Human Resource Dictionary – Phase 1 Development Plan

# Objective

To design and implement a foundational Human Resource Dictionary web application using .NET for backend and Angular for frontend. This application will allow:

- User authentication (login, logout, change password)

- User management (create, edit, delete, view)

- Resource management (create, edit, view resources with technology stacks)

# Overall Technology Stack

- Frontend: Angular (latest version)

- Backend: .NET 6 Web API

- Database: SQL Server

- Authentication: JWT-based token authentication

- ORM: Entity Framework Core

# Detailed Feature Development Plan

Below, each feature includes: What the feature does, How it will be implemented, Estimated time.

## Authentication Module

### Login

Purpose: Allow users to sign in using email and password.

Frontend Tasks (1.5 hours):  
- Build Login UI (form with validations)  
- Call backend API to authenticate  
- Store JWT token in local storage  
- Redirect user on success

Backend Tasks (2 hours):  
- Create login endpoint  
- Validate credentials  
- Generate JWT token  
- Return token to frontend

### Logout

Purpose: Clear user session and return to login.

Frontend Tasks (0.5 hour):  
- Implement logout button  
- Clear stored token  
- Redirect to login page

Backend Tasks (0.5 hour):  
- No backend work (stateless JWT)

### Change Password

Purpose: Allow logged-in users to update their password.

Frontend Tasks (2 hours):  
- Create Change Password form with validation  
- Call API to update password

Backend Tasks (3 hours):  
- Implement password update endpoint  
- Validate old password  
- Hash and store new password

## Home Screen

Purpose: Show quick statistics.

Frontend Tasks (2 hours):  
- Build dashboard UI to display counts  
- Fetch counts via API

Backend Tasks (2 hours):  
- Implement endpoint to return:  
 - Total Users Count  
 - Total Resources Count

## Application User Management

### Create User

Purpose: Admin can create new users.

Frontend Tasks (3 hours):  
- Build Create User form  
- Validate fields  
- Submit to API

Backend Tasks (3 hours):  
- Endpoint to create user  
- Validate input  
- Hash password  
- Store in DB

### Edit/Delete User

Purpose: Admin can modify or remove existing users.

Frontend Tasks (2 hours):  
- Implement Edit form  
- Add Delete confirmation  
- Integrate with API

Backend Tasks (3 hours):  
- Endpoint to update user  
- Endpoint to delete user  
- Handle validation and consistency

### View All Users

Purpose: List all users in a table with options.

Frontend Tasks (4 hours):  
- Build data table with pagination and Filter/search  
- Display user data  
- Actions (edit, delete)

Backend Tasks (1 hour):  
- Endpoint to fetch user list (additional search api)

## Resource Management

### Create Resource

Purpose: Create a resource with multiple technology stacks.

Frontend Tasks (3 hours):  
- Build Create Resource form  
- Multi-select for tech stacks  
- Validations

Backend Tasks (5 hours):  
- Endpoint to create resource  
- Save all fields  
- Handle many-to-many relationship for technology stacks

### Update Resource

Purpose: Edit resource information.

Frontend Tasks (1 hour):  
- Edit Resource form  
- Populate existing data

Backend Tasks (1.5 hours):  
- Endpoint to update resource  
- Validate changes

### View All Resources

Purpose: Display all resources in a list.

Frontend Tasks (2 hours):  
- Data table UI  
- Filter/search

Backend Tasks (1 hour):  
- Endpoint to retrieve resources list(additional search api)

# Summary of Time Estimates

|  |  |  |
| --- | --- | --- |
| Feature | Frontend (Hours) | Backend (Hours) |
| Authentication | 4 | 5.5 |
| Home Screen | 2 | 2 |
| User Management | 9 | 7 |
| Resource Management | 6 | 7.5 |
| Total | 21 | 22 |

# Development Approach

1. Setup Phase:  
 - Initialize Angular project  
 - Initialize .NET Web API  
 - Configure JWT Authentication  
 - Set up database schema  
2. Authentication Module:  
 - Complete login/logout/change password  
 - Test end-to-end flow  
3. User Management:  
 - Create user CRUD APIs  
 - Build user management frontend  
4. Resource Management:  
 - Implement resource CRUD APIs  
 - Build resource management frontend  
5. Home Screen Dashboard:  
 - Develop summary endpoints  
 - Display in dashboard  
6. Testing & QA:  
 - End-to-end testing of workflows

# Deliverables

- Full source code (Angular + .NET)  
- Database schema