Technical Specifications for the "Social IQ" Project

Table of Contents

1. **Project Overview**

Project Name: **Social IQ**

Purpose: To enhance individuals' emotional and social intelligence using AI by providing interactive scenarios that analyze and improve users' emotional understanding, expression, and social interactions through text, voice, and facial expressions.

2. **Objectives**

- Emotional Recognition and Management: Enable users to recognize and manage their own emotions and those of others.
- Personalized Feedback: Provide detailed, AI-driven feedback on users' interactions, including verbal and non-verbal cues.
- Skill Development: Enhance users' practical skills in various social situations, such as negotiations, conflict resolution, and interviews.
- Confidence Building: Boost users' self-confidence through positive reinforcement and constructive feedback.

3. **Functional Requirements**

3.1. **User Management**

- Account Creation and Authentication:
  - Users can create accounts using email or social login.
  - Secure authentication mechanisms (OAuth 2.0).

- Profile Management:
  - Users can edit personal information.
  - Set preferences and goals.

3.2. **Interactive Scenarios**

- Scenario Selection:
  - Users can select from a library of predefined scenarios (e.g., job interviews, social gatherings).
- Custom Scenario Creation:
  - Users can customize scenarios based on specific needs.

3.3. **Real-Time Interaction**

- Text-Based Interaction:
  - Users can engage in text conversations with AI-powered virtual characters.
- Voice Interaction:
  - Users can communicate verbally using speech recognition with real-time analysis of their voice.
- Facial Expression Recognition
  - Real-time analysis of users' facial expressions via webcam.

3.4. **AI Analysis and Feedback**

- ChatGPT Integration:
  - Generate natural language responses and analyze user inputs.

- Voice Tone Analysis:
  - Assess emotional cues from voice (pitch, tone, pace).

- Facial Expression Analysis
  - Detect emotions based on facial movements (smile, frown, eye contact).

- Feedback Delivery
  - Provide immediate and post-session feedback highlighting strengths and areas for improvement.

3.5. **Progress Tracking**

- Performance Metrics:
  - Track key performance indicators (KPIs) such as confidence level, emotional recognition accuracy, and communication effectiveness.

- Historical Data:
  - Access past sessions and feedback reports.

- Goal Setting:
  - Allow users to set and modify learning objectives.

4. **Non-Functional Requirements**

- Scalability:
  - Support concurrent users without performance degradation.
- Security:
  - Ensure data encryption in transit and at rest.
- Privacy:
  - Comply with GDPR and other data protection regulations.
- Usability:
  - Intuitive user interface accessible to non-technical users.

5. **System Architecture**

5.1. **Overview**

The system follows a client-server architecture with modular components to handle various functionalities:

- Client Application:
  - Web and mobile interfaces(later stage)

- Backend Server:
  - API services, user management, and integration logic.

- AI Services:
  - ChatGPT for language processing and sentiment analysis.
  - Voice analysis (Speech prosody) engine (https://dev.hume.ai/docs/expression-measurement/websocket)
  - Facial expression recognition module (https://dev.hume.ai/docs/expression-measurement/websocket)

- Database:
  - User data, session logs, and analytics.

5.2. **Architectural Layers**

1. Presentation Layer
   - User interfaces (web, mobile apps).
2. Application Layer
   - Handles business logic and API endpoints.
3. Integration Layer
   - Manages communication between application and AI services.
4. Data Layer
   - Databases and data storage solutions.

6. **Modules and Components**

6.1. **User Interface Module**

- Web Application

  - Built with React.js for responsiveness.

- Mobile Application

  - React native

6.2. **Authentication Module**

- OAuth 2.0 Implementation

  - Secure login and token management.

6.3. **Scenario Management Module**

- Scenario Library:

  - Database of predefined scenarios.

- Customization Tools

  - Interface for creating custom scenarios.

6.4. **AI Integration Module**

- ChatGPT API Wrapper

  - Manages requests and responses to/from ChatGPT.

- Voice Processing Engine

  - Utilizes Voice Engine API

- Facial Recognition Engine

  - Utilizes Face Expressions Engine API.

6.5. **Feedback Module**

- Analysis Engine:
  - Aggregates data from AI modules.
- Reporting Tools:
  - Generates feedback reports and visualizations.

6.6. **Data Management Module**

- Database Systems:
  - Relational database (Firebase) for structured data.
  - NoSQL database (MongoDB) for unstructured data like logs.
- Data Analytics:
  - Tools for tracking user progress and system performance.

7. **Technologies and Tools**

- Frontend
  - HTML5, CSS3, JavaScript (ES6+), React.js
  - React Native for mobile apps
- Backend
  - Node.js with Express.js framework
- Databases

  - Firebase, MongoDB

- AI Services

  - OpenAI's ChatGPT API

  - Voice Analysis Engine API

  - Facial Expressions Engine API

- Security:

  - SSL/TLS encryption

  - JWT for token-based authentication

8. **Data Flow and Integration**

8.1. **User Interaction Flow**

1. Login and Authentication

   - User logs in via the client application.

2. Scenario Selection

   - User selects or creates a scenario.

3. Real-Time Interaction

   - Text, voice, and video data are captured.

4. Data Processing

   - Text input is sent to ChatGPT.

   - Voice input is processed by the voice analysis engine.

   - Video input is analyzed by the facial recognition engine.

5. AI Response Generation

- ChatGPT generates responses based on context.

6. Feedback Generation

   - Data from all engines are aggregated to produce feedback handled by ChatGPT

7. Feedback Delivery

   - Immediate cues during the session.

   - Detailed report post-session.

8.2. **Integration Points**

- ChatGPT API

  - For natural language understanding and response generation.

- Voice Analysis

  - Integration with speech APIs for tone analysis.

- Facial Expressions

  - APIs or libraries that process video streams to detect expressions.

- Data Aggregation

  - Middleware to collect and synchronize data from different sources.

9. **Security and Privacy Considerations**

9.1. **Data Security**

- Encryption

  - Data encryption at rest using AES-256.

  - Data encryption in transit using SSL/TLS.

- Access Control
  - Role-based access control (RBAC) for administrative functions.
- Regular Audits
  - Security audits and penetration testing.

9.2. **Privacy Compliance**

- Anonymization
  - Personal identifiers are anonymized when storing interaction data.
- Consent Management
  - Users must provide explicit consent for data collection and processing.
- Compliance
  - GDPR, CCPA, and other regional data protection laws.

10. **User Interface Specifications**

10.1. **General UI Principles**

- Intuitive Design
  - Simple navigation and clear instructions.
- Responsive Layout
  - Adaptable to various screen sizes and devices.

10.2. **Key Screens and Features**

- Dashboard
  - Overview of progress, recent activities, and suggested scenarios.
- Scenario Selection Screen
  - Browse and search functionalities.
- Interaction Screen
  - Real-time chat interface.
  - Video window for facial expression capture.
  - Indicators for voice recording status.
- Feedback Screen
  - Displays performance metrics, graphs, and recommendations.
- Profile Screen
  - User information and settings.

11. **Deployment Plan**

11.1. **Environment Setup**

- Development
  - Local and cloud-based development environments.
- Testing
  - Separate staging environment mirroring production.
- Production

- Cloud hosting with providers like AWS, Azure, or Google Cloud.

11.2. **Deployment Steps**

1. Continuous Integration
   - Automated builds and tests upon code commits.
2. Continuous Deployment
   - Automated deployment to staging after successful tests.
3. Monitoring
   - Use of monitoring tools to ensure uptime and performance.

12. **Testing and Quality Assurance**

12.1. **Testing Types**

- Unit Testing
  - For individual components and modules.
- Integration Testing
  - Ensuring modules interact correctly.
- System Testing
  - Complete end-to-end testing of the application.
- User Acceptance Testing (UAT)
  - Feedback from a group of end-users.

12.2. **Testing Tools**

- Automated Testing
  - Jest, Mocha for JavaScript testing.
- Performance Testing
  - Tools like JMeter.
- Security Testing
  - Vulnerability scanners like OWASP ZAP.

Conclusion

The "Social IQ" project aims to provide a comprehensive platform for enhancing emotional and social intelligence through advanced AI technologies. By integrating ChatGPT for language interactions and combining it with voice and facial expression analysis, the platform offers users a holistic tool to improve their social skills in a variety of contexts. The technical specifications outlined above provide a blueprint for building a robust, secure, and user-friendly application that meets the project's objectives.