LoopVerse - SkillSphere Platform

Project Overview

SkillSphere is a comprehensive real-time microlearning and mentorship platform that connects learners with mentors for focused 1-on-1 learning sessions. Built as a full-stack EdTech SaaS solution, it demonstrates modern web development practices with secure authentication, role-based access control, real-time communication, and Al-powered recommendations.

Architecture & Technology Stack

Backend (Node.js/Express)

- Framework: Express.js with RESTful API design
- Database: MongoDB with Mongoose ODM
- Authentication: JWT-based authentication with bcrypt password hashing
- Real-time Communication: Socket.IO for WebSocket connections
- Al Integration: Google Generative AI (Gemini) for personalized recommendations
- File Handling: Multer for profile picture uploads
- Validation: Express-validator for input validation

Frontend (React/TypeScript)

- Framework: React 19 with TypeScript
- **Styling**: Tailwind CSS 4.x for modern, responsive design
- State Management: Zustand for lightweight state management
- Routing: React Router DOM for client-side navigation
- Forms: React Hook Form with Zod validation
- Real-time: Socket.IO client for live features
- Build Tool: Vite for fast development and optimized builds

Key Features Implemented

M Authentication & Authorization

- Multi-role authentication system (Learner, Mentor, Admin)
- JWT token-based session management
- Role-based route protection
- Secure password hashing with bcrypt

User Management

- Learners: Profile creation with interests, goals, and learning preferences
- Mentors: Comprehensive profiles with skills, expertise, hourly rates, and availability
- Admins: Platform oversight with mentor approval and content moderation capabilities

Session Management

- Real-time session booking system
- Calendar integration with timezone support
- Multiple meeting types (video, audio, chat, in-person)
- Session status tracking (pending, confirmed, completed, cancelled)
- Automated pricing calculation based on mentor rates

AI-Powered Features

- Personalized mentor recommendations using Gemini AI
- Learning path generation based on user goals and skills
- Topic suggestions tailored to learner interests
- Intelligent mentor matching algorithms

○ Real-time Communication

- WebSocket-based messaging system
- Live session chat functionality
- Typing indicators and real-time notifications
- Session room management with user presence tracking

Analytics & Progress Tracking

- User progress analytics
- Session completion tracking
- · Learning milestone monitoring
- Platform usage statistics for admins

& Development Approach

1. Modular Architecture

The project follows a clean, modular architecture with clear separation of concerns:

- Backend: Organized into routes, controllers, models, and middleware
- Frontend: Component-based architecture with reusable UI components
- Database: Well-structured schemas with proper relationships and indexing

2. Security-First Design

- JWT authentication with secure token handling
- Input validation and sanitization
- CORS configuration for cross-origin requests
- Environment variable management for sensitive data
- Password hashing and secure session management

3. Real-time Capabilities

- WebSocket integration for instant communication
- Live session management with real-time updates

- Notification system for booking confirmations and updates
- Typing indicators and presence detection

4. Al Integration Strategy

- Google Generative AI integration for intelligent recommendations
- Context-aware prompting for personalized suggestions
- Mentor matching algorithms based on skills and preferences
- Learning path generation with milestone tracking

5. User Experience Focus

- Responsive design with Tailwind CSS
- Intuitive navigation with role-based dashboards
- Interactive booking system with calendar integration
- Real-time feedback and notification systems

Project Structure

```
loopverse/
 — backend/
     LoopVerse/
          ├── Models/  # Database schemas  
├── routes/  # API endpoints  
├── controllers/  # Business logic  
├── middleware/  # Authentication & validation  
├── utils/  # Helper functions & WebSocket
                                   # Helper functions & WebSocket
                                # Main application entry
             – app.js
   - frontend/
     └─ src/
          ├─ components/ # Reusable UI components
            — pages/
                                  # Route components
             services/ # API & WebSocket servicescontexts/ # React contexts
             - hooks/
                                  # Custom React hooks
             - types/
                                  # TypeScript definitions
          └─ utils/
                               # Helper utilities
    README.md
```

Getting Started

Prerequisites

- Node.js (v18 or higher)
- MongoDB (local or cloud instance)
- Google Al API key for Al features

Backend Setup

```
cd backend/LoopVerse
npm install
cp .env.example .env # Configure environment variables
npm run seed # Populate database with sample data
npm run dev # Start development server
```

Frontend Setup

```
cd frontend
npm install
npm run dev  # Start development server
```

Environment Configuration

Create .env file in backend with:

```
MONGODB_URI=mongodb://localhost:27017/loopverse
JWT_SECRET=your_jwt_secret
GOOGLE_AI_API_KEY=your_google_ai_key
PORT=4001
```

% Key Implementation Highlights

Database Design

- User Collections: Separate collections for Learners, Mentors, and Admins
- Session Management: Comprehensive session tracking with timezone support
- Analytics: Event-based analytics for progress tracking
- Relationships: Proper referencing between users, sessions, and feedback

API Design

- RESTful endpoints with consistent response formats
- Comprehensive error handling and validation
- Role-based access control middleware
- Pagination and filtering for large datasets

Real-time Features

- · Socket.IO integration with authentication middleware
- Session-based room management
- Live messaging with typing indicators
- Real-time notifications and updates

Al Integration

- Context-aware recommendation engine
- Personalized learning path generation
- · Intelligent mentor matching
- Natural language processing for user preferences

UI/UX Design Philosophy

The platform emphasizes clean, intuitive design with:

- Role-specific dashboards tailored to user needs
- Interactive booking system with visual calendar interface
- Real-time feedback through notifications and live updates
- Responsive design ensuring optimal experience across devices
- Accessibility considerations with proper semantic markup

Testing & Quality Assurance

- Integration testing for API endpoints
- Component testing for React components
- End-to-end testing for critical user flows
- Performance monitoring and optimization
- Security testing for authentication and authorization

Scalability Considerations

- Database Indexing: Optimized queries with proper indexing
- Caching Strategy: Redis integration ready for session caching
- Load Balancing: Stateless design for horizontal scaling
- CDN Integration: Asset optimization for global delivery
- Microservices Ready: Modular architecture for service separation

Future Enhancements

- Video calling integration with WebRTC
- · Payment processing for session bookings
- Advanced analytics dashboard
- Mobile application development
- Multi-language support
- Advanced AI features with machine learning

Built with for the LoopVerse Web Development Challenge

This platform demonstrates modern full-stack development practices, real-time communication, Al integration, and scalable architecture design suitable for production EdTech applications.