CS306 PHASE 3 PROJECT REPORT

İdil BAYAR 32267

Dilara ALKANALKA 30758

This project report details the implementation of a real-time support system by using Firebase, Xampp and VisualStudio. The system enables the users and administrators to communicate efficiently while relying on PHP for server-side interaction and requiring form submissions to update messages with Firebase.

The core functionality revolves around:

User Support Page:

- This page allows users to submit their support requests. Users enter their name and select subject to resolve or get help for example like “Flight Cancellation", "Delayed Flight", "Lost Baggage".
- Messages are then stored in Firebase Realtime Database by its capability to provide real-time data synchronization across clients and also timestamped, which is defined and enabled in rules of Firebase.
- Passengers can also view the previous message logs by entering their name.

Admin Support Page:

- This page enables airline staff or administrators to view all passenger messages and respond to them in real-time.
- Administrators can send replies to specific support requests that are stored in the Firebase Realtime Database and visible to the related passenger.

Implementation details:

The project uses Firebase Realtime Database, by its ability to provide real-time data storage and retrieval. Due to use of PHP for sever-side processing, real-time order may not be fully implemented without needing a refresh or submission but the system ensures that all messages are stored in Firebase with timestamps to maintain chronological order. These timestamps are used for displaying and sorting messages for both the user interface and administration panel.

A key implementation in design was to generate a unique used ID for each passenger based on their name through a function. This implementation ensures user privacy as users can only access their own messages while the admins can access all messages submitted.

For simplification of development and enabling efficient communication, the rules within the Firebase were configured with open read and write access to the messages node of the database. However, in future implementations for a real -world scenario, stricter Firebase rules and a robust authentication system can be implemented. Also, transitioning to Firebase SDK- based JavaScript listener functions could enable real-time updates without page refreshes and could be done for a smoother user interface and admin panel.

The implementation consists of 7 PHP files and a CSS file to create 2 webpages; one for users to submit and view their support messages, and another for the admin to manage and respond to those messages. PHP files are used to handle the backend logic of data submission, retrieval, and communication with the Firebase Realtime Database. Styles.css is used to style the user and admin pages, ensuring a clean, user-friendly, and visually appealing design. The further description of these PHP files as follows:

- index.php: This file serves as the main user interface for the support system. It provides a form for users to input their name, select the subject of their issue, and type their message. Additionally, with a section for the users to see previously submitted messages with their name. The form submits user input to submit_message.php for processing.
- submit_message.php: This file processes the data submitted by users through index.php. It generates a unique ID for each user (based on their name) and stores the submitted message in the Firebase Realtime Database under this ID. The message is saved along with its subject, timestamp, and the user's name. If the user ID does not exist, a new entry is created, grouping all messages under the corresponding user.
- receive_message.php: This file retrieves and displays the messages submitted by a specific user. It takes the user's name as input, computes the unique ID, and fetches all messages associated with that ID from the Firebase database. Replies to these messages are also fetched and displayed alongside the messages, maintaining proper chronological order.
- admin.php: This file provides the admin interface for the support system. It displays all messages from all users in one place, along with their timestamps, subjects, and content. Each message also includes a form for the admin to reply. The admin can view all replies associated with a specific message.
- admin_get_messages.php: This file fetches all user messages for display on the admin interface (admin.php). Messages are grouped by the unique user IDs and are retrieved from the Firebase Realtime Database. Replies to these messages are also fetched and displayed in chronological order.
- admin_send_reply.php: This file handles admin replies to user messages. When the admin submits a reply through the form on admin.php, this file processes the reply and appends it to the appropriate message thread in the Firebase database. The reply includes the content, timestamp, and sender information as "admin".
- firebase_config.php: This file initializes the connection to the Firebase Realtime Database. It uses the Firebase SDK to authenticate with the Firebase project and provides access to the database for all other PHP scripts.
- styles.css: This file contains the styling for the support and admin pages. It ensures a user-friendly and visually appealing interface for both users and admins.

The screenshots of the support pages can be found below:

Admin Support Panel

tolgatolga - 12/29/2024, 9:44:15 PM

möööö

Reply to this me Send Reply

admin: nwww - 12/29/2024, 9:44:30 PM

bora - 12/29/2024, 9:43:17 PM

oooo

Reply to this me Send Reply

admin: pppp - 12/29/2024, 9:43:55 PM

bora - 12/29/2024, 9:42:52 PM

vvvv

Reply to this me Send Reply

admin: bbbb - 12/29/2024, 9:43:01 PM

Support Page

Your Name: tolgatolga
Subject: Suggestion
Your Message: Enter your messa
Send Message
View Your Messages

Your Messages:

tolgatolga (Suggestion) - 12/29/2024, 9:44:15 PM

möööö

admin: nwww - 12/29/2024, 9:44:30 PM