**Project Proposal – Employee Management System**

1. **Objective**

Purpose of this project is to build a web application that allows managers to manage employee records, leave schedules as well as performance. The application is to use modern frameworks but preferably one inclined to Typescript/Javascript for the **front-end**, C# for the **back-end** and a database like MSSQL, Postgres to store data. The web app is to have different functionalities as well such as extracting reports based on employee performance, number of hours worked, as well as what the employee worked on ,etc.

1. **Project Work Plan**
2. **Time frame – 6 days:**

**Day 1 – 2 -**  Research

**Day 3 -** Database Development

**Day 4 -** Backend Development

**Day 5 -** Frontend Development

**Day 6 -** Evaluation

1. **Stage Analysis:**

**Research**

To carry out research I shall use different sources such as the internet, as well as using common knowledge gained during my school work. On the internet websites such Github, w3schools as well as geeks for geeks are essential as they apply a common knowledge base to all Programming Activities as well as Software Development. Consultation to expertise is another way to carry out research but with limited time, a few shall be consulted.

**Database Development**

For database development I shall be using Microsoft SQL Server. This platform allows us to carry out sql query languages easily. Other reason for using SSMS is as per the requirement of the question. SSMS also allows easy connectivity to backend using whichever framework or language. A provided sqlquery file shall be provided to be executed in SSMS.

**Back-end Development**

For Back-end development l shall use c# , as required by question. For the framework l shall be using **.net web core api**. I shall connect database with backend via a connection string then afterwards create models using existing tables in database. After doing so l shall further-on create controllers to allow usability of tables to perform the **CRUD** functionalities , I further go on and run code and test API endpoints on **swagger.ui.** l also test the CRUD operations on each of the tables. I preserve created endpoints for use in front-en connection.

**Front-end Development**

For Front-end development I shall use javascript. I shall use React.js as a framework to implement my front-end. I have react.js, as it it a popular framework and as per analysis is easier to grasp. The front-end will first consist of a login page with a Signup link to signup page. After logging in a dashboard will appear, depending on role of user two buttons will appear , for a normal employee they will see a Daily report and Leave Request button. For the manager they will see View leave requests and View reports buttons. As the buttons are self explanatory the daily report button will ink to a page for one to generate a report for a day’s work. The Leave request will direct to a page for one to apply for leave. The manager shall then perform the CRUD operations on the data. Using the API endpoints created in the Backend , the front-end shall send HTTP requests and communicat with back-end as well as database.