Hired1st Technical Exercise Documentation

This project is created using Angular 16 as frontend, .Net core 6.0 as backend and MongoDB as database.

**Frontend**

1. Directory Structure:
   1. Arranged files and folders logically.
   2. Common directories include "src" for source code, "service" for service files, “assets” for static files and "components" for UI components.
2. Separation of Concerns: Component-based architecture ensures a clear separation between UI components, services, and business logic.
3. Dependency Injection: Used in registering of services in components.
4. Form Validation: Leverage Angular Reactive Forms for robust form validation and handling.

**Backend**

1. Project Structure:

* Controllers: For API endpoints.
* Services: Business logic.
* Repositories: Data access.
* Models: Define data models.
* Middleware: Custom middleware for error handling

1. User Authentication: Implemented user authentication for sign-in and sign-up functionalities using a secure approach like JWT (JSON Web Tokens). Use ASP.NET Core Identity authentication mechanism.
2. MongoDB Integration: Integrated MongoDB for data storage.
3. Error Handling and Validation: Implemented proper error handling and validation for API requests.
4. Middleware: Used middleware for exception handling
5. Service-Repository Pattern:
   1. Service: Implement methods for business logic.
   2. Repository: Implement methods for data access.
6. Dependency Injection: Registered services and repositories with Dependency Injection.
7. CORS (Cross-Origin Resource Sharing): Configure CORS settings in your backend to define which domains can access your API.
8. All global like connection string, gmail credentials are set up in app-settings file, that are then set to variables in startup.cs