**CHAPTER-1**

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Earlier, this kind of process was being conducted on paper, where the chances of fair evaluation was little bleak due to chances of human error. Also it requires more accuracy in timing and paper work. When tasks are too large in terms of complexity or cost then it is difficult to manage papers related to evidence and inspection. Time to time reporting to the end user or agent is also very difficult if tasks are handled manually.

This system is built on web platform that will manage personal documents and business-related financial documents on a single platform without having any manual paper work. It manages Supply chain management from source to destination with proper flow of goods by evidence and inspection control. This centralized platform will use different machine learning techniques to verify proper quantity of goods. This makes it easy to follow product and process chain of supply by simply reviewing reports which answer who, what, where and when for every element you see.

* 1. **OBJECTIVE**

This system will comprise of a website where the candidates can report their work time to time and provide task related evidence.

Inspection and evidence system specializing in management and controlling the proper flow of documents and task from point of origin to point of distribution

Main purpose of this system is to eliminate the paper work to handle all documents. Management of proper service flow with less time by automating manual processes with the help of different technologies. The proposed system will feature different modules that ensures the proper flow of documents and related task to all those who are concerned with a job. A system that would be able to manage users with authentication, their documents, verification of team members, forms for task, flow of entities from source to destination, assets that are required to perform different task etc.

* 1. **SCOPE**

**What it can do?**

This system will allow users to use all the modules provided in it after successful login followed by registration.

After proper verification of details such as username/email and password, functionality would be accessible to the user.

As per the requirement user can create forms for job and assets, view and update profile, upload and download data sheets, create Jobs with required fields.

User would be able to view job with different format, add team members who are responsible to get the task done, generate report, add attachments (images, files, audio, video) etc.

Some features with it’s scope are given below:

First user will have to do sign up to use the system. After proper verification of user’s details such as email verification, contact number verification, one will be able to use the proposed system.

Different Module wise functionality is given below:

**Dashboard:**

It is provided to the users to view summary like structure of activities.

This sub module will display following details.

Number of jobs created by user.

Total added location

Number of reports generated by user

Space/Storage occupy by the documents stored by the user.

Information about jobs such as name, author, status (Pending, in progress, Closed).

**Profile:**

User information will be display in this sub module.

It includes following details.

User Name

Email

Phone

Profile photo

Delete profile, Edit profile functionality

Change password functionality.

**User Documents:**

If any documents are uploaded by user, then the details such as job name in which documents are uploaded, description, total size, last modified by, and updated time and date would be display in this sub module.

**Job:**

Whenever this type of system will be introduced to user, it is necessary to have one most important feature that is able to manage or prepare single Job/Task.

In Job tab, front end part will display List of Jobs with below necessary details.

Different working functionalities are given below.

**Create job, Edit job, Delete job, Share job, View job**

After implementation of the solution, user will be able to create Job with job details (Title, Description, location, overview, and conclusion).

In overview and conclusion tab, user will be able to use existing forms or can create form as per the requirements.

System will provide dynamic form building functionality to satisfy their needs. Different attachment options would be there to attach images or documents in different format.

Create job functionality will also allow to add task that has to be done, agents who will manage task, assets require to complete job etc.

User will be able to share job with others who are responsible to actively participate in job.

**Report Generation.**

After job is created successfully, user can generate report as per his format as there would be different formatting options available mentioned below.

Basic options:

Print quality (Grayscale, Colorful)

Image options (All images, Select Images)

Layout options (Full, Compact)

Thumbnail size (Normal, 2x)

Additional details (Full size media, print attachments, Download links)

Advanced options:

Choose sections (Check In, Check List, Images, Audio, Video, Job, Attachment, Email details, Conclusion history)

Grid line options (Horizontal, Vertical)

Table header options (Left-to-right, Top-to-bottom)

Mail will be sent to user by which he can easily download report.

**Data Import:**

This sub module will provide functionality to import or export data sheets (CSV, XLS, XLSX) having important data require to add or download for the task that has to be done.

**Forms:**

Different type of forms would be created here.

Forms created in this sub module could be use into other Job Module also.

It will allow creating three basic types of forms given below.

General Form

Overview Form

Conclusion Form

**Assets:**

List of assets require to complete the task would be added from here.

It will allow creating forms to add assets by dynamic form builder functionality.

These assets could be import to the Job module while creating job.

* 1. **TOOLS AND TECHNOLOGY USED**

Table 1.1 Technologies Used

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Type** | **Description** |
| 1 | Operating System | Windows |
| 2 | Technology used in front end | Angular 6 |
| 3 | Application Development Environment | .net core framework |
| 4 | Tools for Development | Visual studio 17/ Visual Studio Code |
| 5 | Technology used in back end | PostgreSQL |

**1.4.1 Front-End Technology:**

Angular 6 have been used as front-end technology.

**Why Angular 6?**

Angular is a platform that makes it easy to build applications with the web. It combines declarative templates, dependency injection, inbuilt backend integration, end to end tooling, and integrated best practices to solve development challenges

Angular apps load quickly with the new Component Router, which delivers automatic code-splitting so users only load code required to render the view they request.

Command line tools: start building fast, add components and tests, then instantly deploy.

Quickly create UI views with simple and powerful template syntax.

With Karma for unit tests, you can know if you've broken things every time you save. And Protractor makes your scenario tests run faster and in a stable manner.

**1.4.2 Back-End Technology:**

.NET core have been used as back-end technology.

**Why .NET core?**

ASP.NET Core is open source and available on GitHub.

ASP.NET Core is supported on Windows, Mac and Linux OSX.

In ASP.NET Core we can use NPM for managing client-side dependencies.

The .NET Core Framework is optimized for the cloud. It is a cross platform framework for building ASP.NET Core applications. It is supported in Windows, Mac and Linux and open source on GitHub.

**1.4.3 Database:**

PostgreSQL have been used to store and retrieve relevant data

**Why .NET core?**

PostgreSQL is a powerful, open source relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.

PostgreSQL requires very minimum maintained efforts because of its stability.  Therefore, if you develop applications based on PostgreSQL, the total cost of ownership is low in comparison with other database management systems.

PostgreSQL was then also designed to be portable so that it could run on various platforms such as Mac OS X, Solaris, and Windows.