FitConnect - Personal Trainer Platform

Project Specification for TestSprite Testing

🎯 Project Overview

FitConnect is a comprehensive full-stack web application that connects personal trainers with clients, featuring intelligent booking, payment processing, and session management. This is a university project with simulated payment processing for educational purposes.

Tech Stack:

- **Backend:** FastAPI (Python), MySQL, SQLAlchemy, JWT Authentication

- **Frontend:** Next.js 14, TypeScript, Tailwind CSS

- **Authentication:** JWT Bearer tokens

- **Database:** MySQL with normalized schema

👥 User Roles & Authentication

User Types:

1. **Client** - Books training sessions, manages programs, makes payments

2. **Trainer** - Manages availability, approves bookings, creates programs

3. **Admin** - System management, analytics, refunds

Authentication System:

- **Type:** JWT Bearer Token Authentication

- **Login Endpoint:** `POST /api/auth/login`

- **Registration:** `POST /api/auth/register`

- **Token Format:** `Authorization: Bearer <jwt_token>`

- **Token Expiration:** 30 minutes

Test Credentials:

```
Client: client@example.com / password123

Trainer: trainer@example.com / password123

Admin: admin@example.com / password123
```

** 🏗️ Core Features to Test**

1. User Management

- User registration with role selection
- Login/logout functionality
- Profile management
- Role-based access control

2. Trainer Profiles

- Trainer profile creation and completion
- Specialty selection (Strength Training, Weight Loss, Yoga, etc.)
- Pricing setup (per hour/session)
- Bio, experience, certifications
- Availability management

3. Booking System (3 Methods)

A. Direct Booking

- Browse available trainers
- View trainer time slots

- Book specific time slot
- Trainer approval/rejection workflow

B. Booking Requests
- Submit flexible booking requests
- Trainer reviews and selects optimal time
- Confirmation workflow

C. Smart Scheduling (AI-Powered)
- Enter preferences (dates, times, duration)
- AI algorithm finds optimal matches
- Ranked suggestions presented
- Book from suggestions

4. Session Management
- Session lifecycle tracking
- Mark sessions as complete/cancelled
- Session history and analytics
- Progress tracking

5. Payment System (Simulated)
- Credit card processing (simulated)
- Payment validation
- Transaction tracking
- Payment history
- Refund processing

Test Payment Cards:
- Visa: `4111 1111 1111 1111`
- Mastercard: `5555 5555 5555 4444`
- Amex: `3782 822463 10005`

6. Messaging System

- In-app messaging between clients and trainers
- Conversation management
- Message status tracking
- Real-time communication

7. Workout Programs

- Trainers create custom programs
- Program assignment to clients
- Progress tracking
- Program management

8. Analytics Dashboard

- Session analytics
- Client/trainer insights
- Payment statistics
- Performance metrics

** 🔗 Key API Endpoints**

Authentication

```
POST /api/auth/login
POST /api/auth/register
GET /api/auth/me
POST /api/auth/logout
```

Trainers

```
GET /api/trainers
GET /api/trainers/{id}
POST /api/trainers
```

Booking Management

```
POST /api/booking-management/booking-request
GET /api/booking-management/booking-requests
POST /api/booking-management/approve-booking
POST /api/booking-management/reject-booking
GET /api/booking-management/my-bookings
```

Smart Scheduling

```
POST /api/bookings/smart-booking
POST /api/bookings/optimal-schedule
POST /api/bookings/greedy-optimization
```

Payments

```
POST /api/payments/
GET /api/payments/my-payments
GET /api/payments/stats
POST /api/payments/refund
```

Time Slots

```

GET /api/time-slots/trainer/{id}/available

POST /api/time-slots/bulk-create

POST /api/time-slots/book

```

** 🎯 Test Scenarios Priority**

High Priority (Core Functionality)

1. **Complete Booking Flow:**

   - Client login → Browse trainers → Book session → Trainer approval → Payment → Session completion

2. **Authentication Flow:**

   - Registration → Login → Token validation → Role-based access

3. **Payment Processing:**

   - Booking confirmation → Payment form → Transaction processing → Payment history

Medium Priority (Advanced Features)

1. **Smart Scheduling:**

   - Enter preferences → AI suggestions → Book optimal time

2. **Trainer Management:**

   - Profile setup → Availability setting → Booking approval

3. **Messaging System:**

   - Send messages → Conversation management → Message status

Low Priority (Analytics & Reports)

1. **Analytics Dashboard:**

   - View statistics → Generate reports → Performance metrics

---

** 🧪 Testing Requirements**

Authentication Testing:

- Valid/invalid login credentials
- Token expiration handling
- Role-based access control
- Protected route access

Booking Flow Testing:

- All three booking methods
- Trainer approval/rejection workflows
- Time slot availability
- Booking conflicts resolution

Payment Testing:

- Valid/invalid card numbers
- Payment processing simulation
- Transaction recording
- Refund processing

UI/UX Testing:

- Responsive design
- Form validation
- Error handling
- User experience flows

** 📊 Database Schema Highlights**

Core Tables:

- `users` - User accounts with roles
- `trainers` - Trainer profiles
- `bookings` - Session bookings
- `booking_requests` - Pending requests
- `time_slots` - Availability slots
- `payments` - Payment transactions
- `messages` - In-app messaging
- `programs` - Workout programs

** 🚀 Application URLs**

Frontend: `http://localhost:3000`

Backend API: `http://localhost:8000`

API Documentation: `http://localhost:8000/docs`

** ⚠️ Important Notes**

1. **Payment System:** Completely simulated - no real transactions
2. **Test Data:** Pre-populated for demonstration
3. **Educational Project:** Designed for university coursework
4. **JWT Tokens:** 30-minute expiration for security

5. **Database:** MySQL with proper relationships and constraints