**Software Project Documentation**

**Title Page**

* **Employee Details With Documents**
* **Tools Used**
* **Author: Arif**

|  |  |
| --- | --- |
| **Table of Contents** | **Page No.** |
| 1.Introduction | 3 |
| 2.Project Overview | 3 |
| 3. Tools Used | 4 |
| 4. Architecture | 4 |
| 5. Design | 5 |
| 6. Implementation | 5 |

1. Introduction

This project presents a small web application developed using ASP.NET MVC that streamlines the creation and management of employee records while ensuring the security of sensitive documents. The application allows users to create employee profiles, upload important documents in an encrypted format, and view them in a decrypted form. This dual approach not only protects sensitive information from unauthorized access but also ensures that users can easily access and manage their data as needed.

Key features of the application include the ability to edit and delete employee records, providing flexibility in managing personnel information. Additionally, users can view the total number of employees, facilitating better oversight and resource management within the organization.

2. Project Overview

2.1 Employee Creation: Users can easily add new employees by entering their details into a structured form. This process is intuitive and designed to minimize data entry errors.

2.2 Document Management: The application allows users to upload important documents related to each employee in an encrypted format. This ensures that sensitive information is securely stored and accessible only to authorized users.

2.3 Document Upload Type: The application allows users to upload important documents related to each employee, including Emirates ID, passport, and visa. Documents must be in accepted formats: DOC, PDF, and XLS.

2.4 Decryption and Viewing: Users have the ability to view documents in a decrypted format, providing them with the information they need while maintaining data security.

2.5 Edit and Delete Functions: Users can update employee records as necessary, allowing for accurate and current information management. Additionally, the option to delete employee profiles ensures that outdated or incorrect data can be easily removed.

2.6 Employee Count: The application features a dashboard that displays the total number of employees, enabling administrators to monitor workforce size at a glance

3. Tools Used

This project was developed using a variety of technologies to ensure a robust and efficient web application.

* Framework: ASP.NET MVC was utilized to create the web application, providing a structured and scalable architecture.
* Programming Language: C# was used for backend development
* Database: Microsoft SQL Server 2019(MSSQL) was employed for database management, allowing for secure and efficient data storage and retrieval.
* Development Environment: Visual Studio 2022 served as the integrated development environment (IDE), offering powerful tools for coding, debugging, and testing, which streamlined the development process.
* Frontend Technologies:
* JavaScript and jQuery were used to enhance interactivity and improve user experience on the client side.
* HTML and CSS were utilized for structuring and styling the user interface, ensuring a clean and responsive design.
* Bootstrap was integrated for rapid UI design, allowing for a mobile-first approach and ensuring compatibility across various devices.

4. Architecture

The architecture of the Employee Management System is based on the Model-View-Controller (MVC) design pattern, which separates the application into three interconnected components:

1. Model:
   * The Model represents the application's data and business logic. In this project, it consists of employee entities and their associated data, such as personal details and uploaded documents. The Model interacts with the Microsoft SQL Server database to perform CRUD (Create, Read, Update, and Delete) operations.
2. View:
   * The View is responsible for rendering the user interface and displaying data to the user. It uses HTML, CSS, and Bootstrap for layout and design, ensuring a responsive and user-friendly experience. The View is dynamically generated based on the data provided by the Model, allowing users to interact with employee records seamlessly.
3. Controller:
   * The Controller acts as an intermediary between the Model and the View. It processes user input, retrieves data from the Model, and selects the appropriate View for rendering. The Controller handles requests for creating, editing, and deleting employee records, as well as managing document uploads and decryption.

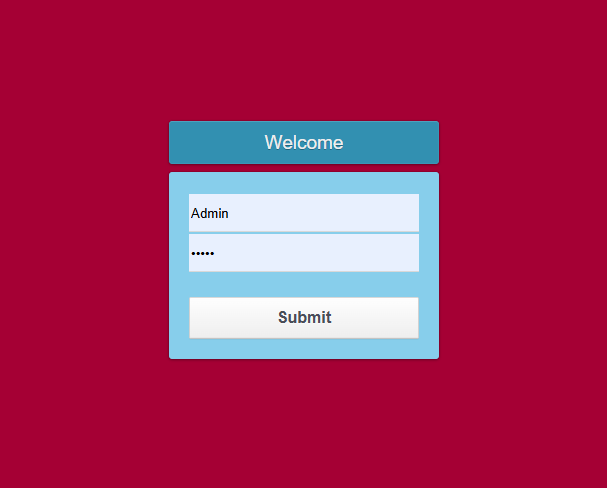
5.Design

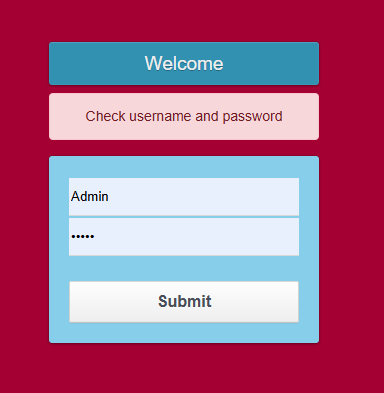
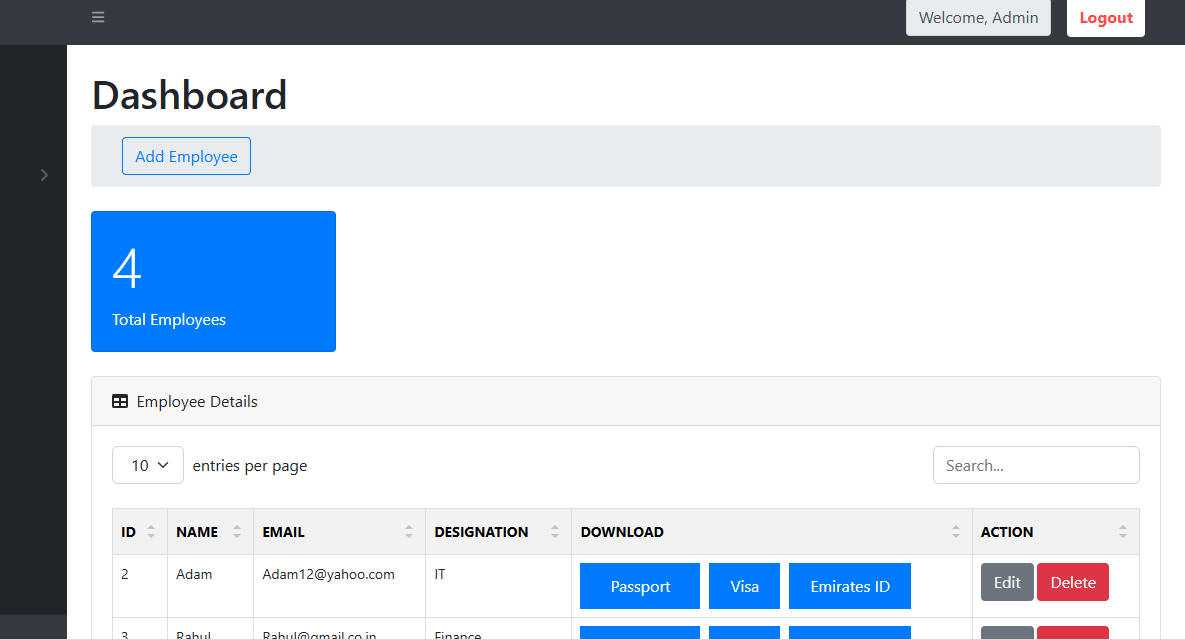
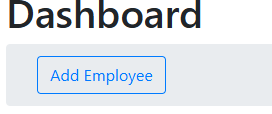
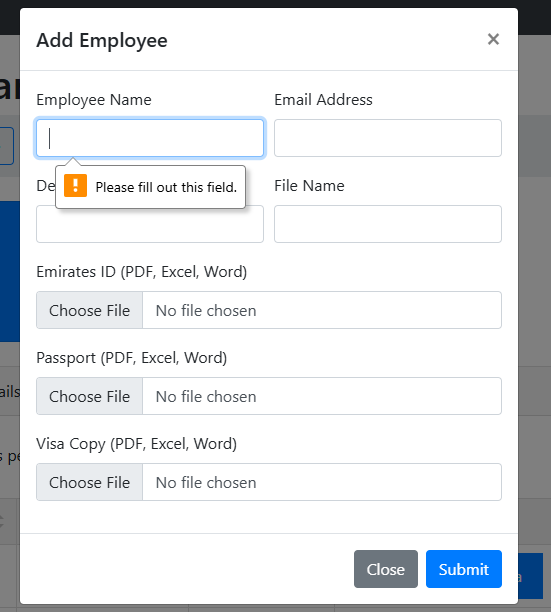
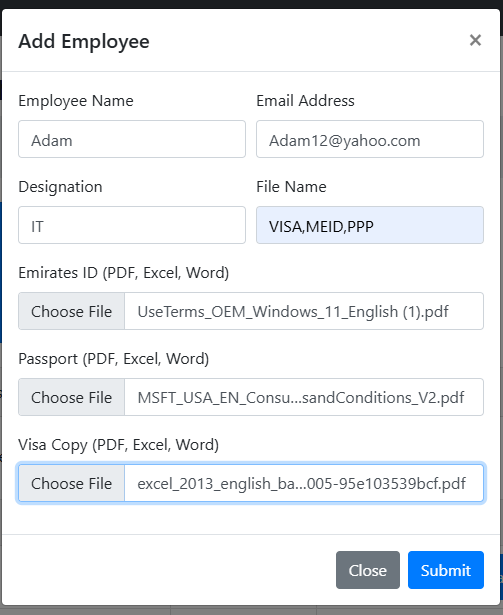
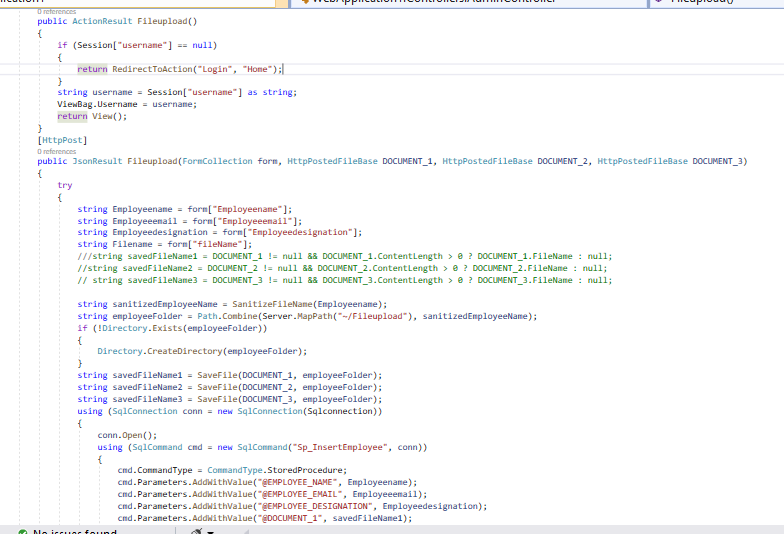
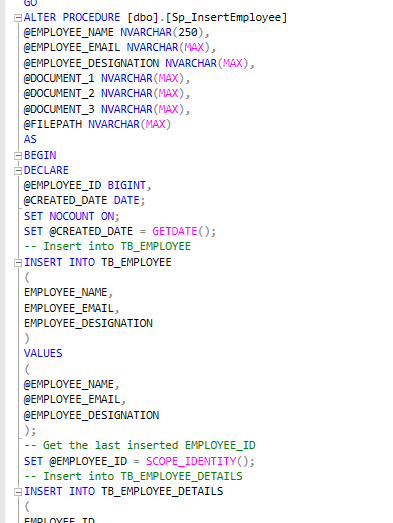
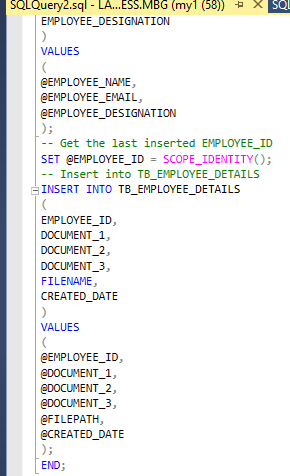
The design of the application emphasizes usability, security, and scalability. Key design aspects include:

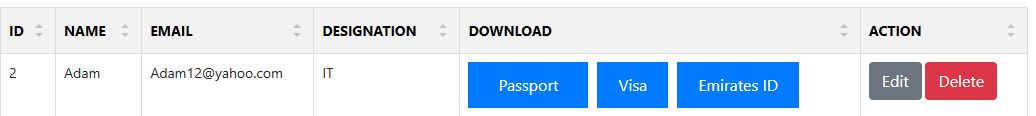
1. User Interface:
   * The UI is designed with simplicity in mind, using Bootstrap to ensure a clean and modern appearance. The layout is responsive, providing a consistent experience across various devices and screen sizes.
2. Database Design:
   * The database schema is designed to effectively store employee information and associated documents. Relationships between entities are carefully defined to maintain data integrity and facilitate efficient queries.

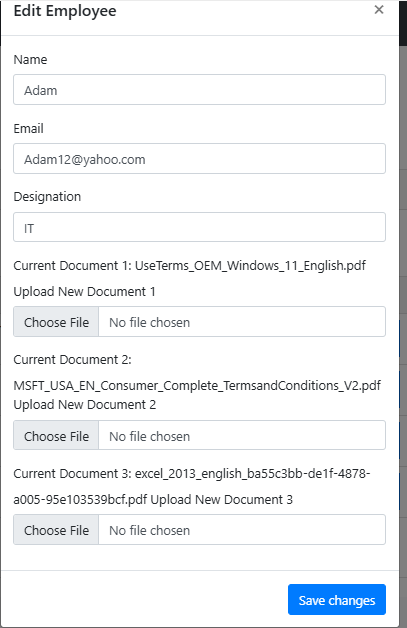
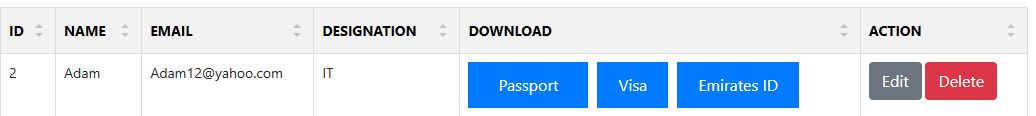
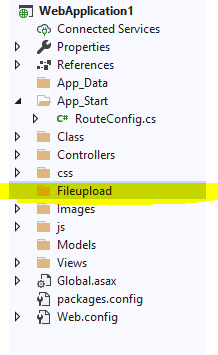
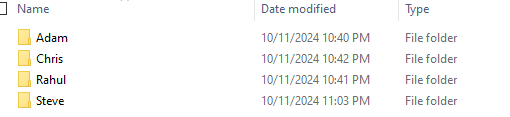
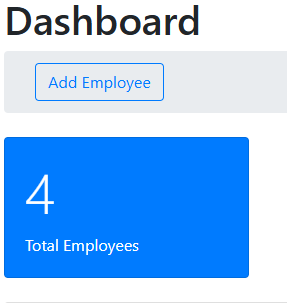
6. Implementation

* Login Page



* In login Page we need to enter the user name and password. If it’s incorrect it will show a alter message “Check user name and password”.
* 6
* If it is correct. It will redirect to the home page
* 
* In Home Page Will show Login name , And logout on top
* 
* After Welcome It will show the login name, When click log it will log out from here.
* We are using session, so without log in accessing page not possible.
* BackEnd
* 
* I am using form collection here and using stored procedure here for check username and password from here
* Create Employee
* 
* Click on Add employee for add new employee,Then the pop box will appear from here
* Then this popup box will appear here
* 
* All fields are required here
* Fill the details
* 
* Then Click submit
* Then it will save
* 
* This is action for save in controller
* This is the stored procedure
* 
* 
* The employee id I am using scope identity. Scope identity Property is used to create a column in a table that automatically generates numeric values for new rows. This is particularly useful for primary keys.
* It is used to retrieve the last generated identity value from the TB\_EMPLOYEE table after the insertion of a new employee record.
* If nee to edit click the edit button



* Once click the edit button this popup appear
* 
* Make the changes and click save changes
* 
* If need delete record click Delete button
* Then record will be deleted.
* The Upload file will be in encrypted format, when we click the document from download passport,visa, emirates id in encrypted format.
* The file will be saved in folder named FileUpload t will be saved under the employee name folder
* 
* 
* The number of employees will shown appear here as shown in the cards
* 
* The total employee will show in Datable will Details as shown in below
* 