# RFP（Request for Proposal）：ログイン機能開発

## 提案依頼書概要

**プロジェクト名**: Miraim - 婚活男性向け内面スタイリングアプリ  
**対象機能**: 会話型ログイン機能  
**発注者**: Miraimプロジェクトチーム  
**提案締切**: 2025/7/14  
**プロジェクト開始予定**:2025/7/14  
**納期**: 2025/7/30

## 1. プロジェクト背景

### 1.1 サービス概要

Miraimは婚活に不安を感じる男性向けの内面スタイリングサポートアプリです。AI会話練習、カウンセリング、診断機能、スタイリング提案を通じて、ユーザーの婚活成功をサポートします。

### 1.2 外注の目的

従来の一般的なログインフォームではなく、**会話形式でユーザー情報を取得するログイン機能**を開発し、ユーザーにとってより親しみやすく、サービスの特色を活かした認証体験を提供したいと考えています。

## 2. 開発対象機能

### 2.1 基本機能：会話型ログイン

#### 2.1.1 新規登録フロー（会話形式）

ユーザーとの自然な会話を通じて、以下の情報を段階的に取得：

**必須取得項目**： - 氏名（ニックネーム可） - メールアドレス - パスワード - 年齢 - 職業 – 婚活状況- 出身地- 居住場所- 趣味

**任意取得項目**： - 婚活の悩み・目標

**会話例**：

Bot: こんにちは！Miraimへようこそ。まずはお名前を教えていただけますか？  
  
User: 田中太郎です  
  
Bot: 田中さん、よろしくお願いします！続いて、メールアドレスを教えてください。ログイン時に使用します。  
  
User: tanaka@example.com  
  
Bot: ありがとうございます。次に、安全なパスワードを設定しましょう。8文字以上で英数字を含むものをお願いします。  
...

#### 2.1.2 ログインフロー（会話形式）

既存ユーザー向けの親しみやすいログイン体験：

**フロー**： 1. ユーザー識別（メールアドレスまたはニックネーム） 2. パスワード認証 3. ウェルカムメッセージとパーソナライズされた案内

**会話例**：

Bot: おかえりなさい！メールアドレスまたはお名前を教えてください。  
  
User: tanaka@example.com  
  
Bot: 田中さんですね！パスワードを入力してください。  
  
User: [パスワード入力]  
  
Bot: お疲れ様です、田中さん！今日も婚活スキルアップ、頑張りましょう！

### 2.2 技術要件

#### 2.2.1 フロントエンド

* **フレームワーク**: Next.js (React)
* **言語**: TypeScript
* **スタイリング**: Tailwind CSS
* **UIライブラリ**: 既存のデザインシステムに準拠
* **レスポンシブ対応**: モバイルファースト

#### 2.2.2 バックエンド連携

* **API仕様**: RESTful API
* **認証方式**: JWT（JSON Web Token）
* **データ形式**: JSON
* **既存APIとの統合**: FastAPIで構築された既存エンドポイントとの連携

#### 2.2.3 セキュリティ要件

* **パスワード暗号化**: bcrypt等の安全なハッシュ化
* **入力検証**: クライアント・サーバー両側での検証
* **XSS/CSRF対策**: 適切なセキュリティヘッダー設定
* **レート制限**: ブルートフォース攻撃対策

### 2.3 UX/UI要件

#### 2.3.1 会話インターフェース

* **チャット風UI**: メッセージバブル形式
* **タイピングエフェクト**: Botの応答時の演出
* **入力候補表示**: 選択肢がある場合の候補提示
* **エラーハンドリング**: 分かりやすいエラーメッセージ
* **進捗表示**: 登録完了までの進捗可視化

#### 2.3.2 アクセシビリティ

* **WCAG 2.1 AA準拠**: 基本的なアクセシビリティ対応
* **キーボード操作**: フルキーボード操作対応
* **スクリーンリーダー対応**: 適切なARIA属性設定

### 2.4 デザイン・UI 方針

#### 2.4.1 基本方針

**目指すイメージ**：LINEのような親しみやすいチャットUI

**技術スタック**： - Next.js + TypeScript - 既存プロジェクトで使用中のライブラリ（MUI、Tailwind CSS）と統一

#### 2.4.2 ブランドカラー

* **メインカラー**: オレンジ系グラデーション（#FF8551 → #FFA46D）
* **アクセントカラー**: 成功=グリーン、注意=イエロー、エラー=レッド
* **会話UI**: ユーザー・Bot・システムメッセージで背景色を使い分け

#### 2.4.3 参考サイト・UX

1. **FiNC** - 1問ずつ進む親しみやすいフロー
2. **LINE** - 馴染みのあるチャットデザイン

#### 2.4.4 会話トーン

* **親しみやすく、サポーティブ**な口調
* 堅苦しくない丁寧語
* エラーも温かみのある表現で

**例**： - 良い → “こんにちは！一緒に始めましょう” - 悪い → “入力してください”

## 3. 追加オプション機能

必須ではないですが、オプションもできたら希望します。

### 3.1 音声入力対応

#### 3.1.1 機能概要

テキスト入力に加えて、音声入力でも情報取得が可能な機能。

#### 3.1.2 技術要件

* **音声認識**: Web Speech API またはクラウド音声認識サービス
* **対応言語**: 日本語
* **音声品質**: ノイズキャンセリング対応（可能な範囲で）
* **フォールバック**: 音声認識失敗時のテキスト入力切り替え

#### 3.1.3 UX要件

* **音声入力UI**: マイクボタンとビジュアルフィードバック
* **音声認識状態表示**: 聞き取り中・処理中の状態表示
* **音声確認機能**: 認識結果の確認・修正機能
* **プライバシー配慮**: 音声データの取り扱いについての説明

## 4. 既存システム連携

### 4.1 技術スタック

* **フロントエンド**: Next.js 14+ (App Router)
* **バックエンド**: FastAPI (Python)
* **データベース**: MySQL 8.0
* **認証**: JWT + OAuth2
* **インフラ**: Azure Container Apps

### 4.2 API仕様（参考）

#### 4.2.1 ユーザー登録

POST /api/auth/register  
Content-Type: application/json  
  
{  
 "name": "田中太郎",  
 "email": "tanaka@example.com",  
 "password": "securePassword123",  
 "age": 28,  
 "occupation": "エンジニア",  
 "konkatsu\_status": "beginner"  
}

#### 4.2.2 ログイン

POST /api/auth/login  
Content-Type: application/json  
  
{  
 "email": "tanaka@example.com",  
 "password": "securePassword123"  
}

### 4.3 データベーススキーマ（参考）

CREATE TABLE users (  
 id INT PRIMARY KEY AUTO\_INCREMENT,  
 name VARCHAR(100) NOT NULL,  
 email VARCHAR(255) UNIQUE NOT NULL,  
 password\_hash VARCHAR(255) NOT NULL,  
 age INT,  
 occupation VARCHAR(100),  
 konkatsu\_status ENUM('beginner', 'experienced', 'returning'),  
 created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
);

## 5. 成果物

### 5.1 納品物

1. **ソースコード**
   * フロントエンド（React/TypeScript）
   * バックエンド連携部分
   * テストコード
2. **ドキュメント**
   * 実装仕様書
   * API仕様書
   * セットアップガイド
   * ユーザーマニュアル
3. **デザイン成果物**
   * UI/UXワイヤーフレーム
   * デザインモックアップ
   * インタラクションフロー図

### 5.2 品質基準

* **コードカバレッジ**: 80%以上
* **パフォーマンス**: 初回ロード3秒以内
* **ブラウザ対応**: Chrome, Firefox, Safari, Edge（最新2バージョン）
* **モバイル対応**: iOS Safari, Android Chrome

## 6. プロジェクト管理

### 6.1 開発手法

* **進捗報告**: 適宜
* **コミュニケーション**: Slack

### 6.2 品質管理

* **コードレビュー**: Pull Request必須
* **ユーザビリティテスト**: 開発途中でのユーザーテスト実施

## 7. 予算・スケジュール

### 7.1 想定予算

* **基本機能（会話型ログイン）**: [200万(2,000p)]
* **追加オプション（音声入力）**: [100万(1,000p)]
* **総予算上限**: [300万(3,000p)]

## 8. 提案要件

### 8.1 提案書に含めるべき内容

1. **技術アプローチ**
   * 会話フロー設計方針
   * 技術選定理由
   * セキュリティ対策
2. **実装計画**
   * 詳細スケジュール
   * マイルストーン設定
   * リスク分析と対策
3. **チーム体制**
   * 担当者
   * プロジェクト管理体制

## 9. 契約条件

### 9.1 著作権・知的財産権

* 成果物の著作権は発注者に帰属
* オープンソースライブラリの利用は事前承認制

### 9.2 機密保持

* プロジェクト情報の機密保持契約締結
* 開発環境でのデータ取り扱い規定

### 9.3 保証・サポート

* 運用サポート（別途契約）

## 10. 問い合わせ先

**プロジェクト担当者**: 小林徹  
**メールアドレス**: toru09may@gmail.com  
**電話番号**: 090-1828-9947  
**提案締切**:

## 11. 添付資料

1. 既存システム技術仕様書
2. ブランドカラー・基本デザイン（参考）
3. 会話フロー例

**注意事項**: - 本RFPの内容は機密情報として取り扱ってください - 提案内容について不明な点がございましたら、お気軽にお問い合わせください - 技術的な制約がある場合は、代替案もご提案ください