LEAK PEAK PROJECT REPORT

GROUP MEMBERS:
MUHAMMAD SOHAIB (22K-4751)
ABDUL HADI KHAN (22K-4724)
ZEHRA QURESHI (22K-4744)

Project Overview:

Introduction

The "Leak Peak" project is a comprehensive web application designed to manage user information, subscriptions, and monitor data breaches. It aims to provide a secure platform for users and administrators, allowing them to track user activities, manage subscriptions, and identify potential security threats. This project leverages modern web technologies and follows best practices in security and database management.

Objectives

- To create a user-friendly interface for users and administrators to manage accounts and subscriptions.
- To implement robust authentication and authorization mechanisms using JWT and session management.
- To track user activities through logging and provide detailed reports for security audits.
- To monitor and manage subscription plans, ensuring users are informed about renewals and payment details.
- To maintain a record of compromised accounts and breaches, providing users with insights into their security status.

Technology Stack

- Frontend: HTML, CSS (Tailwind CSS), JavaScript
- Backend: Node.js, Express.js
- Database: MySQL
- Authentication: JSON Web Tokens (JWT), Express-Session

- Email Notifications: Nodemailer
- Task Scheduling: Node-Cron

Key Features

1. User Management:
    - Registration and login functionality for users and administrators.
    - Role-based access control to differentiate between user and admin functionalities.
2. Subscription Management:
    - Ability to view, manage, and renew subscription plans.
    - Integration of payment details for premium users, including credit card information.
3. User Activity Logs:
    - Recording of user actions for auditing and monitoring purposes.
    - A dedicated page to display user logs, accessible by authorized personnel.
4. Breach Monitoring:
    - Tracking of compromised accounts and related breaches.
    - Users can check if their accounts have been involved in any data breaches.
5. Email Notifications:
    - Automated email reminders for subscription renewals.
    - Alerts for users regarding potential security threats related to their accounts.
6. Admin Management:
    - Admins can manage users, view logs, and handle administrative tasks such as adding or deleting accounts.

Database Design

The database for the "Leak Peak" project is designed using an Entity-Relationship (ER) model, which includes the following entities and relationships:

- Entities:
    - Users: Stores user information including email, username, password, and user category.
    - Admins: Contains administrator details with their role and credentials.

    - User _Logs: Records actions performed by users along with timestamps.
    - Subscriptions: Manages user subscriptions, including plan details and renewal dates.
    - Breach_Types: Defines various types of data breaches.
    - Compromised_Accounts: Records accounts that have been compromised.
    - Subscription_Plans: Holds information about different subscription offerings.
    - Breaches: Details specific breaches, including their type and source.
    - Countries: Lists countries associated with users and credit cards.
    - Credit_Cards: Manages credit card information linked to users.
- Relationships:
    - Users can have multiple logs, subscriptions, and compromised accounts.
    - Admins can manage multiple users and track deletion events.
    - Subscriptions are linked to specific subscription plans, and breaches are categorized by type and source.

User Interface

The user interface is designed to be intuitive and responsive, utilizing Tailwind CSS for styling. Key pages include:

- Home Page: Introduction to the application and navigation to different sections.
- User Logs Page: Displays logs of user activities in a structured table format.
- Admin Management Page: Allows admins to manage user accounts and view logs.
- Subscription Management Page: Enables users to view and manage their subscription plans.

Security Considerations

The project implements several security measures to protect user data:.

- JWT is used for authentication, ensuring secure communication between the client and server.
- CORS is configured to restrict access to the API from unauthorized origins.
- Regular monitoring of user activities helps identify suspicious behavior.

Project Flow:

The flow of the "Leak Peak" project outlines the sequence of operations and interactions between users, administrators, and the system. This section provides a detailed description of how the application functions from user registration to breach monitoring, ensuring a clear understanding of the user journey and system processes.

1. User Registration and Authentication

- User Registration:
  - Users visit the registration page and fill out a form with their email, username, and password.
  - Upon submission, the system validates the input and creates a new user entry in the database..
- User Login:
  - Users access the login page and enter their credentials (email and password).
  - The system checks the credentials against the database. If valid, a JWT is generated and stored in local storage for session management.
  - Users are redirected to the dashboard or home page upon successful login.

2. User Dashboard

- After logging in, users access their dashboard, which displays:
  - Overview of their account status.
  - Subscription details, including the current plan and renewal dates.
  - Links to view user logs and breach monitoring.

3. Subscription Management

- Viewing Subscriptions:
    - Users can navigate to the subscription management section to view available plans and their features.
    - Users can upgrade, downgrade, or renew their subscriptions as needed.
- Payment Processing:
    - For premium subscriptions, users enter their credit card information securely.
    - The system processes the payment and updates the subscription status in the database.

4. User Activity Logging

- Action Tracking:
    - Every action performed by users (e.g., login, logout, subscription changes) is recorded in the User_Logs table.
    - Each log entry includes the user ID, action type, and timestamp for auditing purposes.
- Viewing Logs:
    - Users can view their activity logs on the user logs page, where actions are listed in a structured table format.

5. Breach Monitoring

- Breach Checks:
    - Users can check if their accounts have been compromised by querying the Compromised_Accounts table.
    - The system compares user email addresses with known compromised accounts and displays relevant information.
- Alerts and Notifications:

    - If a breach is detected that affects a user, the system sends an alert via email, informing them of the potential risk and recommended actions.

6. Admin Management

- Admin Login:
    - Administrators log in through a secure admin portal using their credentials.
    - Admins receive a higher level of access to manage user accounts and view all user logs.
- User Management:
    - Admins can view, edit, or delete user accounts as necessary.
    - Admin actions are logged in the Admin_Deletion table to maintain a record of changes.
- Monitoring User Activity:
    - Admins can access a comprehensive view of user logs to monitor user activities and identify any suspicious behavior.

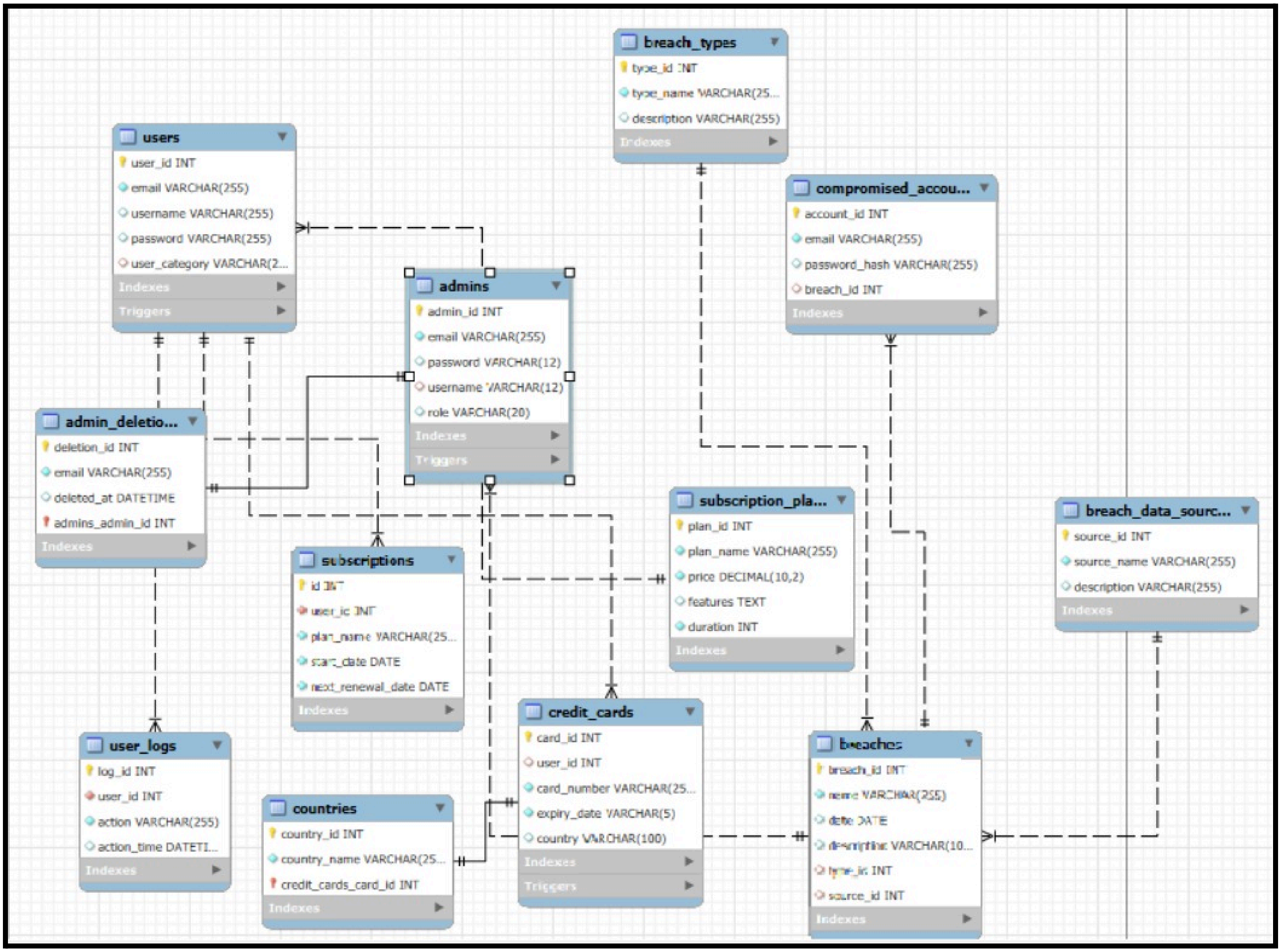
7. Data Breach Management

- Recording Breaches:
    - When a data breach occurs, details are recorded in the Breaches table, including the type of breach, source, and affected accounts.
    - The system categorizes breaches by type and source for easier analysis.
- Breach Reports:
    - Admins can generate reports on breaches to assess the impact and take necessary actions to enhance security measures.

8. User Logout

- Users can log out from the application, which invalidates the JWT stored in local storage.
- The system clears user session data, ensuring that sensitive information is not accessible after logout.

ER Diagram:


breach_types
type_id INT
type_name VARCHAR(25...
description VARCHAR(255)
Indexes
users
user_id INT
email VARCHAR(255)
username VARCHAR(255)
password VARCHAR(255)
user_category VARCHAR(2...
Indexes
Triggers
compromised_accou...
account_id INT
email VARCHAR(255)
password_hash VARCHAR(255)
breach_id INT
Indexes
admins
admin_id INT
email VARCHAR(255)
password VARCHAR(12)
username VARCHAR(12)
role VARCHAR(20)
Indexes
Triggers
admin_deletio...
deletion_id INT
email VARCHAR(255)
deleted_at DATETIME
admins_admin_id INT
Indexes
subscription_pla...
plan_id INT
plan_name VARCHAR(255)
price DECIMAL(10,2)
features TEXT
duration INT
Indexes
breach_data_sourc...
source_id INT
source_name VARCHAR(255)
description VARCHAR(255)
Indexes
subscriptions
id INT
user_id INT
plan_name VARCHAR(25...
start_date DATE
next_renewal_date DATE
Indexes
credit_cards
card_id INT
user_id INT
card_number VARCHAR(25...
expiry_date VARCHAR(5)
country VARCHAR(100)
Indexes
Triggers
breaches
breach_id INT
name VARCHAR(255)
date DATE
description VARCHAR(10...
type_id INT
source_id INT
Indexes
user_logs
log_id INT
user_id INT
action VARCHAR(255)
action_time DATETI...
Indexes
countries
country_id INT
country_name VARCHAR(25...
credit_cards_card_id INT
Indexes