**Report:**

This report details the implementation of a Marriage Matchmaking API backend service using FastAPI and SQL Alchemy. The system is designed to help users find potential matches based on their profile information, with features for creating, reading, updating, and deleting user profiles, along with a specialized matching algorithm.

**Core Functionality Implementation** : The foundation rests on a complete set of CRUD (Create, Read, Update, Delete) operations for user management. Each endpoint has been designed to handle user records with proper validation and error handling. Email validation employs regex pattern matching to ensure proper formatting, while a unique constraint prevents duplicate registrations, maintaining data integrity across the platform.

**Data Transformation and Storage** : The system implements a hybrid approach where interests are stored efficiently as strings in the database but are automatically transformed into more list structures during API interactions. This transformation is managed through Pydantic validators, providing a seamless interface between the storage layer and API consumers while optimizing database performance.

**Matchmaking Algorithm Design** : The matchmaking evaluates potential matches based on same city requirement, shared interests, gender compatibility, and age appropriateness (within a ±10 year range). These criteria are combined using SQLAlchemy's logical operators (or\_ and and\_), creating efficient database queries that scale well with user growth.

**Validation and Error Management**: The system maintains data integrity through validation at multiple levels. Input validation occurs during both user creation and update operations.

**Assumptions:**

Email id is set as unique.