# Report On
# Software development internship

By

Student Full Name: Mst sabrina Mobassira
Student ID: 17301175

An internship report submitted to the Computer Science and Engineering in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of computer Science and engineering
Brac University
September 2021

## Declaration

It is hereby declared that

1. The internship report submitted is my own original work while completing degree at Brac University.
2. The report does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The report does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Sabrina Mobassira

**Mst. Sabrina Mobassira**
17301175

## Approval

The project “Daily activities – Internship on software development” submitted by

1. Mst. Sabrina Mobassira (17301175)

Of summer, 2021 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in computer Science and engineering on September 26th, 2021.

**Examining Committee:**

**Supervisor’s Full Name & Signature:**

Maruf Hasan

**Mr. Maruf Hasan**
Sr. Technology Analyst
Intelligent Image Management Ltd. (IIMI)
House: 200, Road: 2, Baridhara DOHS, Dhaka-1206, Bangladesh.

**Co Supervisor’s Full Name & Signature:**

Iqbal

**Dr. Muhammad Iqbal Hossain**
Assistant Professor
Department of Computer Science and Engineering
BRAC University

## Acknowledgement

Firstly, all praise to the Great Allah for whom my internship has been completed.

Secondly, to my supervisor Mr. Maruf Hasan, Sr. Technology Analyst (IIMI) for his kind support and advice to complete my tasks in internship. He helped us whenever I needed help.

Thirdly, Dr. Muhammad Iqbal Hossain sir guided me on the whole time during the internship period.

And finally, to our parents without their throughout support it may not be possible. With their kind support and prayer, I am now on the verge of my graduation.

# Abstract

IIMI is a multinational IT company, works for data transformation. It maintains comprehensive, well-documented, clearly defined security and confidentiality procedures throughout the organization. The company has eight departments and I worked with the software team under a supervisor. I was appointed in the company as a trainee for six months. This internship was a great experience of professional life as well as a great opportunity to explore my interest in software development.

This also helped me to develop my communication skill also has taught me to work in a team. During this period, I have learned about database designing, programing language C#. Also, gained knowledge about web page or application development with ASP.NET framework, MVC template, C#, CSS, JQuery, bootstrap, stylesheet. I have completed my intern with project implementation and publication. This project has introduced me to SQL database server and visual studio (IDE) for project development. The company wants to follow the daily activities of the employees in an organized easy way with a web page. From this project, people can get information about working activities by date, time, operator name, tasks name in a secured process.

**Table of Contents**

## List of Figures:

# Chapter 1

## 1. Description and Background of the Company:

### 1.1. Company overview:

IIMI IMAGE MANAGEMENT PRIVATE LIMITED is part of IIMI (Intelligent Image Management Inc.). IIMI helps organization of all types and sizes transform their data into information and knowledge, driving insight and action. Currently IIMI is a family of 4200+ full-time dedicated staff members working out of four different countries. Such as USA, Sri Lanka, Bangladesh and India. IIMI started its operation in 1996 and officially incorporated in Nevada USA in 1999.

In terms of working process, IIMI (Intelligent image management) provides data entry, indexing, data conversion, data mining, post scan processing, back office and BPO services to companies of all sorts. IIMI India and other offshore operations are extremely limited and more or less carried out by the USA. There is no share holder of any sort.

### 1.2. Qualification of employees:

The IIMI family consists of around 3500 employees, as well as a pool of skilled on-call operators. Every employee undergoes a thorough background check before being hired and goes through a rigorous training program. IIMI has women and minorities represented at every level of the company and no child labor.

## 1.3. Departments of IIMI:

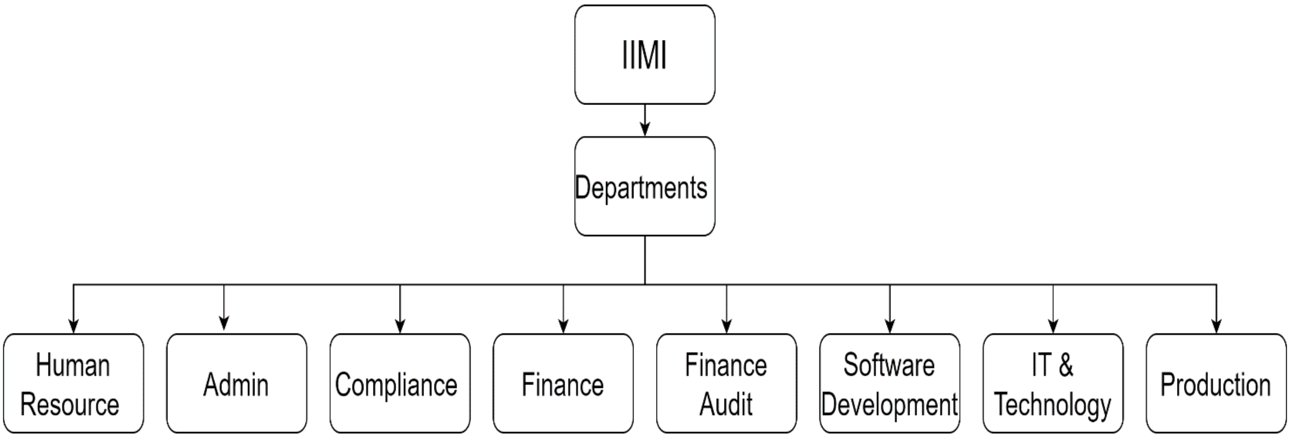


**Fig 1.1: Departments of the company**

## 1.4. Hierarchy chart:

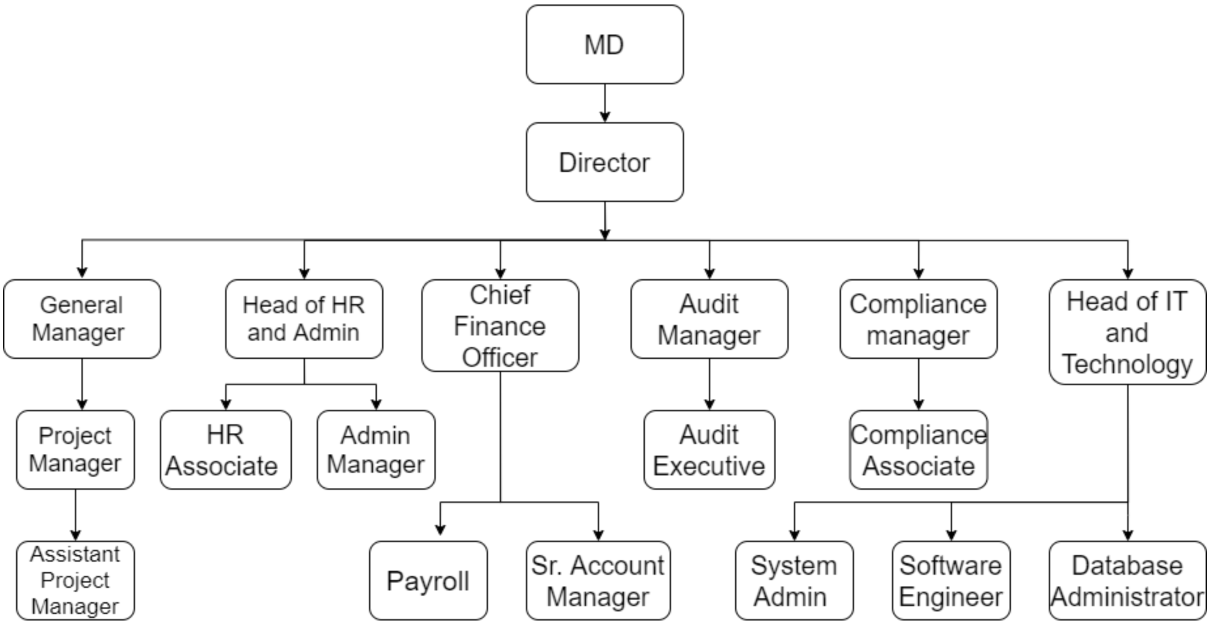


**Fig 1.2: Hierarchy Chart**

# Chapter 2

## 2. The Internship Activities and Responsibilities:

### 2.1. The position and the core responsibility:

I had been appointed in the company as a trainee. I was working with the software team. As a software developer intern, I had to work under the supervision of Sr. technology analyst of the company. The responsibilities that I was given were conducted and advised by the supervisor. Firstly, I learned database server and programming language. My responsibilities were to cooperate and communicate with the software team. I also had the responsibility to create a project with the software team. So, I had discussed about project design, implementation with my team member and understand the process so that I could present it to the supervisor. The project and everything were finalized after the modification given by the supervisor if needed. To enhance the communication skill sometimes I had to interact with the people and response to their needed inquiries and concerns.

### 2.2. The supervisor and the team member:

**Supervisor:**

Mr. Maruf Hasan

Sr. Technology Analyst

Intelligent Image Management Ltd. (IIMI)

House: 200, Road: 2, Baridhara DOHS, Dhaka-1206, Bangladesh.

**Team member:**

Tofael Ahmed

Application Developer

Develop application in different application framework like .NET, PHP, Python

## 2.3. Difference between class study and internship:

The very basic knowledge regarding database and web application was taught in my classroom from which I have gained the theoretical knowledge of database and programming language. Whereas, during my internship I got the opportunity to be acquainted with the creation of database by using SQL server and web application by visual studio. I was able to observe the steps by step functions and programming languages as well as the web development functions. Therefore, my classroom taught me the basic knowledge and internship helped me to apply that knowledge. Moreover, I have gained the practical experience in a different environment in the office.

## 2.4. Impact on career:

Database design is one of the most important essential for system development. In order to pursue a career in the software development sector database design is an essential requirement. In terms of organizations, a well-structured database system is a first and foremost requirement. Moreover, C# is one of the most used and recognized programming languages in terms of web development along with software development and several more. So, the impact this opportunity created in my career is really severe which might result in opening up lots of opportunities in near future.

# Chapter 3

## 3. Learning and skills development:

### 3.1. Server capacity planning and sizing:

Server capacity planning is the method of using knowledge of current server capacity use to improve present and future server performance. Web server capacity planning requires checking monitoring and alerting on server components like processor, CPU, memory, disk, and network.

At the very first days of my internship, I learn about the capacity planning of file server. Sizing of drives, processors, modules, memory, Microsoft SQL server etc. In the mean time I have introduced my documentary on capacity planning for file server.

### 3.2. Database development in SQL server:

Database Design is important because it helps produce database systems. That meet the requirements of the users. Have high performance. Database designing is crucial to high performance database system. Note, the genius of a database is in its design. Data operations using SQL is relatively simple.

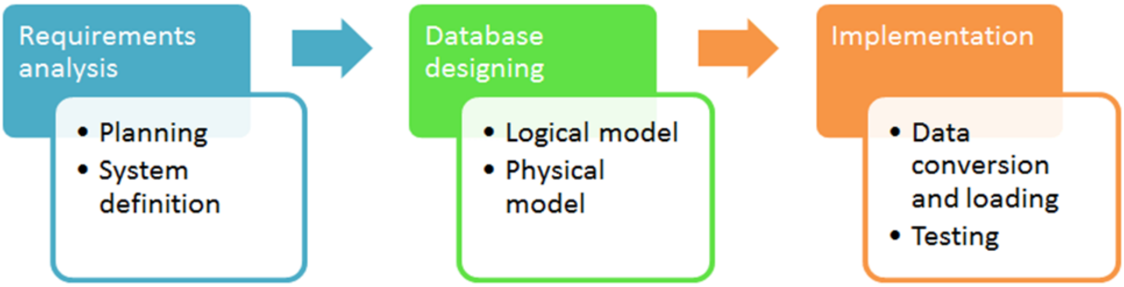


**Fig 3.1: Database development life cycle**

## Database Designing:

Logical model - This stage is concerned with designing a database model based on specification. The entire design is on paper without any physical implementations or relevant DBMS considerations.

Physical model - Taking into account the DBMS and physical implementation variables, this stage implements the logical model of the database.

## Implementation:

Data conversion and loading - this stage is concerned with importing and converting data from the old system into the new database.

## Objects:

### Database Tables:

A database most often contains one or more tables. Each table is identified by a name. We use tables to store and organize data in the database. A table contains columns and rows with data. Every table is broken up into smaller entities called fields. A field is a column in a table that is designed to maintain specific information about every record in the table. A record is also called a row, is each individual entry that exists in a table.

### Entities:

Entities are business objects that your database contains and are used to logically separate the data in the database. Entities List is used to determine the tables, which is necessary to create as part of the physical design phase. You create a separate table in the database for each entity. Entities are characterized by attributes.

**Attributes:**

An attribute is a property of an entity. For example, the employee entity has attributes such as employee id, first name, last name, birthday, social security number, address, country, etc. Some attributes are unique values.

SQL is a standard language for storing, manipulating and retrieving data in databases. SQL can execute queries against a database, can retrieve data from a database. It can insert, update, delete records in a database. SQL can create new databases, tables in a database. Also, create stored procedures and views in a database. As well as can set permissions on tables, procedures, and views.

**Normalization:**

Normalization divides a big table into smaller tables and describes connections between them to improve the visibility in organizing data. A set of methods to break down tables to their constituent parts until each table represents one and only one thing and its columns serve to fully describe only the one thing that the table represents.

In the SQL server we can create all these things by syntax and commands. All the commands that I used it this project is like in bellow:

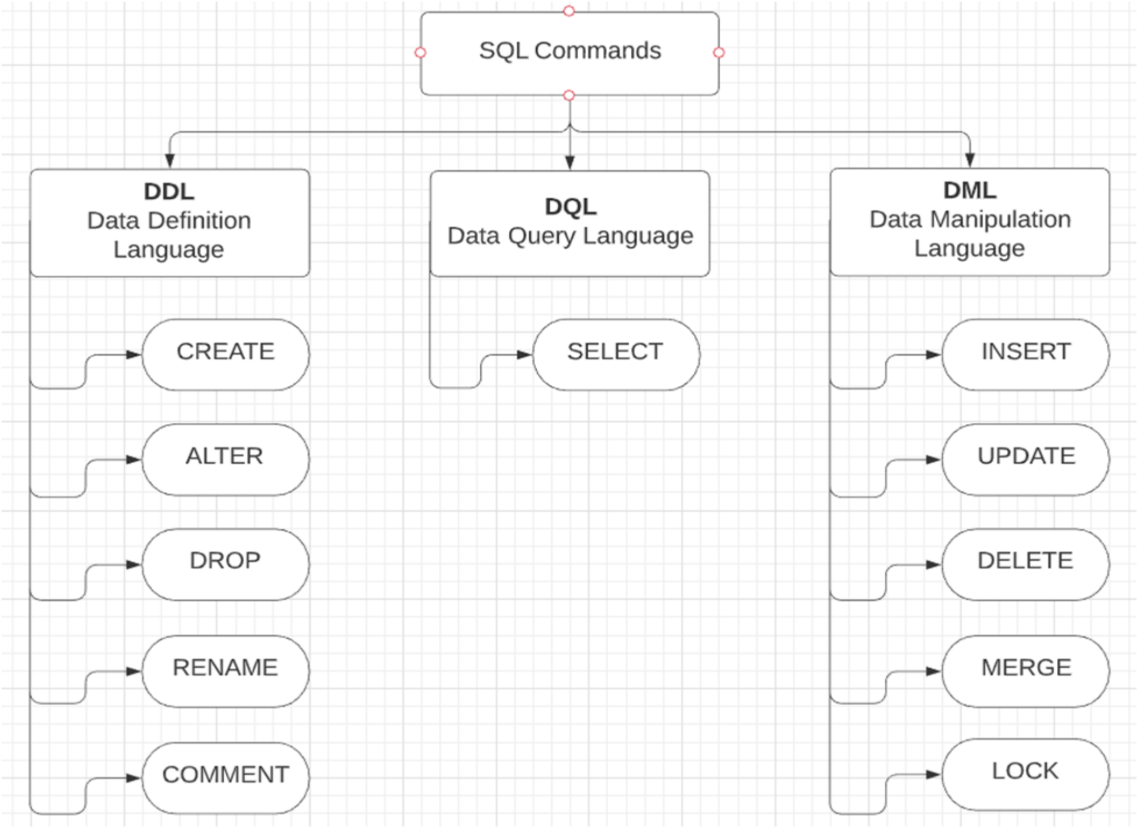


**Fig 3.2: Used command for this project**

Also learn the backup SQL database with SQL server management studio and restore SQL database using SSMS.

### 3.2.1. Graph and chart of the database:

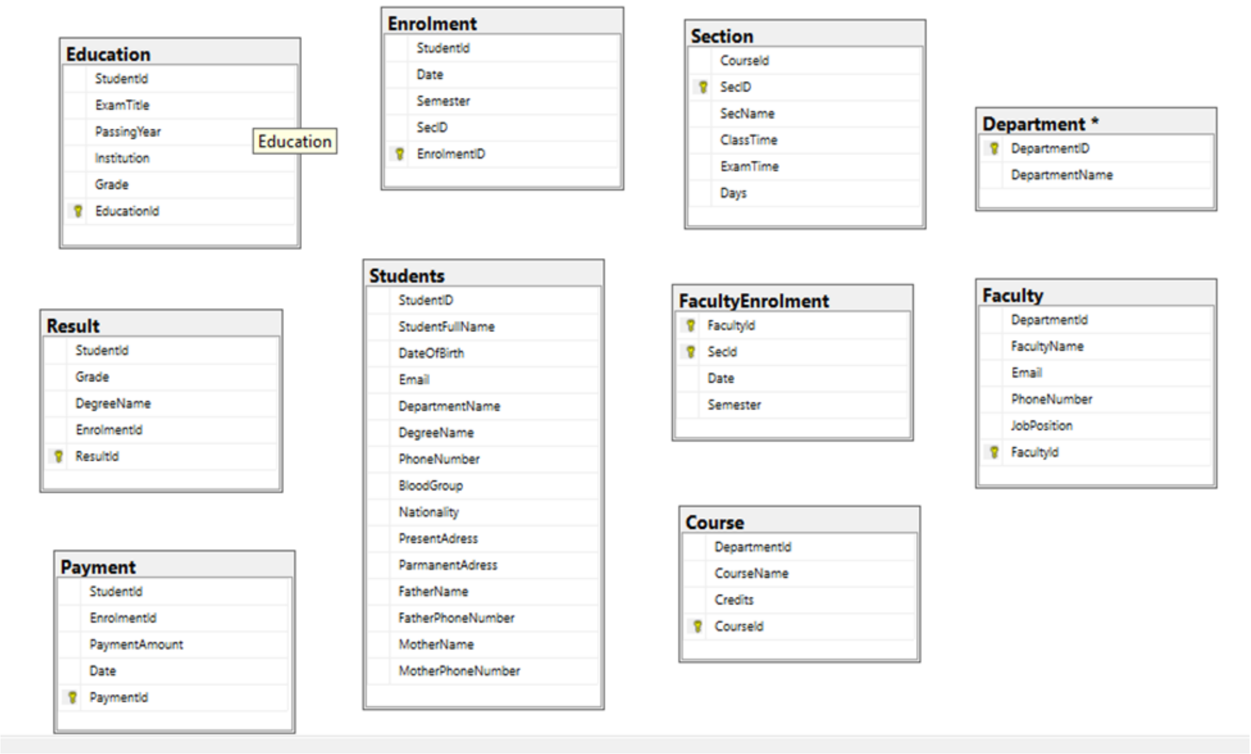


**Fig 3.3: Tables of the database student management**

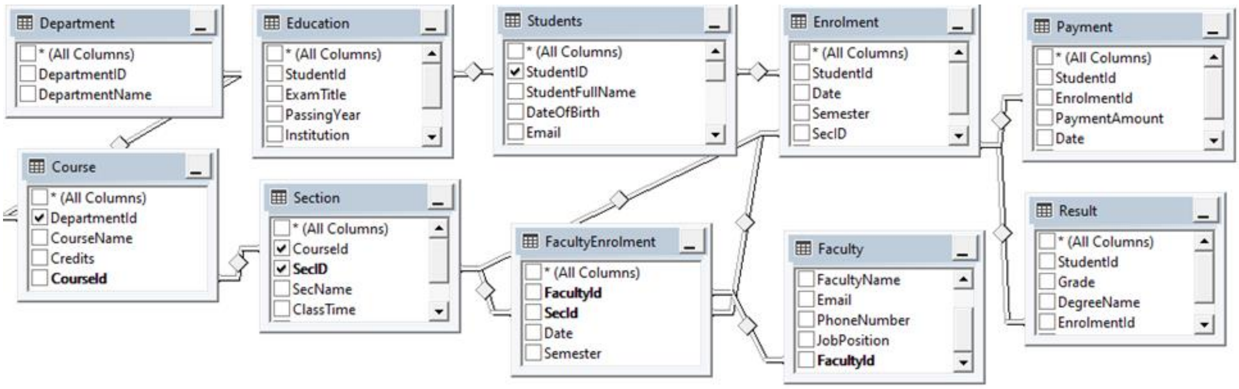


**Fig:3.4: Views of Inner Joins:**

## 3.3. C# language:

### 3.3.1. Introducing C#:

C-Sharp (C#) is an object-oriented programming language. This language is developed by Microsoft specifically to work with the .NET framework. It's a popular programming language in the world which is easy to learn and use. C# is the upgraded version of C and C++ languages. It has roots from the C family and similar to the others languages like C, C++ and Java. So, C# makes it easy for the programmers to switch these languages. Applications that we can write with C# language are web applications, desktop applications, database applications, windows store applications. Moreover, can develop web sites, mobile apps, games etc. These applications can give a clear structure to programs, allows code to be reused and low-cost development. C# use ASP.NET framework so the easiest way to get started with C# in visual studio, IDE.

**Variables:**

To store values of the data types we need to use variables as a name for storage.

Variable naming rules:

Must contains letter, digit, underscore (_).

First character should be letter and can accept underscore.

Can't use special characters like: *, =, + for variables.

Unique name and can't use space ( ).

**Data types:**

Int (Integer): Here stores both positive and negative whole numbers and non-decimal numbers. There is a limitation of size which depends on programming language.

Double: Stores both negative positive floating-point numbers and has size limitation.

Float: Stores real numbers.

String: Stores fixed value like “Sabrina”, “10,000”.

Bool: Stores True or 1, False or 0.

Char: Stores characters

**If else:**

In C# if else is a condition of the coding language. Also, there are some statements. The statements of C# are if (true condition), else (false condition), else if (first false condition), and switch (alternative execution). The conditions came from some mathematical logic for example: less than, greater then, equal to, not equal to, less than or equal to, greater than or equal to.

**For loop:**

For loop is using for multiple execution of a code. We can ensure fixed looping times by using for loops. Otherwise, we can also use while loop. The for loop continue to execute the code unless the condition come out false. In the loop firstly have to take a variable. Then a condition as must need a true false parameter to conform the back-to-back loop execution. After that need to increase the value. For example: (int i=0; i>2; i++).

**Array:**

To store multiple values for one variable, need to use array ([]) which is a data structure and ensure a fixed value. In array data types remain same and no need separate variables. Generally, in C# programmers use single dimensional array from three types of arrays. Another two types are multidimensional and jagged.

**Methods:**

Method is used as block code. It executes if it is called in the program. We use parameter to pass the data. In method there use functions as action. We use methods to use the code many times or for future tasks.

**Oop:**

Oop means object-oriented programming. The benefits of this programming are its very easy to use and simple to execute, takes less time to develop an application. Also, can possible to create programs in future by using the same code. So, it’s a reusable programming process which is faster too. The coding is very easy to maintain, have the process of debugging system and can be modified as well.

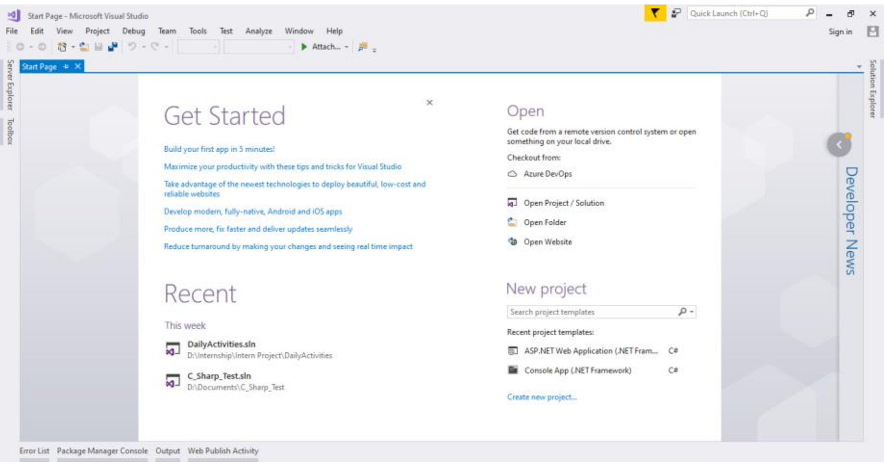
### 3.3.2. IDE

IDE means integrated development environment which is a software building application that combines common developer features, tools into a single graphical user interface needed by software developer. It consists source code editor, local build automation and debug. Examples of IDEs are NetBeans, Eclipse, IntelliJ and Visual Studio. Here I am working with visual studio for learning, practicing and upcoming project.

### 3.3.3. Visual studio

Visual Studio is an integrated development environment or IDE. It is a software application for programmers to write and edit code. This is the program where developers can develop their software applications. The user interface of the visual studio is used to edit, debug and build code for the software developers. It develops apps for android, iOS, Mac, windows, cloud, web and web applications. In this IDE applications can be developed by both forms based and web-based design. The main tasks here are to write the code fast and then debug it. Moreover, very efficient tasks are testing, releasing, executing, extending, customizing and collaborating which can be done perfectly.

After installing and setup, programmers explore to file menu to create new C# applications.



Get Started

Build your first app in 5 minutes!

Maximize your productivity with these tips and tricks for Visual Studio

Take advantage of the newest technologies to deploy beautiful, low-cost and reliable websites

Develop modern, fully-native, Android and iOS apps

Produce more, fix faster and deliver updates seamlessly

Reduce turnaround by making your changes and seeing real time impact

Recent

This week

DailyActivities.sln
D:\Internship\Intern Project\DailyActivities

C_Sharp_Test.sln
D:\Documents\C_Sharp_Test

Open

Get code from a remote version control system or open something on your local drive.

Checkout from:

Azure DevOps

Open Project / Solution

Open Folder

Open Website

New project

Search project templates

Recent project templates:

| | |
|---|---|
| ASP.NET Web Application (.NET Fram... | C# |
| Console App (.NET Framework) | C# |

Create new project...

**Fig 3.5: Visual studio**

### 3.3.4. ASP.NET framework

ASP.NET is an open-source web framework for building websites and web applications on .NET framework which is using HTML, CSS, and JavaScript. This framework is used for three types of operating system like windows, Mac and Linux. There are three types of frameworks in ASP.NET framework those are Web forms, ASP.NET web pages and ASP.NET MVC for web applications. All of them are developed and can be able to make awesome web applications. Generally, programmers build dynamic web applications by ASP.NET MVC framework. ASP.NET MVC is a framework based on three components architecture Model, View and Controller.

### 3.3.5. MVC

Microsoft had introduced ASP.NET MVC in .NET 3.5. MVC architecture supported in ASP.NET. MVC stands for Model, View, and Controller. It means it's divided an application into three components Model, View and controller. These components are used to handle particular development aspects of an application.

The following figure illustrates the interaction between Model, View, and Controller.

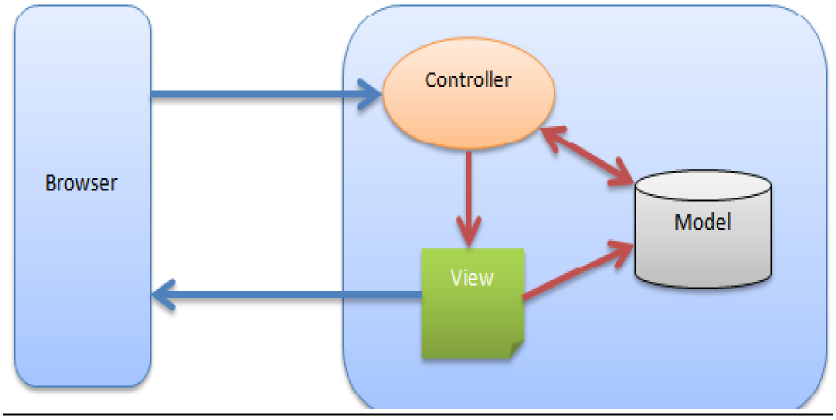


**Fig 3.6: MVC Architecture**

<u>**Model**</u>: The model structures data in a reliable form and prepares it based on the controller's instructions. It manages system data. Model uses CSS, scripts, SQL and stores data from a database.

<u>**View**</u>: Displays data to user easy-to-understand format, based on the user's action. It is the user interface and works as a frontend for images, front view. The view uses HTML, CSS, Razor syntax to communicate with model, controller and styles for coloring. Mainly it represents everything.

**Controller**: The controller manages the user instructions and passes it to the model, view. It Takes in user commands, sends commands to the model for data updates, sends instructions to view to update interface. The controller works in the backend, app data.

# Chapter 4

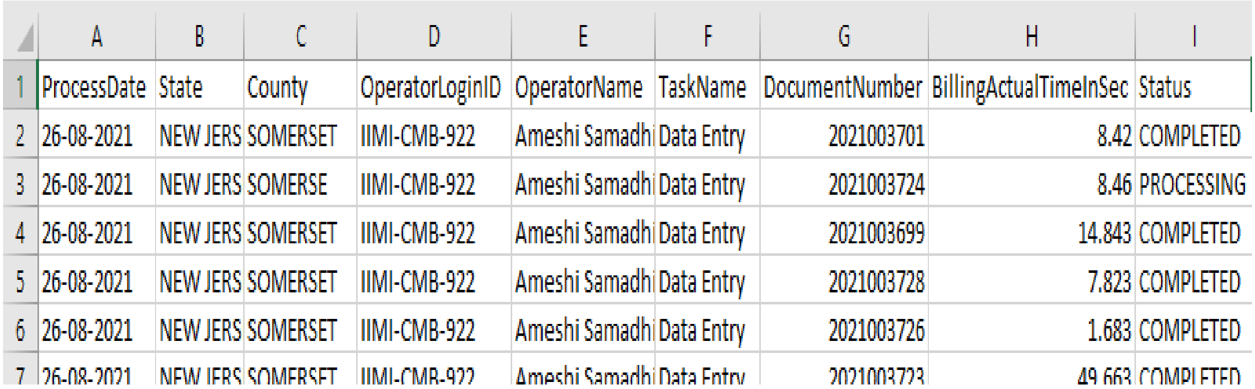
## 4. Project requirements and implementation:

### 4.1. Project requirements:

**Project name**: Daily Activities (Web application)

Step:1

- Read excel/csv file (provide from company’s production).
- Database connection (SQL server).
- Import excel/csv data to the database.
- Those data need to show into grid view.
- In the grid date wise filter.
- In the grid field wise filter.



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeInSec | Status |
| 2 | 26-08-2021 | NEW JERS | SOMERSET | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003701 | 8.42 | COMPLETED |
| 3 | 26-08-2021 | NEW JERS | SOMERSE | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003724 | 8.46 | PROCESSING |
| 4 | 26-08-2021 | NEW JERS | SOMERSET | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003699 | 14.843 | COMPLETED |
| 5 | 26-08-2021 | NEW JERS | SOMERSET | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003728 | 7.823 | COMPLETED |
| 6 | 26-08-2021 | NEW JERS | SOMERSET | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003726 | 1.683 | COMPLETED |
| 7 | 26-08-2021 | NEW JERS | SOMERSET | IIMI-CMB-922 | Ameshi Samadhi | Data Entry | 2021003723 | 49.663 | COMPLETED |

**Fig 4.1: CSV file from production**

Step:2

- Organized the page button and color.
- Total count of document number.
- Sum of the billing actual time in sec.
- Delete selected records or old records.
- User should be created by admin (no self-registration required).
- Add more column in the grid.
- Import more data for future.
- Change the log in button in home page.
- Project publication.

### 4.1.1. Required technologies:

- HTML, CSS, J Query
- C# language
- SQL Database (SQL Server)
- Asp.net
- MVC pattern
- Entity Framework code first using MVC
- Visual studio as IDE

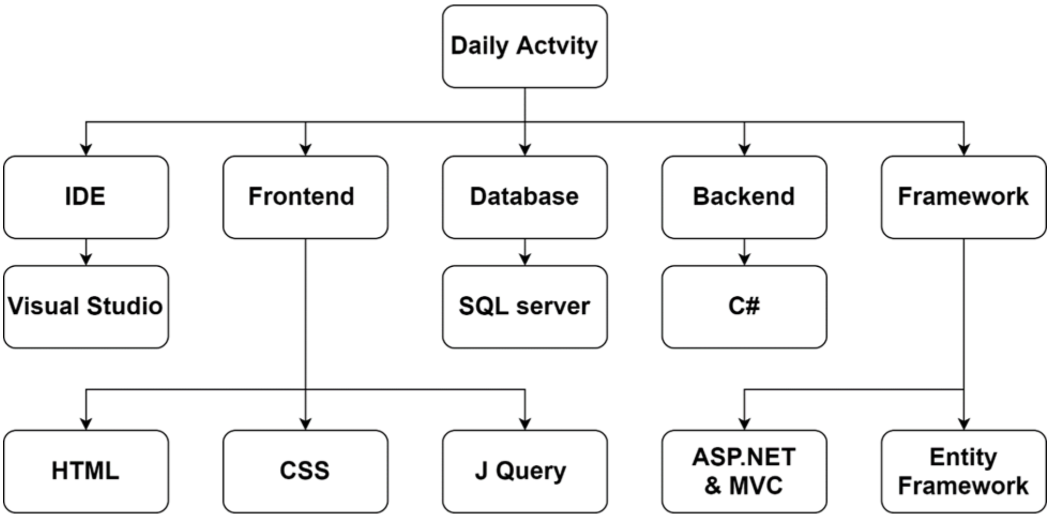


**Fig 4.2: Project overview**

## 4.2. Project implementation

### Project creation:

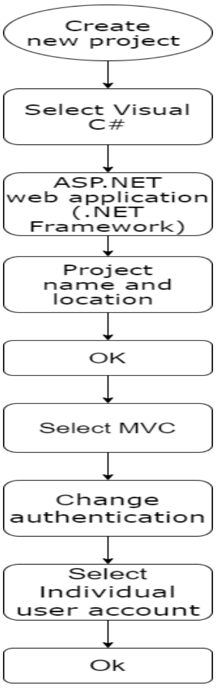
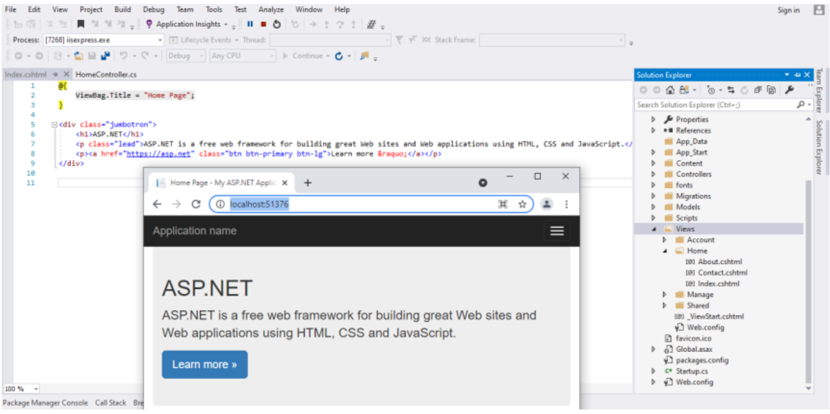

Create new project
Select Visual C#
ASP.NET web application (.NET Framework)
Project name and location
OK
Select MVC
Change authentication
Select Individual user account
Ok

**Fig 4.3: Flowchart of project creation**

This is an ASP.NET web application. The project is developing in visual studio. To create this project, we followed some general steps (Fig 4.3**)**. Firstly, selected visual C# as the programming language and ASP.NET (.NET Framework) framework. Then our project name is Daily Activities which is located in the device. After that, we choose the MVC pattern for our project template. We need to change the user authentication. And for this project, I selected an individual user account. The project executes in a browser. Here it is executed in google chrome as fig 4.4.



```
@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>ASP.NET</h1>
    <p class="lead">ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.</
    <p><a href="https://asp.net" class="btn btn-primary btn-lg">Learn more &raquo;</a></p>
</div>
```

ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS and JavaScript.

**Fig 4.4: Project view & execution in the browser**

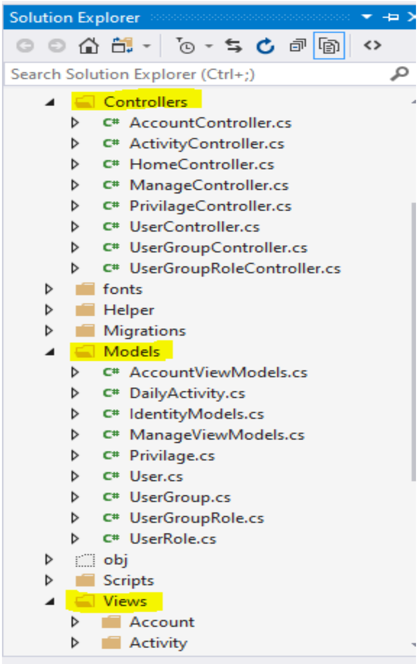


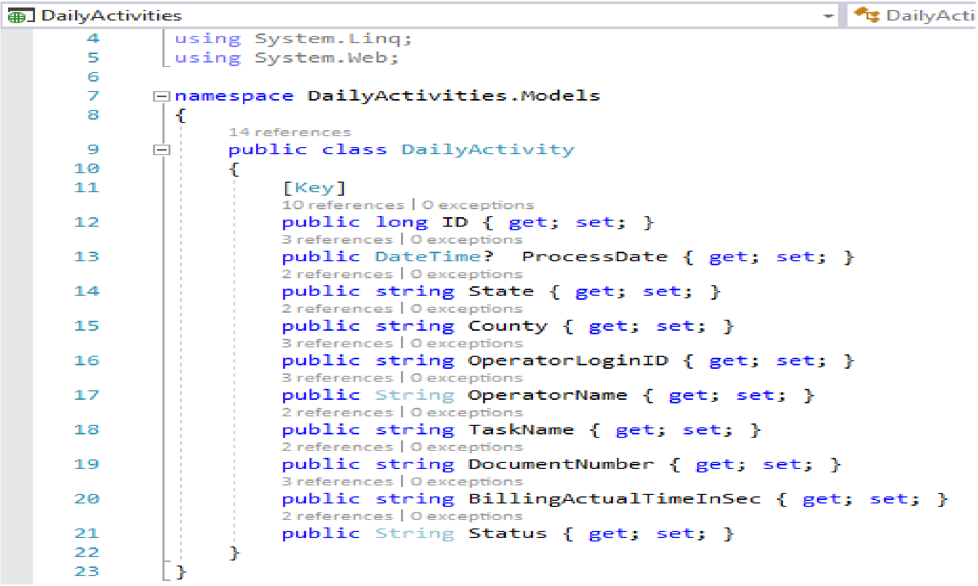
**Fig 4.5: MVC pattern of daily activities project**

In the solution explorer of the project, we can see MVC folders: model, view and controller (Fig 4.5). There are also some default folders. For example, 1. scripts which is containing client-side codes in J Query language. These are maintaining the brows connection like browsing, choosing option and open option in view. It means scripts J Query works for view or view page. Server-side codes are model and controller and client-side codes are in view. 2. Content folder is containing bootstrap which is a stylesheet in CSS language and JavaScript. And these are maintaining web site color, theme, date time etc. We can customize this theme by creating a new class in the content folder. HTML is used for the button, button page, level tag.

The project layout has three parts. Those are header, body and footer. The home page of the application is created in the home index view page. Login and register are created in the account controller. Index, about, contact is created on the home controller page.

## Create class:

Created daily activities class in the model and its properties which is related to the database table as well as with the required excel sheet table. Here we need to set the data types according to the individual properties. For example, Id - long and task name- string (Fig 4.6). Then, registered the class into the controller.



```
using System.Linq;
using System.Web;

namespace DailyActivities.Models
{
    14 references
    public class DailyActivity
    {
        [Key]
        10 references | 0 exceptions
        public long ID { get; set; }
        3 references | 0 exceptions
        public DateTime?  ProcessDate { get; set; }
        2 references | 0 exceptions
        public string State { get; set; }
        2 references | 0 exceptions
        public string County { get; set; }
        3 references | 0 exceptions
        public string OperatorLoginID { get; set; }
        3 references | 0 exceptions
        public String OperatorName { get; set; }
        2 references | 0 exceptions
        public string TaskName { get; set; }
        2 references | 0 exceptions
        public string DocumentNumber { get; set; }
        3 references | 0 exceptions
        public string BillingActualTimeInSec { get; set; }
        2 references | 0 exceptions
        public String Status { get; set; }
    }
}
```

**Fig 4.6: Daily activities class and properties**

## Database connection (SQL server):

At first, we fixed the connection string in web.congig.

- Here, data source = SQL SA address (SQL server); Initial catalog =Dily activities (project name); Integrated security = false; user ID = SA and password = SQL server password.
- Then updated comment in the package manager console:

  Enable-migrations

- Then in configurations:

  AutomaticMigrationsEnabled =true;

  AutomaticMigrationDataLossAllowed = true;

- Fixed the application DB context in identity model for registration.

  public DbSet<DailyActivity> DailyActivities { get; set; }

- Then updated comment in the package manager console:

  Update-Database

So, In the SQL server the database has been updated and connected to the daily activities project as the Fig 4.7.

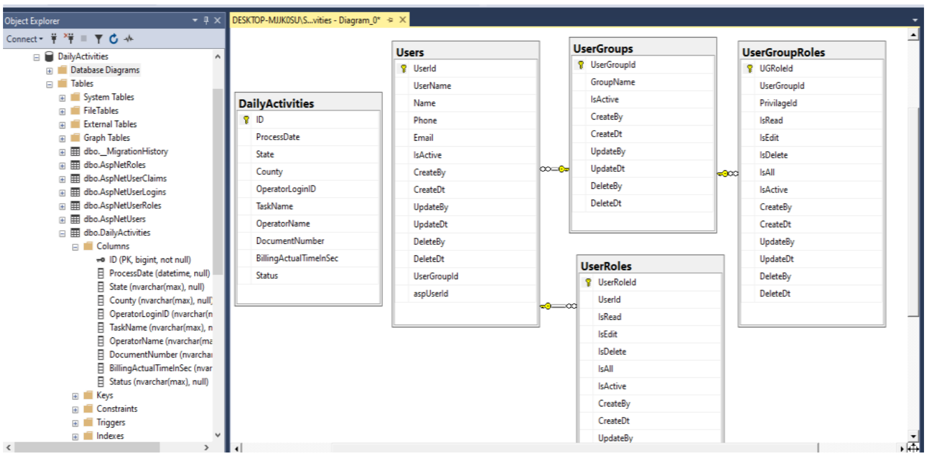
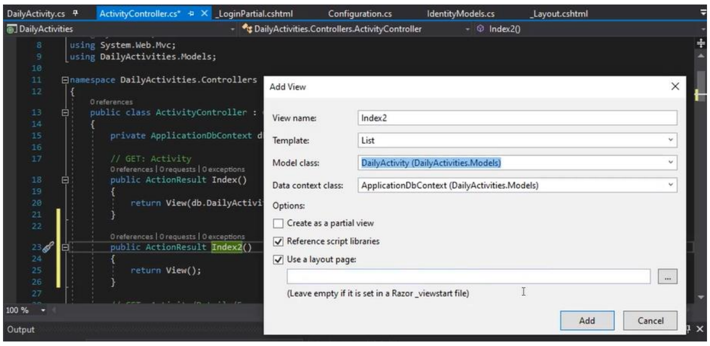


**Fig 4.7: SQL server connection and tables**

## Create a new page on the web page:

To import data, I added a new method in the activity controller. Methods name is index2. Then I created a view folder of index2 like in (Fig 4.8). Here the template is a list.



```
using System.Web.Mvc;
using DailyActivities.Models;

namespace DailyActivities.Controllers
{
    0 references
    public class ActivityController :
    {
        private ApplicationDbContext d

        // GET: Activity
        0 references | 0 requests | 0 exceptions
        public ActionResult Index()
        {
            return View(db.DailyActivi
        }

        0 references | 0 requests | 0 exceptions
        public ActionResult Index2()
        {
            return View();
        }
```

Add View

View name: Index2

Template: List

Model class: DailyActivity (DailyActivities.Models)

Data context class: ApplicationDbContext (DailyActivities.Models)

Options:

☐ Create as a partial view

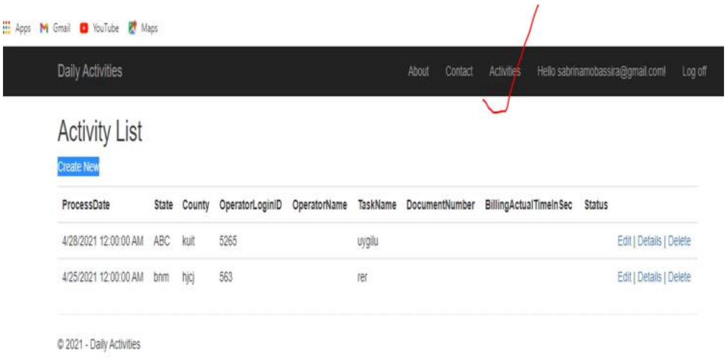
☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

**Fig 4.8: Index2 creation**

Then, I registered it to the login partial as ‘activities’ page in the browser. So, the menu bar of the web page executed a new page which is activities. And this page now contains the whole table and the properties.



Activity List

| ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeInSec | Status | |
|---|---|---|---|---|---|---|---|---|---|
| 4/28/2021 12:00:00 AM | ABC | kuit | 5265 | | uygilu | | | | Edit \| Details \| Delete |
| 4/25/2021 12:00:00 AM | bnm | hjcj | 563 | | rer | | | | Edit \| Details \| Delete |

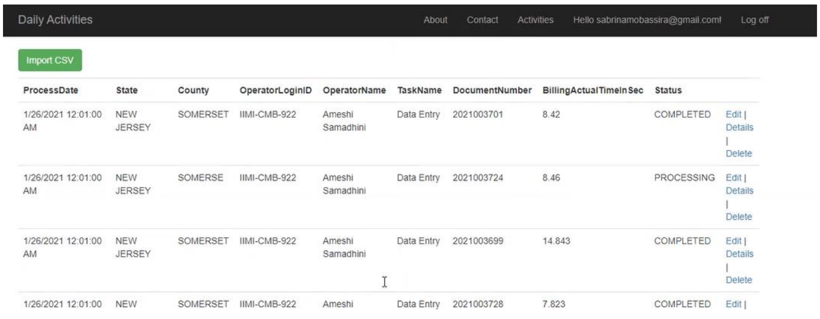
© 2021 - Daily Activities

**Fig 4.9: New page in the web**

## Import data:

For Import the data, we have followed three steps. They are 1. file browse 2. file read and 3. data insert. Here in this project, the import file is a CSV file. We can import a data table with all properties.

First of all, we created an import button on the index2 (activities) page. That was an action link in HTML. This import button represents a hyperlink or hypertext anchor button. With this button, we can browse, read, and import data. In the activity controller, we created a post method for index2. So, this method is for importing data.



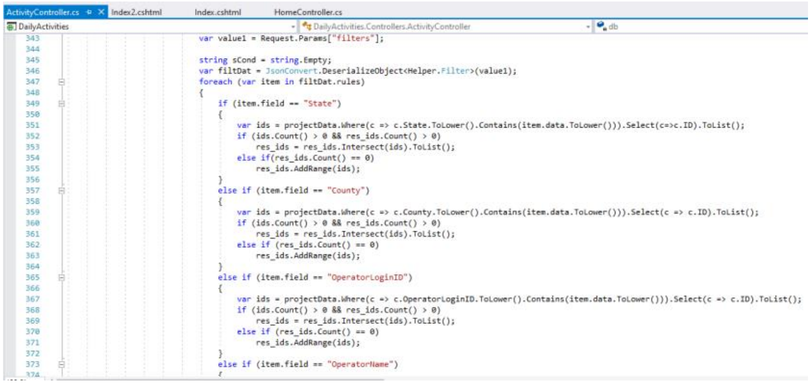
| ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeInSec | Status | |
|---|---|---|---|---|---|---|---|---|---|
| 1/26/2021 12:01:00 AM | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003701 | 8.42 | COMPLETED | Edit \| Details \| Delete |
| 1/26/2021 12:01:00 AM | NEW JERSEY | SOMERSE | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003724 | 8.46 | PROCESSING | Edit \| Details \| Delete |
| 1/26/2021 12:01:00 AM | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003699 | 14.843 | COMPLETED | Edit \| Details \| Delete |
| 1/26/2021 12:01:00 | NEW | SOMERSET | IIMI-CMB-922 | Ameshi | Data Entry | 2021003728 | 7.823 | COMPLETED | Edit \| |

**Fig 4.10: import data without grid view**

## Grid view:

The grid view of the project is on the activities page of the web application. Without grid view, we cannot browse or find out any data from the web page. There will be lots of data and lists. So, we have to filter all those fields and properties.

Here in the excel file, there are nine fields. We fixed all those fields in the activity controller code page (Fig 4.11) for the grid view as well as we can filter all of them individually.



```
var value1 = Request.Params["filters"];

string sCond = string.Empty;
var filtDat = JsonConvert.DeserializeObject<Helper.Filter>(value1);
foreach (var item in filtDat.rules)
{
    if (item.field == "State")
    {
        var ids = projectData.Where(c => c.State.ToLower().Contains(item.data.ToLower())).Select(c=>c.ID).ToList();
        if (ids.Count() > 0 && res_ids.Count() > 0)
            res_ids = res_ids.Intersect(ids).ToList();
        else if(res_ids.Count() == 0)
            res_ids.AddRange(ids);
    }
    else if (item.field == "County")
    {
        var ids = projectData.Where(c => c.County.ToLower().Contains(item.data.ToLower())).Select(c => c.ID).ToList();
        if (ids.Count() > 0 && res_ids.Count() > 0)
            res_ids = res_ids.Intersect(ids).ToList();
        else if (res_ids.Count() == 0)
            res_ids.AddRange(ids);
    }
    else if (item.field == "OperatorLoginID")
    {
        var ids = projectData.Where(c => c.OperatorLoginID.ToLower().Contains(item.data.ToLower())).Select(c => c.ID).ToList();
        if (ids.Count() > 0 && res_ids.Count() > 0)
            res_ids = res_ids.Intersect(ids).ToList();
        else if (res_ids.Count() == 0)
            res_ids.AddRange(ids);
    }
    else if (item.field == "OperatorName")
    {
```

**Fig 4.11: Grid Mechanism code**

In the index 2 page we used many types of code like, <Script> = J Query, <Style> = CSS, HTML codes and @ < > @ or @ is used for C# language.

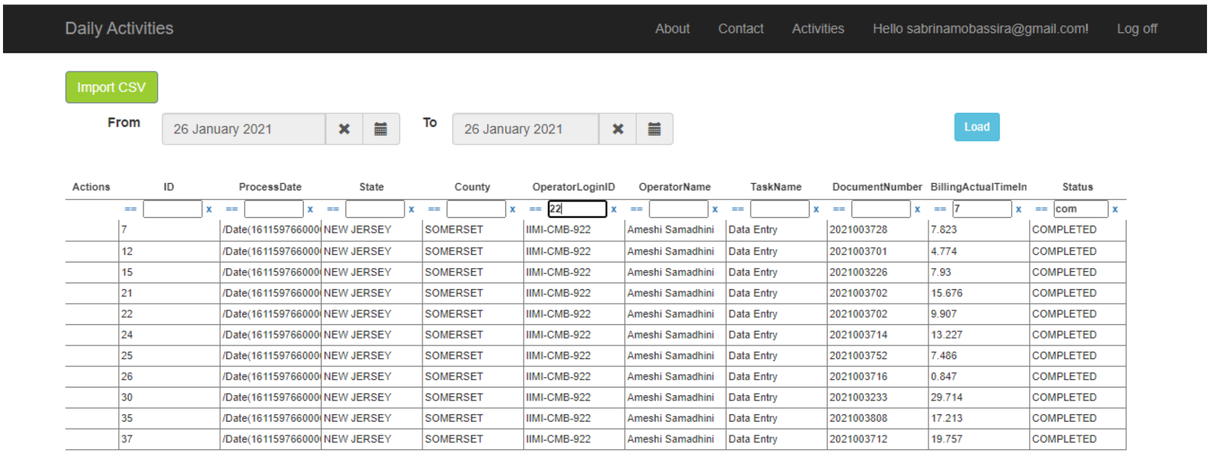
## Date wise filter in the grid:

In the grid, we can filter dates wise too. We need to select a particular date of the activities which we are going to find. There are two options for a date filter. One from which date the task started and the ending date. So, we can search the data according to the date (Fig 4.12).

## Load button:

The load button works as a search button of the date-wise or field-wise filter.

To change the color of any button, we created a style class and selected a color pattern. For example, ‘import CSV or ‘load’ button. Then we registered the class method in the controller. These are in bootstrap theme and works for the web page style.



From 26 January 2021 ✖ To 26 January 2021 ✖

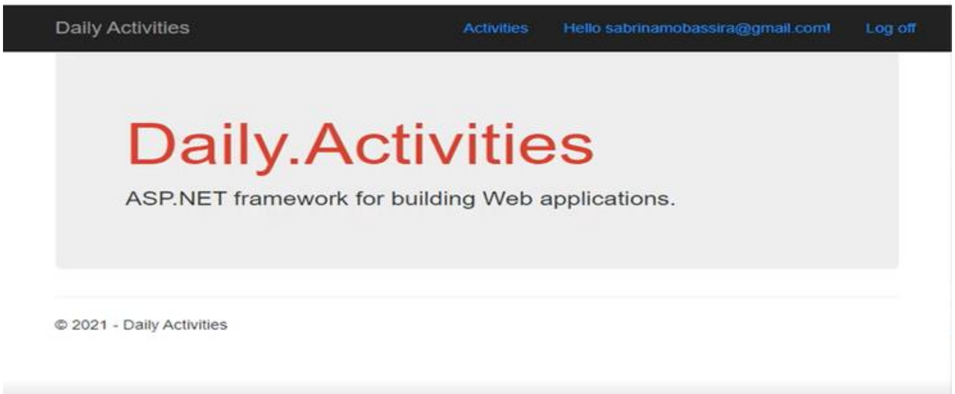
| Actions | ID | ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeIn | Status |
|---|---|---|---|---|---|---|---|---|---|---|
|  | == x | == x | == x | == x | == 22 x | == x | == x | == x | == 7 x | == com x |
|  | 7 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003728 | 7.823 | COMPLETED |
|  | 12 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003701 | 4.774 | COMPLETED |
|  | 15 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003226 | 7.93 | COMPLETED |
|  | 21 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003702 | 15.676 | COMPLETED |
|  | 22 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003702 | 9.907 | COMPLETED |
|  | 24 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003714 | 13.227 | COMPLETED |
|  | 25 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003752 | 7.486 | COMPLETED |
|  | 26 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003716 | 0.847 | COMPLETED |
|  | 30 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003233 | 29.714 | COMPLETED |
|  | 35 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003808 | 17.213 | COMPLETED |
|  | 37 | /Date(161159766000( | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003712 | 19.757 | COMPLETED |

**Fig 4.12: Grid view with date-wise filter and field-wise filter**

## Home page and menu bar organization & Color:

Anything that we want to change on the home page we need to edit code in the home controller index page. The project name, button and colors all can change on the index page.

And to change the menu bar we need to edit the code on the login partial page of the shared folder. For example, I do not need the about and contact icon on the menu bar. So, I remove both of them from here.



Daily.Activities

ASP.NET framework for building Web applications.

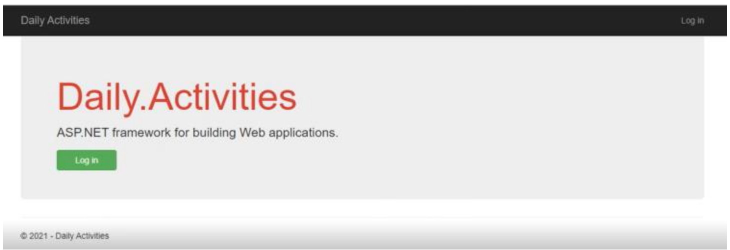
© 2021 - Daily Activities

**Fig 4.13: Home page organization**

## Log in button:

We moved the login button from the menu bar to the home page (Fig 4.14). For this, we added the action link HTML code from the activity controller to the home controller index.

To change the color of any button here, we changed the bootstrap CSS color code. For example, "@class = btn btn -success" means green color. "@class = btn btn -danger" means red color.



Daily.Activities

ASP.NET framework for building Web applications.

© 2021 - Daily Activities

**Fig 4.14: Create login button**

## User should be created by admin:

The login authentication depends on the admin user. No self-registration is required for login. Everyone needs permission from the administration to log in to the page. We created a user role authentication on the menu bar as an administrator page. There created five classes in model, controller and view. And this administrator page shows a drop-down view on the web page (Fig 4.15).

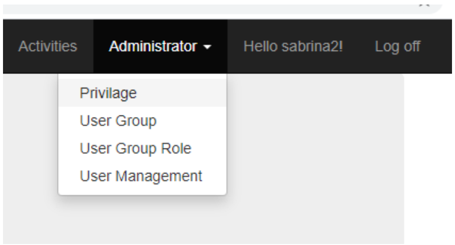
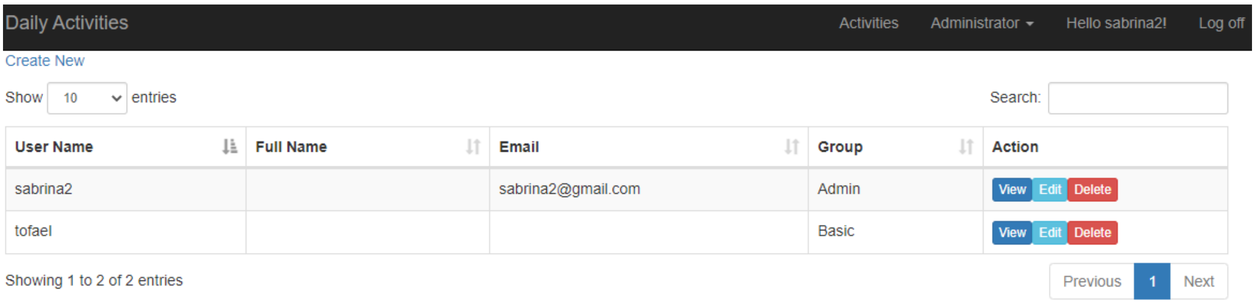


**Fig 4.15: Dropdown of user roles**

The user roles are a privilege, user group, user group role and user management. To access the registration of pages, we created a 'privilege page. In user management, we can fix our admin and basic user (Fig 4.16). So that admin and basic users can be one or more people.

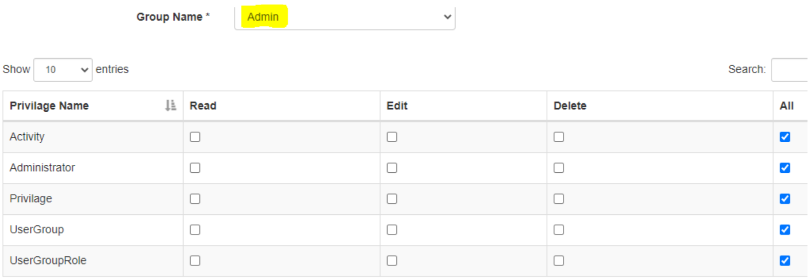


| User Name | Full Name | Email | Group | Action |
|---|---|---|---|---|
| sabrina2 | | sabrina2@gmail.com | Admin | View Edit Delete |
| tofael | | | Basic | View Edit Delete |

Showing 1 to 2 of 2 entries

**Fig 4.16: User management**

The admin can view, edit, delete any of the pages. On the other hand, the user only can view the 'activities' page in the web application. And we organized all these functions in the user group role page like in these pictures. (Fig 4.17 & Fig 4.18). So, if a user login to the page will not see the administrator page. But if an admin login to the page will see and edit all of those pages.



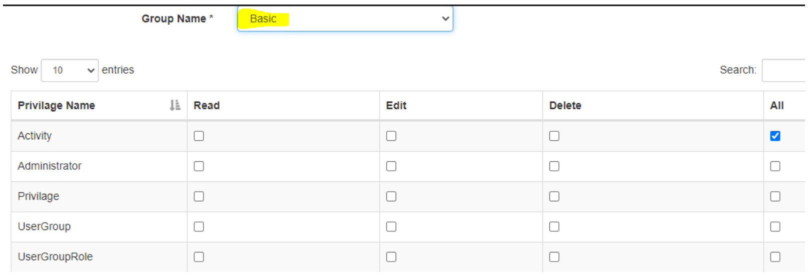
Group Name * Admin

Show 10 entries

Search:

| Privilage Name | Read | Edit | Delete | All |
|---|---|---|---|---|
| Activity | □ | □ | □ | ☑ |
| Administrator | □ | □ | □ | ☑ |
| Privilage | □ | □ | □ | ☑ |
| UserGroup | □ | □ | □ | ☑ |
| UserGroupRole | □ | □ | □ | ☑ |

**Fig 4.17: permission all for admin**



Group Name * Basic

Show 10 entries

Search:

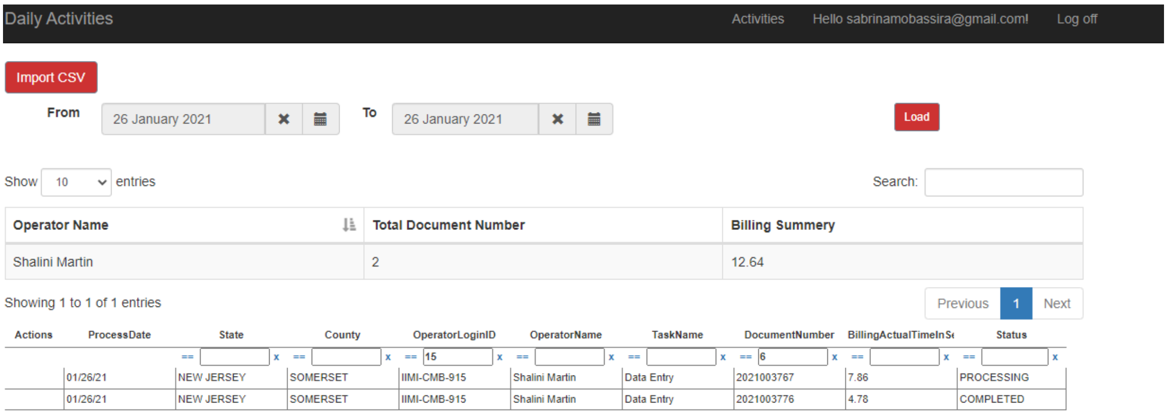
| Privilage Name | Read | Edit | Delete | All |
|---|---|---|---|---|
| Activity | □ | □ | □ | ☑ |
| Administrator | □ | □ | □ | □ |
| Privilage | □ | □ | □ | □ |
| UserGroup | □ | □ | □ | □ |
| UserGroupRole | □ | □ | □ | □ |

**Fig 4.18: permission just for activity page**

## Total count of document number & sum of the actual time:

We created another grid view on the index2 page or activities web page. This grid can filter out its field. Users can count the document numbers according to their search list in the grid. One employee's tasks, document name, date, status, time filter out by the grid field. Users also can count the total document number of all those selected documents for example, here in the

(Fig 4.19) it is 2. The Sum of the time of those tasks and activities also can count by this grid view.



Daily Activities

From 26 January 2021 To 26 January 2021

Show 10 entries

Search:

| Operator Name | Total Document Number | Billing Summery |
|---|---|---|
| Shalini Martin | 2 | 12.64 |

Showing 1 to 1 of 1 entries

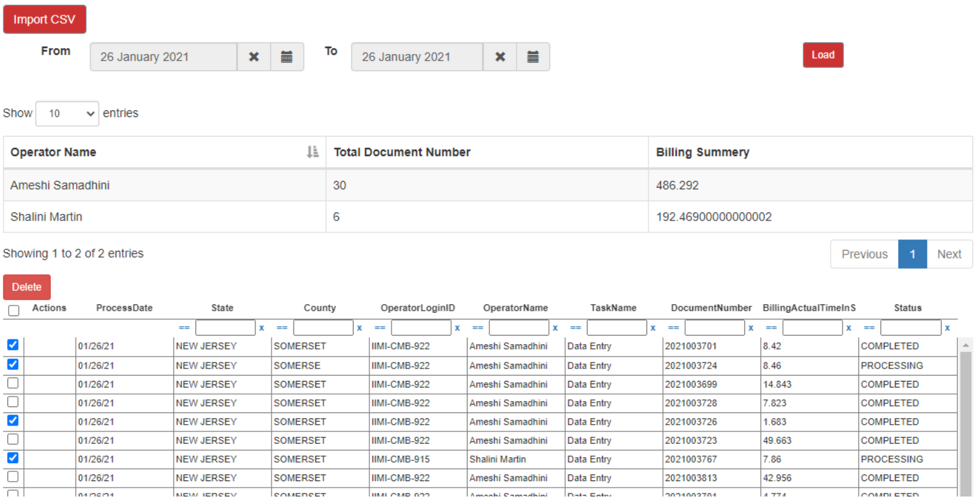
| Actions | ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeInSe | Status |
|---|---|---|---|---|---|---|---|---|---|
| | | == x | == x | == 15 x | == x | == x | == 6 x | == x | == x |
| | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-915 | Shalini Martin | Data Entry | 2021003767 | 7.86 | PROCESSING |
| | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-915 | Shalini Martin | Data Entry | 2021003776 | 4.78 | COMPLETED |

**Fig 4.19: Count of document number and billing summary**

## Delete data from the grid:

Users can delete the old data or unnecessary data from the grid as well as from the database. The organization will import lots of data about their daily tasks, activities and information. Users can delete one or more selected data from the grid.

So, here in the grid, all the data are organized by the fields, properties and working date. In the ‘action’ field of the grid view, we can select some or all options for delete (Fig 4.20). And user can delete those data by the delete button. On the index2 page, we created a delete button and select icons. We used this kind of code line in index2 for selection. For example, multiselect: true.



| Operator Name | Total Document Number | Billing Summery |
|---|---|---|
| Ameshi Samadhini | 30 | 486.292 |
| Shalini Martin | 6 | 192.46900000000002 |

Showing 1 to 2 of 2 entries

| □ | Actions | ProcessDate | State | County | OperatorLoginID | OperatorName | TaskName | DocumentNumber | BillingActualTimeInS | Status |
|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003701 | 8.42 | COMPLETED |
| ☑ | | 01/26/21 | NEW JERSEY | SOMERSE | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003724 | 8.46 | PROCESSING |
| □ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003699 | 14.843 | COMPLETED |
| □ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003728 | 7.823 | COMPLETED |
| ☑ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003726 | 1.683 | COMPLETED |
| □ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003723 | 49.663 | COMPLETED |
| ☑ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-915 | Shalini Martin | Data Entry | 2021003767 | 7.86 | PROCESSING |
| □ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003813 | 42.956 | COMPLETED |
| □ | | 01/26/21 | NEW JERSEY | SOMERSET | IIMI-CMB-922 | Ameshi Samadhini | Data Entry | 2021003701 | 4.774 | COMPLETED |

**Fig 4.20: Delete button and select option in the grid**

## Add more column in grid:

I can add more columns to the complete grid of this web page. When we have completed a grid view on the web page but still have to add one or more columns/ fields, we should follow these steps according to the MVC pattern:

- Add that new field in the model code (daily activities) with another line like other fields.
- Update database in package manager console.
- Add that new field in the 'get field' method of the controller code (Activity controller) with another line like other fields.
- Add grid filter code in the "jq grid mechanism" in the activity controller.

## Project publication:

We published our web project for company's purpose. It was their organizational project.

First of all, we have to install IIS express on the PC to publish the web project. To install IIS, I have fixed some windows features. The selected features are internet information service, in the web management tool: IIS 6 management compatibility and IIS management server. Moreover, in the world wide web services, application development features are .NET extensibility 3.5 & 4.8, application initialization, ASP, ASP.NET 3.5 & 4.8, CGI, ISAPI extensions and filter. After that, the windows publication requirements have changed.

Now in the visual studio, there is an option of project publication. It's an app service publication and the configuration we selected for this publication is 'release'.

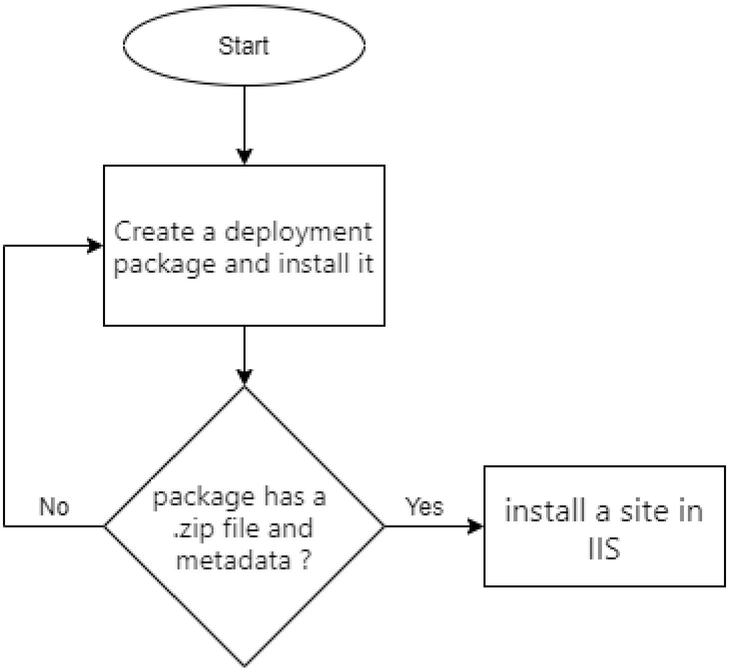

Start
Create a deployment package and install it
package has a .zip file and metadata ?
No
Yes
install a site in IIS

**Fig 4.21: Project publication flowchart**

## 4.3. Purpose of project:

The purpose of the project is to create an online web page where the employee can update their daily activities in an advanced way with the help of administrative regulation. The page will be monitored and observed by the company as well as the employees. It will be organized in a way where the login is restricted by the admin as anyone cannot log in to the page by their own. After getting the authentication every employee can share their information or activities and can check regularly if needed. If they want to delete any data they can select and delete it. This project aims to create a time-sparing environment for the company and an easily organized activity documentary of the employees.

# Chapter 5

## 5. Conclusion:

This internship program created an opportunity for me to explore the real professional environment. Where I was able to utilize my theoretical knowledge to learn and make something practical out of it. Moreover, since I had to work with a supervisor, at the very beginning I felt few problems regarding communication and interaction because of the new place and environment, however gradually I have managed to cooperate with the supervisor and my software development team members. Whenever I felt and had difficulties, my supervisor was always there to sort things out. Lastly, I was working on a project which is about organizational daily activities. Initially, it has been a process of my self-learning and self-development. But finally, I was able to be involved in a web application project with the software development team. Later on, I will work on this project to enrich some more requirements. This application encourages me to build up my career in the software development sector.

## Reference:

[1] "Tutorial: Get Started with Entity Framework 6 Code First using MVC 5", *Docs.microsoft.com,* 2021. [Online]. Available: https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application. [Accessed: 26- Sep- 2021].

[2] "IIMI Home - Intelligent Image Management", *Intelligent Image Management*, 2021. [Online]. Available: https://www.iimdirect.com/. [Accessed: 26- Sep- 2021].

[3] "Create ASP.NET MVC Application", *Tutorialsteacher.com*, 2021. [Online]. Available: https://www.tutorialsteacher.com/mvc/create-first-asp.net-mvc-application. [Accessed: 26- Sep- 2021].