

Vehicle Parking App – V2

Modern Application Development II

Author:

SUVRAT SURANA

21F1000047

21f1000047@ds.study.iitm.ac.in

Introduction:

It is a multi-user app (one administrator and other users) that manages different parking lots, parking spots and parked vehicles (assuming that the app is only for 4-wheeler parking).

The system supports vehicle spot booking, occupancy tracking with timestamps (Parked In / Release), real-time cost calculation, CSV report generation, and visual summaries. Admins can manage lots, users, and gain insights through analytical dashboards.

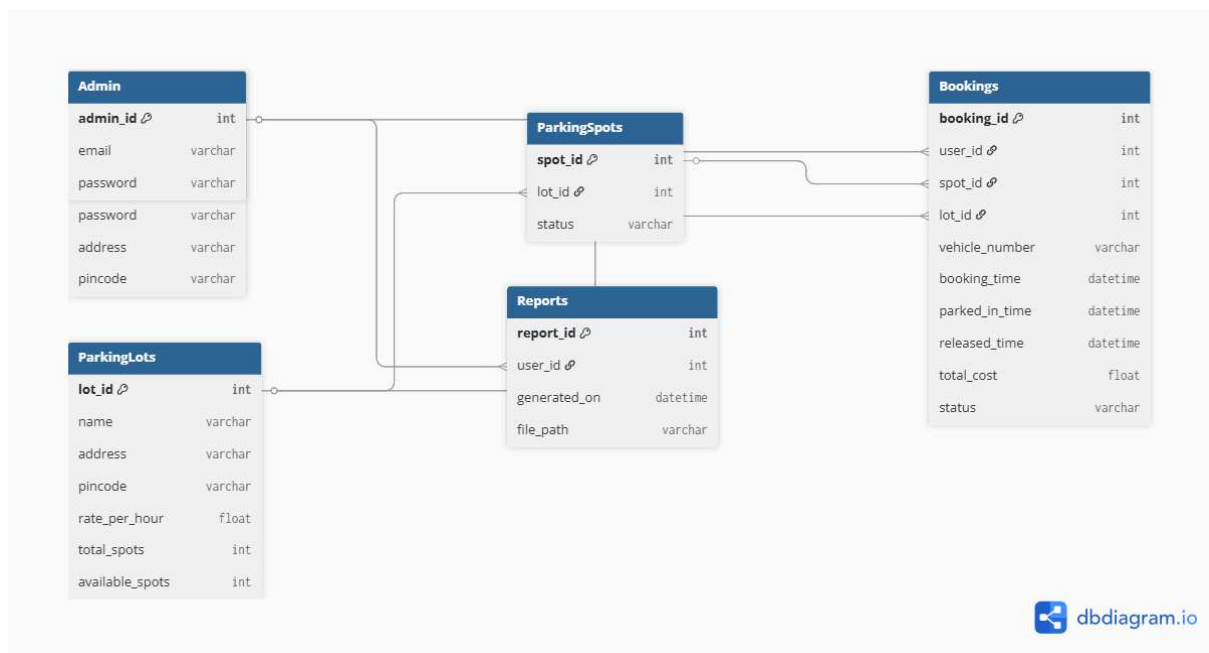
User Features:

- Users must sign up using a user-friendly registration form.
- Role-Based-Access-Control is implemented to restrict user privileges.
- Users can search for parking lots using pin code, city, or location keywords.
- Book parking spots by providing a vehicle number. Once booked, they can:
 - Click "**Parked In**" to mark the start time.
 - Click "**Release**" to mark exit time and compute parking cost.
- Users can view parking history and export it as a .csv file.
- User data is visualized via bar charts showing:
 - Locations used
 - Duration parked
 - Total expense by vehicle

Admin Features:

- Admin can create, edit, and delete parking lots.
- Deletion is restricted if any spot in the lot is currently occupied.
- Admin manages the number of spots in each parking lot.
- Admin can check:
 - Is Parking Spot Occupied/Available
 - Parked-in timestamp
 - Duration of stay
- Full access to user data: email, name, address etc.
- Revenue data is visualized:
 - Pie charts for revenue by location
 - Bar charts for number of Occupied and Available Parking Spots
 - Monthly Report Available

ER Diagram:



Technology:

- **Backend:** Python, Flask
- **Frontend:** HTML, CSS, Jinja2, Bootstrap
- **Database:** SQLite with SQLAlchemy (ORM)
- **Task Queue:** Celery
- **Background Workers:** Redis
- **Email:** Flask-Mail + MailHog (SMTP)
- **Visualization:** Chart.js
- **Security:** Flask-JWT-Extended for token-based security and RBAC
- **Other:**
 - Flask-Caching (Redis)
 - Marshmallow (serialization)
 - Pandas (CSV export)

End-Point Reseources:○ **Admin Endpoints**

- Ø GET /admin/dashboard` - Dashboard statistic
- Ø GET /admin/parking-lots` - List all lots
- Ø POST /admin/parking-lots/create` - Create new lot
- Ø POST /admin/parking-lots/{id}/edit` - Edit lot
- Ø POST /admin/parking-lots/{id}/delete` - Delete lot
- Ø GET /admin/users` - List all users
- Ø POST /admin/users/{id}/toggle-status` - Toggle user status
- Ø GET /admin/bookings` - List all bookings
- Ø POST /admin/bookings/{id}/cancel` - Cancel booking
- Ø GET /admin/analytics` - Analytics data

○ **User Endpoints**

- Ø POST /register` - User registration
- Ø POST /login` - User login
- Ø GET /user/dashboard` - User dashboard
- Ø POST /user/reserve/{lot_id}` - Reserve parking spot
- Ø POST /user/end-session` - End parking session
- GET /user/history` - Booking history

Presentation Video:**[Presentation Video](https://drive.google.com/file/d/1murBhruCtIQ5E1wiaYUSvtBaGgyaah8T/view?usp=sharing)**

<https://drive.google.com/file/d/1murBhruCtIQ5E1wiaYUSvtBaGgyaah8T/view?usp=sharing>