

Technical Planning for a venue FITNESS RENTAL PLATFORM

Creating a fitness venue rental platform involves designing and developing a system that allows users to book fitness spaces such as gyms, yoga studios, and sports hall. Below is a structured technical plan.

a) Define Technical Requirement.

1) User Registration & Authentication.

- Registration for venue owners and customers.
- Social media & email login options

2) Venue Listings:-

- Detailed fitness venue profiles (Photos, amenities, equipment, capacity, location).

3) Booking System:-

- Real time availability calendar
- Booking options for hourly, daily or subscription-based rental.

4) Payment integration:-

- Secure payment processing
- Dynamic pricing option based on peak hours.

5) Admin Panel.

- Manage users, booking, venues & payment

6) Notifications:-

Email and SMS alerts for booking, cancellations, & payment confirmations.

7) Advanced Features.

1) Dynamic Scheduling:-

- Integration with fitness class schedule

2) Membership Management:-

- Subscription option for regular customers

3) Live Availability

- Booking with Google calendar or other third party tool.

4) Analytics Dashboard:-

- Provide data insights for venue owners. (eg Peak booking hours revenue)

5) Mobile App:-

- Dedicated apps for iOS & Android for seamless access.

2. Design System Architecture:-

Frontend:-

- Technology: React.js, Next.js, or

Angular for a responsive user interface

Features:

- 1 Interactive Search and filtering.
- 2 Dynamic booking calendar.
- 3 User-friendly dashboards for venue owners and customers.

Backend:

Technology: Node.js with Express.js

or Django (Python).

Core Modules:

- 1 User authentication and management.
- 2 Venue management system.
- 3 Payment and booking system.
- 4 Notification service.

3 Plan API Requirements:

Authentication APIs:-

- Post / register: User registration.
- Post / login: User login.
- GET / Profile: Fetch user details

Venue APIs:

- GET / venues: Retrieve a list of venues.
- GET / venues / :id: Fetch specific venue details
- Post / venues: Add a new venue (venue owner)
- PUT / venues / :id: Update venue details
- DELETE / venues / :id: Remove a venue.

Booking APIs:

- POST / bookings: Create a booking.
- GET / bookings: Retrieve user bookings

- PUT/bookings/:id: Update booking status.

Payment APIs:

- POST/Payments: Process a Payment.
- GET/Payments: Retrieve Payment history.

Admin APIs:

- GET/users: Retrieve all user data.
- GET/reports: Generate financial and usage reports.

4. Write Technical Documentation:

Components of Documentation:

System Overview:-

- High-level architecture of the Platform.
- API Documentation:
 1. Endpoint details (e.g. URL, methods, payload and responses).
 2. Use tools like Swagger or Postman for API visualization.

- 3 Database Schema:
ER diagrams showing relationships between users, venues and bookings.

5 Venue Owner Workflow:

Venue Setup:

Owner registers and adds venue details.

6 Booking Managements:

Owner accepts or declines booking.

Analytics:

Owner views insights (e.g revenue, booking trends).

Collaborate and Refine:

- Conduct workshops with stakeholders
- Build an MVP for early testing
- Use feedback to refine functionalities and workflows.

Deployment and Maintenance:

Deployment:

- Use cloud platforms like AWS,

Azure, or Google Cloud.

- Ensure CI/CD pipelines for seamless updates.

Maintenance:-

- Regularly monitor system performance
- Update features based on user feedback.
- Regularly review progress.
- Prototyping Build a basic prototype.