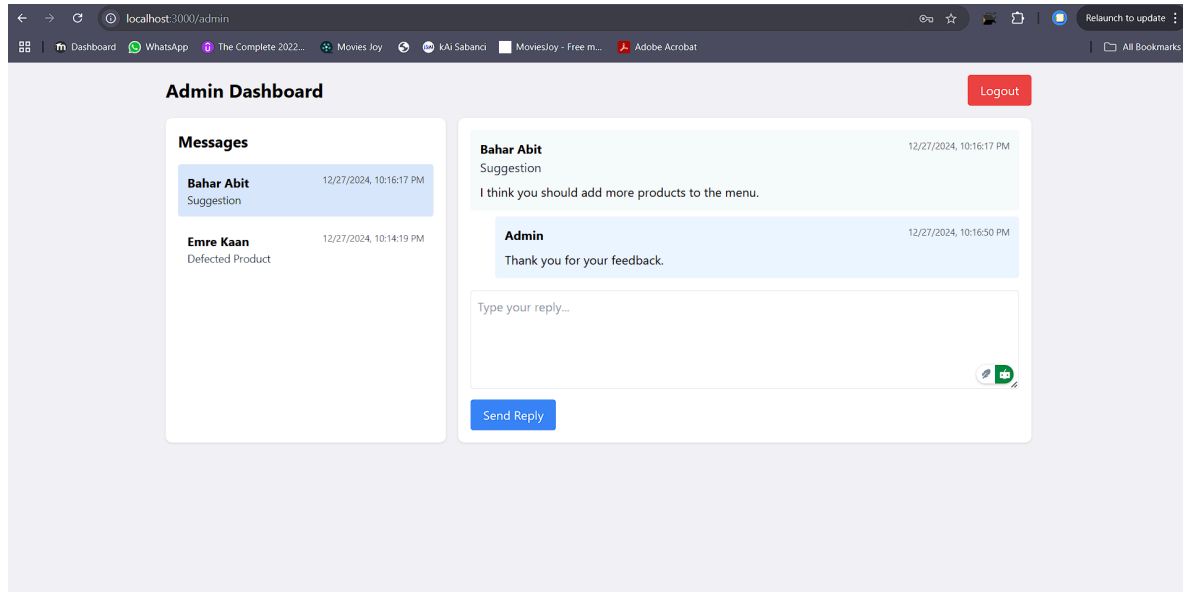
1. **Message List:**
 - Displays a list of messages from all users, showing the name, issue type, and timestamp for each message.
2. **Message Details:**
 - On selecting a message, its full content is displayed in a detailed view on the right.
 - The admin can view the message and respond directly from this interface.
3. **Reply Functionality:**
 - A text box allows the admin to type a response to the selected message.
 - The "Send Reply" button submits the admin's reply, which is sent in real time to the user's support page.
4. **Logout Button:**
 - Enables the admin to securely exit the dashboard.

Real-Time Updates:

- Messages sent by users are instantly visible in the admin dashboard.
- Admin replies are immediately synced to the user's support page for seamless communication.

Implementation Notes:

- **Firestore Integration:** The dashboard is fully integrated with Firestore, ensuring real-time synchronization of messages and replies.
- **User-Friendly Interface:** The layout provides a clear and intuitive experience for managing multiple user communications.
- **Security:** Only authenticated admins have access to this dashboard, ensuring secure message handling.



This screenshot showcases the admin dashboard with detailed message management and reply capabilities, enabling efficient communication with users.

Key Features:

1. **Message Overview:**

- A consolidated list of all user messages is displayed on the left, showing the sender's name, issue type, and timestamp.
- The admin can select any message to view its details and respond.

2. **Message Details Panel:**

- The selected message appears in the detailed view on the right, alongside the conversation history between the admin and the user.
- Ensures clarity by displaying both the user's initial message and the admin's responses.

3. **Reply Functionality:**

- The admin can compose and send replies directly using the reply input field.
- Replies are added to the conversation in real time, providing a seamless communication experience.

4. **Session Control:**

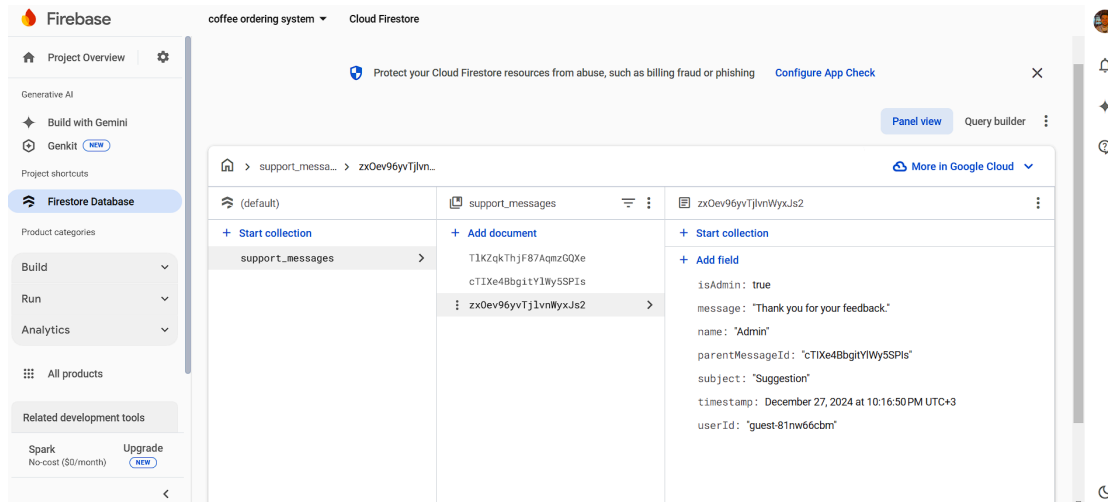
- The "Logout" button ensures secure termination of the admin's session after completing tasks.

Real-Time Synchronization:

- Messages and replies are synchronized instantly via Firebase Realtime Database.
- Users see admin replies immediately on their support page, facilitating quick resolutions.

Implementation Highlights:

- **User-Friendly Interface:** The dashboard layout organizes messages effectively, improving the admin's efficiency.
- **Message History:** Previous exchanges between the admin and users are displayed for reference.
- **Secure Communication:** Only authenticated admins can access and manage messages, ensuring data privacy.



The screenshot above illustrates the Firebase Firestore database configuration and data structure used for the real-time support page.

Key Features:

1. Collection: support_messages:

- The main collection (support_messages) stores all user and admin messages.
- Each document represents an individual message, uniquely identified by a generated ID.

2. Document Details:

- Fields include:
 - message: The content of the message (e.g., "Thank you for your feedback").
 - name: Indicates the sender (e.g., "Admin").
 - subject: Specifies the issue type (e.g., "Suggestion").
 - timestamp: Records the exact date and time of the message.
 - userId: Links the message to the corresponding user.

3. Real-Time Updates:

- The database is configured to provide instant synchronization between the support page and admin dashboard.
- Any changes, such as new messages or replies, are reflected in real time.

4. Admin Flag (isAdmin):

- A boolean field (isAdmin) distinguishes messages sent by the admin (**true**) from those sent by users.

Implementation Notes:

- **Secure Data Management:**
 - Firestore rules are applied to ensure users can only access their messages, while admins have full access.
- **Structured Data:**
 - The database design ensures scalability and efficient query handling for future extensions.
- **Timestamped Messages:**
 - Each message is timestamped to maintain chronological order and improve traceability.

Benefits of Firebase Integration:

- Seamless real-time communication between users and admins.
- Easy scalability to handle an increasing number of messages.
- Enhanced security with user-specific access rules.