Modern Application Development- V2

Student Details

Name: Nachiket Kishor Gore
Student ID: 23f1000542
Email: 23f1000542@ds.study.iitm.ac.in

Project Details-

Title:

Vehicle Parking System Application - V2

Objective:

To develop a multi-user platform that provides efficient parking management and solutions, enabling easy booking, tracking, and administration of parking slots for users and admins.

Problem Statement:

The Vehicle Parking System Application aims to solve the challenges of urban parking management by allowing users to easily search for parking lots, book slots, and monitor their reservations. The app automates repetitive tasks such as sending booking reminders, generating monthly activity reports, and updating slot statuses. Admins can efficiently manage parking lots, monitor user activity, and ensure optimal utilization of parking resources.

Approach:

1. Admin Roles

- Implement role-based access control (RBAC) for secure management.
- Admins can create, update, and delete parking lots and slots.
- Admins monitor all user bookings, transactions, and parking history.
- Admins can view analytics dashboards, export data, and manage penalties for unattended bookings.

2. User Interaction

- Users can search for parking lots by country, city, address, pincode, or prime location.
- Users can book slots, view booking history, and manage their reservations.
- Users receive automated reminders before and after their booking time.
- Users can view wallet balance, top up, and track transaction history.

3. Automated Backend Jobs

- Scheduled jobs send booking reminders to users.
- Monthly activity reports are generated and emailed to users, including spend analytics and booking trends.
- Admin and Users can export parking and transaction history in CSV format.

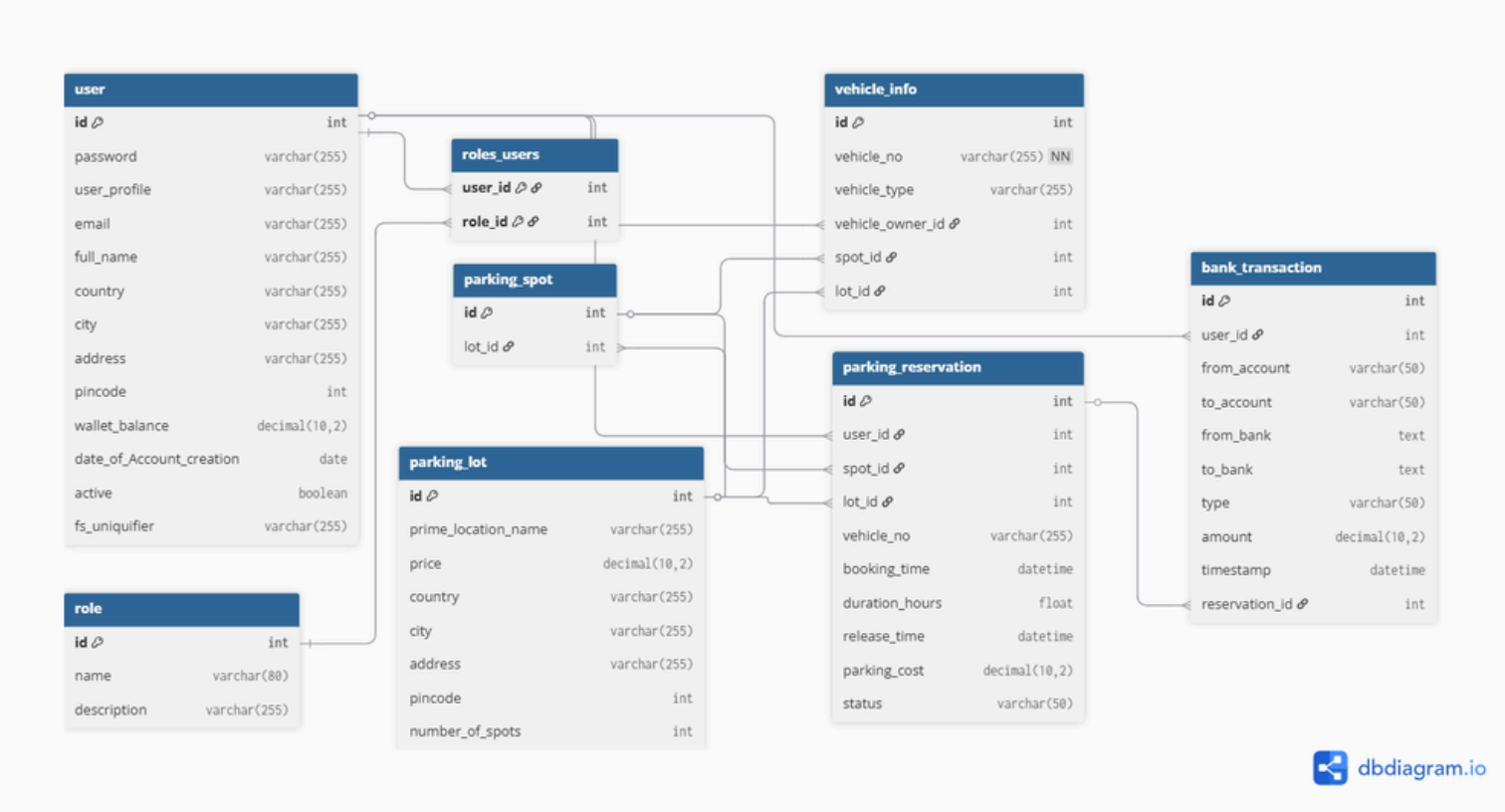
4. Performance and Caching

- Redis is used for caching frequently accessed data to improve API performance.
- Cache expiry mechanisms ensure data freshness for real-time slot status and analytics.

Frameworks and Libraries Used:

- **Flask:** Backend API, routing, and business logic.
- **VueJS:** Frontend user interface and dynamic content rendering.
- **SQLite:** Database for users, parking lots, slots, reservations, and transactions.
- **Redis:** Caching for performance optimization.
- **Celery & Celery Beat:** Background job processing and scheduled tasks (reminders, reports).
- **Bootstrap & CSS:** Responsive layouts and modern UI styling.
- **ChartJS:** Graphical reports and statistics for dashboards.
- **Flask-Security:** Token-based and role-based authentication.
- **Flask-Mail:** Email notifications for reminders and reports.
- **Jinja2**: HTML template rendering (for emails and reports).

ER Diagram:



Presentation:

https://drive.google.com/file/d/16Kj-Ijj4djGfQc0O20qlry1VHYWaXrqR/view?usp=drive_link

Key Features:

- User Dashboard: View monthly spends, booking trends, status distribution, and location analytics.
- Admin Dashboard: Monitor revenue, booking status, country/city/address analytics, and best customers.
- Parking Management: Create, edit, and delete lots and slots; view slot status and booking history.
- Booking & History: Users can book slots, view history, update status, and apply penalties.
- Wallet & Transactions: Top up wallet, view transaction history, and manage payments.
- Automated Reminders & Reports: Email notifications for upcoming bookings and monthly activity summaries.
- Export & Analytics: Export booking and transaction data; view charts and rankings.

Conclusion:

The Vehicle Parking System Application provides a robust solution for urban parking challenges, streamlining the booking process, improving resource utilization, and empowering both users and admins with real-time data and automation. The use of modern frameworks and technologies ensures scalability, security, and a professional user experience.