PRESENT BY : MARYAM SALEEM

MENTOR : SIR HAMZAH SYED

RENT A CAR WEBSITE DOCUMENTAION

Frontend:

- Home Page
- Search Page
- Booking Page
- Payment Page
- User Dashboard
- Admin Dashboard (for car and booking management)

API Layer:

- Handles requests and responses between frontend and backend

Backend:

- User Authentication
- Car Management
- Booking Management
- Payment Processing

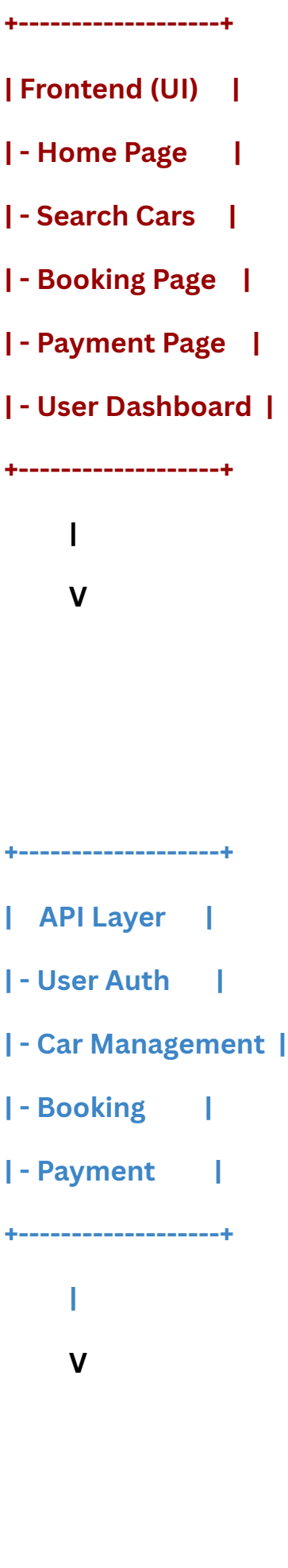
Database:

- Customer Information
- Car Details
- Bookings
- Payment Records

Payment Gateways:

- Stripe
- EasyPaisa

-
- **STRUCTURE**


+------------------+
| Frontend (UI) |
| - Home Page |
| - Search Cars |
| - Booking Page |
| - Payment Page |
| - User Dashboard |
+------------------+
|
v
+------------------+
| API Layer |
| - User Auth |
| - Car Management |
| - Booking |
| - Payment |
+------------------+
|
v
+------------------+


| Backend (Server) |
| - User Auth |
| - Car Management |
| - Booking |
| - Payment Proc. |
+------------------+
|
V
+------------------+
| Database |
| - Customer Table |
| - Car Table |
| - Booking Table |
| - Payment Table |
+------------------+
|
V
+------------------+
| Payment Gateways |
| - Stripe |
| - EasyPaisa |
| - JazzCash |
+------------------+

Document: Car Rental Website Structure

This document defines the structure of your car rental website, explaining the flow of the frontend ,backend, API, and payment method. It will help you understand how different components interact and how your website will function.

1. Frontend (User Interface):The frontend is the part where customers interact. It should be user-friendly and responsive.

Key Features:

- Home Page: Introduction to the website and featured cars.

- Search Cars Page: Customers can search and filter cars.

- Booking Page: Customers can book cars.

- Payment Page: Secure payment options.

- User Dashboard: Customers can view their bookings and details.

Technologies:

- React.js: For building dynamic and interactive user interfaces.

- Next.js: For server-side rendering and static site generation.

- Tailwind CSS: For responsive and modern styling.

2. Backend (Server-Side Logic):

The backend handles business logic and data processing.

Key Features:- User Authentication: Register/login functionality.

- Car Management: Add/update/delete cars.

- Booking Management: Create/update/cancel bookings.

- Payment Processing: Secure payment integration.

Technologies:- Node.js: For backend development.

Express.js: For API development.

Database: MySQL or MongoDB for data storage.

---

3. API (Application Programming Interface):

The API handles communication between the frontend and backend.

Key Endpoints:

1. User Authentication:

  - `POST /api/auth/register`: To register a new user.

  - `POST /api/auth/login`: To log in a user.

2. Car Management:

  - GET /api/cars: To fetch available cars.

  - `POST /api/cars`: To add a new car (admin only).

3. Booking Management: - `POST /api/bookings`: To create a new booking.

  - `GET /api/bookings`: To fetch a user's bookings.

4. Payment Processing:

  - `POST /api/payment`: To process payments.

---

4. Payment Method:The payment method should be secure and reliable.

- Stripe: For global payments.

- EasyPaisa/JazzCash: For local payments.

DETAIL

1. The customer enters payment details on the payment page.

2. Payment details are processed via the backend API.

3. The payment gateway (e.g., Stripe) completes the payment.

4. The payment status is saved in the database.

5. Database Structure:All data is stored in the database.

Tables:

1.Customer Table:

- `customer_id`, `name`, `email`, `password`, `created_at`.

2. Car Table:

- `car_id`, `model`, `brand`, `year`, `color`, `rental_price`, `availability_status`.

3. Booking Table:

- `booking_id`, `customer_id`, `car_id`, `start_date`, `end_date`, `total_price`, `booking_status`.

4. Payment Table:

- `payment_id`, `booking_id`, `payment_method`, `amount`, `payment_status`, `payment_date`.

6. Website Flow:1.

Customer Flow:

- The customer visits the website, registers/logs in.
- Searches for a car and books it.
- Makes a payment and receives confirmation.

2. Admin Flow:

- The admin adds/updates cars.
- Manages bookings and payments.

7. Security Measures:

- SSL Certificate: For secure data transmission.
- Data Encryption: Encrypt sensitive data (e.g., passwords, payment info).
- Authentication: Secure user authentication (e.g., JWT tokens).
- Regular Backups: To prevent data loss.

8. Deployment:

- Hosting Platform: Vercel (for Next.js apps)
- Domain Name: Register a custom domain

Conclusion:

The structure of your car rental website is clear and organized, with defined flows for the frontend, backend, API, and payment method. This structure will help you build a scalable and efficient website.