

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and
Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By

ALENA THOMAS JOSEPH KURIEN, (Reg. No. 38110021)

SHARON ANN MATHEW, (Reg. No. 38110683)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI
SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ALENA THOMAS JOSEPH KURIEN (Reg. No: 38110021) & SHARON ANN MATHEW, (Reg. No: 38110683)** who carried out the project entitled “**ASSET MANAGEMENT SYSTEM**” under my supervision from September 2021 to March 2022.

Internal Guide

Mrs. C.M. Suja Cherukullapurath Mana

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We **ALENA THOMAS JOSEPH KURIEN** (Reg. No. **38110021**) and **SHARON ANN MATHEW**, (Reg. No: **38110683**) hereby declare that the Professional Training Report on “**ASSET MANAGEMENT SYSTEM**” done by us under the guidance of **Mrs. C.M. Suja Cherukullapurath Mana** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 24/03/2022

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T.Sasikala M.E., Ph.D.**, Dean School of Computing, **Dr.S.Vigneshwari M.E., Ph.D.**, and **Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mrs. C.M. Suja Cherukullapurath Mana**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Organizations face a huge challenge in monitoring the details, quantity, working condition, depreciation, and maintenance status of assets in their organization. The issue is faced not only by large organizations but also in smaller organizations. Assets are kept in records by means of inventory. The need to monitor these records of assets constantly is efficient management of the assets inventory of an organization to maximize the organization's value resulting in established and intangible assets to be located easily, in a more reliable and efficient manner. It helps with tracking the whereabouts, quantity, working conditions, depreciation, and maintenance status of all assets. A detailed list of all assets in the organization displaying their description, inventory number, respective location, working condition (status) and all other details will be always available.

With the advent of ever evolving technology, today's fast-paced world has adapted itself to the convenience of having data digitalized and easily accessible. This growth has influenced majority of organizations to have their very own digital database systems over the years. It overcomes the manual-based spreadsheet system, which is usually characterized by errors, loss of data, varying reports, or difference in results at stages involved and helps cut down maintenance costs by managing all the information efficiently.

The Asset Management System is a project that aims at creating an application that will be used to store data of the Organization's mobile assets, devices, fixtures, inventory, maintenance activities etc.

The front end provides an interactive forefront for the managing authority of the Organization to access the inventories information through a simple login via a website developed using HTML, CSS, PHP. The back end consists of all detailed record of assets in the organization which is achieved by creating the database using MySQL, PHP, XAMPP.

The application was tested and evaluated for usability and functionality using appropriate dummy test data and can be accessed from any terminal with Internet access and the login credentials.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	1
	1.1 BASICS	1
	1.2 EFFECTS & BENEFITS	2
	1.3 IMPORTANCE OF AMS	3
	1.4 MySQL	4
	1.5 PHP	4
	1.6 XAMPP	5
	1.7 IMPROVISATION METHODS	5
	1.8 EVALUATION	6
	1.9 APPLICATION	7
	1.10 NEED FOR ASSET MANAGEMENT	8
2	AIM AND SCOPE OF THE PROJECT	10
	2.1 AIM	10
	2.2 SCOPE OF PROJECT	10
	2.3 SKILLS REQUIRED	11
	2.4 BACKEND & FRONTEND DESIGN	11
	2.4.1 MySQL SERVER (DATABASE)	12
	2.4.2 HTML	12
3	EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS	13
	3.1 INSTALLATION OF XAMPP AND VS	13
	3.1.1 XAMPP	13
	3.1.2 VISUAL STUDIO FOR FRONTEND	13
	3.2 DATABASE MANAGEMENT	14
	3.2.1 DATABASE APPLICATION	14
	3.2.2 DATA MODEL	15
	3.2.3 MODULES	15
	3.2.4 DOCUMENT MANAGEMENT & IMPORT	15
	3.2.5 DEPLOYMENT & TESTING	16
	3.3 DEVELOPMENTAL TOOLS	16

	3.3.1 FRONTEND DESIGN – HTML, CSS, JS	16
	3.3.2 BACKEND DEVELOPMENT	17
	3.3.3 SECURITY FEATURE - SMS/CALL API	17
	3.4 HARDWARE REQUIREMENTS	18
	3.4.1 COMPUTER	18
	3.4.2 INTERNET ACCESS	18
	3.4.3 DEVELOPMENT PLATFORM	18
	3.4.4 DATABASE SERVER	18
	3.4.5 GUI DEVELOPMENT PLATFORM	18
	3.4.6 SOFTWARE	18
4	RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS	19
	4.1 SITES	20
	4.1.1 WB02	20
	4.2 BASIC ARCHIVES	20
	4.2.1 COMPANY	20
	4.2.2 EMPLOYEE	21
	4.2.3 CORPORATE GROUP	21
	4.2.4 INTERNAL EMPLOYEE	21
	4.2.5 EXTERNAL EMPLOYEE	22
	4.2.6 SUPPLIER	22
	4.4 TWILIO CALL / SMS API	23
5	SUMMARY AND CONCLUSION	24
	5.1 OVERVIEW	24
	5.1.1 BROWSER	24
	5.1.2 BROWSER CONFIGURATION	24
	5.1.3 OPERATING SYSTEM	24
	5.2 CONCLUSION	25
	REFERENCES	26
	APPENDIX	27
	(A) SOURCE CODE	27
	MySQL CODE FOR DATABASE	31
	HTML / CSS	34
	BACKEND CODE	37
	CALL / SMS API	40

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1	STRUCTURE OF ASSET MANAGEMENT SYSTEM	2
1.2	MYSQL	4
1.3	PHP	4
1.4	XAMPP	5
1.5	WORKFLOW OF ASSET MANAGEMENT SYSTEM	8
4.1	LOGIN PAGE	19
4.2	DASHBOARD	19
4.3	SITES	20
4.3.1	WB02	20
4.4.1	COMPANY	20
4.4.2	EMPLOYEE	21
4.4.3	CORPORATE GROUP	21
4.4.4	INTERNAL EMPLOYEE	21
4.4.5	EXTERNAL EMPLOYEE	22
4.4.6	SUPPLIER	22
4.5	TWILIO CALL / SMS API	23

CHAPTER 1

INTRODUCTION

1.1 BASICS

Asset management refers to the process of developing, operating, maintaining, and selling assets in a cost-effective manner. Most used in finance, the term is used in reference to individuals or firms that manage assets on behalf of individuals or other entities.

It involves comprehensive and structured approach to the long-term management of assets as tools for the efficient and effective delivery of community benefit. For an organization to manage its assets efficiently all assets owned by the organization must be registered and recorded in the organization's assets inventory. Assets Inventory Management is any system that monitors and maintains things of value to an entity or group. It may apply to both tangible assets such as buildings and to intangible concepts such as intellectual property and goodwill. It is a systematic process of deploying, operating, maintaining, upgrading, and disposing of assets cost effectively.

Efficient management of the assets inventory of any organization maximizes the organization's value and makes assets easier to locate and manage.

The system employed in managing the assets inventory varies from organization to organization. Some organizations make use of the spreadsheet system to keep track of the assets they own. This method of assets management is time wasting and is prone to errors that the personnel may not even be aware he made. Other organizations make use of Assets Inventory Management software which automates the manual spreadsheet system. These software vary from organization to organization based upon the requirement of each individual organization. The requirement includes the size of the organization and the volume of they own.

The manual system employed by organizations involve the physical movement of their audit staff from office to office manually count the assets residing in each office before entering them into the spreadsheet. This method wastes time, is characterized by errors and it is difficult to keep inventory. records updated if assets are transferred

often. An outdated assets inventory register allows for the presence of ghost assets. Sometimes, ghost assets are sold off by employees in secret while in other cases are stashed away somewhere. Although these assets are no longer being used, the organization still pays for maintenance and insurance on some of them. Furthermore, as the number of asset acquisitions grows, it becomes increasingly impractical and more difficult to track the location, working status, transfers, disposals, and adjustments to these assets. To ensure an accurate, well detailed, up-to-date, and secure database of assets present in an organization, an alternative is to use a web-based inventory management system where records can be digitally archived, thereby reducing filing activity at the end of each term's end.



Fig 1.1 Structure of Asset Management System

1.2 EFFECTS & BENEFITS

An effective asset management process helps businesses monitor and manage their assets using a systematized approach. With a solid process, organizations can improve productivity and efficiency of an asset thus enhancing the return on improvement.

Key components of an effective asset management process include:

- i) Preparing an asset register: An inventory list of available assets
- ii) Identify asset conditions and rating system

- iii) Track asset depreciation accurately at timely intervals
- iv) Determine the value of assets and their replacement cost
- v) Move away from reactive to predictive asset maintenance
- vi) Focus more on change management to ensure database integrity
- vii) Eliminate paperwork and automate tasks like data collection, asset reporting, and more.

Smart businesses don't waste time trying to complete basis and repetitive asset management tasks manually by using the same outdated process repeatedly. Investing in asset management software allows businesses to automate tedious and time-consuming asset management activities offering administration team to focus more on value-added activities.

Benefits of automating asset management:

- i) Prolongs the life of your assets
- ii) Aids in rehabilitating, repairing, and replacing assets efficiently
- iii) Meets consumer demands with a focus on system sustainability
- iv) Helps focus on activities that are critical to sustained performance
- v) Enables businesses to meet service expectations and regulatory requirements
- vi) Improves responses to emergencies or unexpected risks

1.3 IMPORTANCE OF ASSET MANAGEMENT SYSTEM

- i) Enables a firm to account for all its assets

The process makes it easy for organizations to keep track of their assets, whether liquid or fixed. Firm owners will know where assets are located, how they are being used, and whether there have been changes made to them. Consequently, the recovery of assets can be done more efficiently, hence, leading to higher returns.

- ii) Helps guarantee the accuracy of amortization rates

Since assets are checked on a regular basis, the process of asset management ensures that the financial statements record them properly.

- iii) Helps identify and manage risks

Asset management encompasses the identification and management of risks that arise from the utilization and ownership of certain assets. It means that a firm will always be prepared to manage any risk that comes its way.

iv) Removes ghost assets in the company's inventory

Instances exist where lost, damaged, or stolen assets are erroneously recorded on the books. With a strategic asset management plan, the firm's owners will be aware of the assets that have been lost and will eliminate them in the books.

1.4 MySQL

A relational database management system that uses SQL (Structured Query Language). Used commonly as a web database, it may hold anything from a single piece of information to a whole inventory of accessible items. MySQL database, being an open-source Relational Database Management System (RDBMS) that employs SQL language, helps to automate data retrieval, and provides excellent assistance in PHP MySQL web application development.



Fig 1.2 My SQL

1.5 PHP

PHP (Hypertext Preprocessor) is a general-purpose program scripting language used for creating dynamic and interactive web pages. It is a server-side language that can be integrated in HTML, making it easy to add functionality to web sites without having to access other files for data.



Fig 1.3 PHP

1.6 XAMPP

XAMPP is an open-source and cross-platform virtual server solution stack bundle that includes the Apache web server, MySQL database, and PHP. The integration of PHP and MySQL requires no setup. For development purposes, you can use a personal web server to work locally from your own laptop or PC instead of an external server.



Fig 1.4 XAMPP

1.7 IMPROVISATION METHODS

Implementing an asset management system will not deliver results if the process itself is mediocre at best. So, before looking for ways to automate the process, it is critical to streamline their existing asset management practices.

Factors an organization need to focus on to improvise their asset management:

- i) Understand the reason behind asset's existence
 - Purpose the asset serves
 - Reason behind procurement
- ii) Gain knowledge about the asset's condition
 - Asset's current condition
 - Fulfillment of its purpose
 - Reliability
- iii) See how effectively the asset is being utilized
 - Current asset delivery value
 - Fulfillment of existing expectations

iv) Identify future demands

- If there will be a change of service expectations
- Robust enough to deliver future demands

v) Assess the existing maintenance program

- Maintenance and improvement methods on the asset's lifecycle
- Effectiveness of the current asset maintenance program
- Accuracy of estimate for maintenance costs

vi) Predict depreciation rate and associated risks

- If the useful life of the asset reviewed periodically
- Difference in expectations significantly from previous estimates
- Depreciation charge for future periods adjustments
- Impact of the depreciation

vii) Forecast asset disposal

- Expected timeline for the asset to be permanently withdrawn from use
- Gain or loss from the disposal of the asset noted in the financial statement

1.8 EVALUATION

1. Systems Integration

The information collected in your asset management solution is only as useful as its application. Most packages integrate well with other systems, such as your project management software, or ERP. The more integrated your data, the more it can be used for business intelligence and predictive analytics.

2. User interface

The user interface (UI) must be accessible to everyone you expect to use it. If people find the software difficult to use, they won't. Consider your workforce and how likely or not they are to adopt this technology, bearing the future in mind as well. There may be

a balance between functionality and accessibility. Evolve the interface along with your staff by using an adaptable solution. With the capability for role-based logins—employees will only have to interact with the data they need to.

3. Support

Whichever asset management solution you choose, it will become an integral part of your organization—with the potential for significant disruption if it goes down. Before committing, find out what support is available, what the SLAs are, and how much help is provided with implementation. Every business is different—you'll likely need some assistance configuring the system for your needs such as service managers and technicians can easily remotely monitor, manage, diagnose and resolve issues for equipment in the field. It is important to find a strong technology partner that will provide continuous support.

4. Scalability & Overall flexibility

It is essential determine how great the asset management solution can scale. Licenses and implementation are the biggest issues—how easy is it to add or remove licenses. How easy is it to install or access the software on new devices. Space availability to manage different assets in the future. The last thing you want is to migrate to a new solution at a critical juncture in your organization's growth.

1.9 APPLICATION

Smaller operations may benefit from a computerized maintenance management system. The automation software assists with scheduling, management and reporting of maintenance activities. Features include handling workflows, resourcing, and routing, operating and repair guidance, and reporting and auditing.

For large operations, an enterprise asset management system provides a central platform for managing all fixed assets. It integrates asset data from across the asset lifecycle: acquisition, operations, maintenance, depreciation and renewal or replacement.

This application will be used for the management of mobile assets such as different types of infrastructure, plants, and technical devices like; electric panels, heating and

cooling, light fixture, fire extinguishers, etc., count and details of furniture, and the related logistical, economical and maintenance activities for both scheduled and corrective ones.

This management system will help the supervising authority to keep track, organize & update the data pertaining to their asset inventories and support in their operative procedures.

The management system will be made as per the requirements ensuring complete security and non-breach of data.

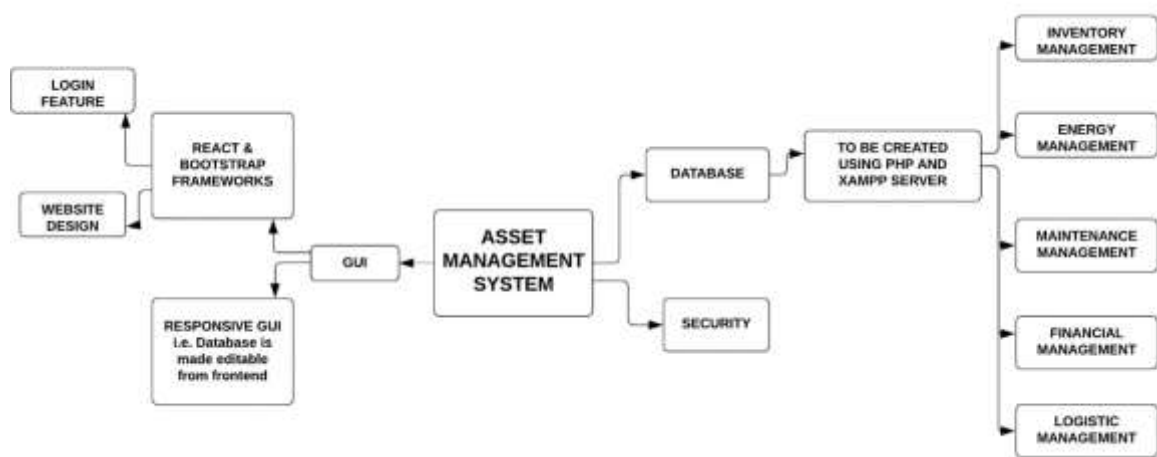


Fig 1.5 Workflow of the Asset Management System

1.10 NEED FOR ASSET MANAGEMENT

Asset management has always been a tedious and lengthy process in the past since it has been very difficult to keep an account of large amounts of data whilst keeping track of its pertaining data constantly. Not only has the process been difficult but also making it securely accessible has been one of the major concerns.

Within the framework of computerized memory management systems, asset management is now much easier, digitalized, secured and, easily accessible.

All details such as data regarding the mobile assets, counts of devices, fixtures scheduled, inventory available, maintenance activities done previously as well as the ones scheduled are all available and stored together under one asset management application for an organization.

Unlike in the non-digitalized versions where there was a high risk of misplacing data or having data leaked, the digitalized version ensures that there are logs to keep track of every data entered and deleted as well as a recovery option to recover data if it has been deleted incorrectly.

There is higher security since the data can only be accessed by a person who has been given authority to access the data which results in guarantee of supervision of access and having no data breached.

An asset management system is the solution to having all assets related data of an organization together under one umbrella in a secured and simple login access manner.

CHAPTER 2

AIM AND SCOPE OF THE PROJECT

2.1 AIM

To create an Asset Management System wherein the database end will be created using MySQL and XAMPP server and the GUI of the web application will be scripted in HTML, CSS and JS along with React and Bootstrap framework for the management of mobile assets and inventory of an organization as per their requirements along with a database alteration alert trigger voice/ call API as an added security feature.

2.2 SCOPE OF PROJECT

In the digital age, organizations face massive challenges in keeping track of the whereabouts, quantity, working condition, depreciation and maintenance status of things of value in their organization. This problem is faced not only by large organizations but also the medium and smaller organizations. Those things of value known as assets are kept in records by means of inventory. The need to monitor constantly these records of assets cannot be over emphasized as efficient management of the assets inventory of any organization helps maximize the organization's value by making fixed and intangible assets to be easier to locate, more reliable and efficient.

These employs tracking the whereabouts, quantity, working condition, depreciation and maintenance status of assets. This is to have at all times, a detailed list of all assets in the organization displaying their description, serial number, acquisition cost, inventory number, respective location, working condition (status) amongst others. This overcomes the manual-based spreadsheet system, which is characterized by error(s), loss of data, varying reports or results at stages involved in the manual-based spreadsheet system.

In this paper, top-down design methodology was used to proffers solution to the problem using a number of tools such as XAMPP and MySQL for the developmental phase. The application was tested and evaluated for usability and functionality using

appropriate test data. The front end provides an interactive means for the user of the system to input information into the database. The database (back end) provides a detailed record of all assets in the organization displaying their description, serial number, acquisition cost, inventory number, respective location, working condition (status) amongst others. The complete web-based system could be used from any terminal with internet access.

Also, functionality such as updating assets' details and adding new locations and offices to the database are all possible through the web interface.

2.3 SKILLS REQUIRED

- (i)** Creativity
- (ii)** Imagination
- (iii)** Good knowledge on database, frontend web development
- (iv)** Relevant software knowledge

2.4 BACKEND (DATABASE) AND FRONT-END DESIGN

This comprises of both Functional and Non-Functional requirements. The functional requirement for the proposed Assets Inventory Management System is a web-based application, which means it has a web based platform. HTML, CSS and JS has been used as the technology in developing this application. The database query language used in this application is the Structured Query Language (SQL). MySQL as the backend database for the system's main engine. These technologies have been selected based on their capabilities, flexibility and modularity in developing an online system. The Non-Functional Requirements of the proposed software application is expected to have certain properties which would ensure its productivity; amongst these properties are reliability, consistency, and accuracy. This application has been tested to pass these requirements.

2.4.1 MySQL SERVER (DATABASE)

SQL Server developed by Microsoft, is their more expensive but more robust alternative to MS Access. MySQL Server is more commonly used by businesses for small to medium sized databases. MySQL Server is a powerful database with a lot of functionality and it also has built in security features. There are free 'express versions' of MySQL Server that can be downloaded for free; however these versions offer less in functionality. MySQL Server utilizes Transact-SQL as its primary query language. MySQL Server allows procedures to be stored within the database similar to MS Access queries. This reduces network traffic as only variables are sent to the database along with the corresponding procedure to be used and not entire queries. This is particularly useful for queries which will be used regularly. This is also a benefit for the security of the system as it does not allow SQL injection attacks. Similar to MS Access, MySQL Server only works on Windows based systems which may be restrictive for users. A major disadvantage of MS SQL Server is that it is very expensive and not commonly used by web hosts.

2.4.2 HTML

The **HyperText Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

It was teamed along with CSS and Javascript to enhance the appearance of the web application. Bootstrap and React framework was included to make the frontend editable.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS

3.1 INSTALLATION OF XAMPP AND VS

3.1.1 XAMPP

Step 1) Go to URL: <https://www.apachefriends.org/download.html> for XAMPP server download and select the windows version and download for installation.

Step 2) Run the .exe file

Step 3) Deactivate any antivirus software

Step 4) Start the setup wizard

Step 5) Choose the location

Step 6) Finish the installation process

3.1.2 Visual Studio for Frontend

Step 1) Download the Visual Studio Community Version 2019

Step 2) Run the .exe file and follow the instructions to install Visual Studio Community Version on the system.

Step 3) Select ASP.NET and web development from the options and click to install in bottom right corner as shown below. We can also select .NET desktop development option for windows forms and console applications etc. Here we are selecting both the options. We can also modify it after installation.

Step 4) Click on launch and it will be prompted to sign in for the first time. The sign-in step is optional so it can be skipped. The dialog box will appear for the first time only and ask to choose the Development Settings and color theme. Once select required options, click on Start Visual Studio option.

Step 5) To create a new ASP.NET Web application, Go to File → New → Project

Step 6) As soon as we select the project, we will notice the different options of the Project. We are simply choosing ASP.NET Web Application(.NET Framework) and clicking Next. We can see the options of C#, Windows, and Library below the chosen project.

Step 7) The next step is to configure the project. Here, we have to choose the Project Name and Solution name and click on Create Button. A project name is a subset of the solution name. We can put a different name for the solution.

Step 8) Choose the type of the ASP.NET Web Application. We are creating a web application so first, choose the project type as Empty for a simple application. Then choose the Web-Forms which will add the basic folders to create Web Forms Application. After that click Create button.

Step 9) Now add a Web Form file to the project “name” which contains the web-specific code for the project. Select Add and then select Web Form from the menu as shown below.

Step 10) Now write the code in the .aspx file which will display the output.

Step 11) Now to execute the code, click on the Run button and output is found.

3.2 DATABASE MANAGEMENT

Database Management System is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

DBMS allows users to create their own databases as per their requirement. The term “DBMS” includes the user of the database and other application programs. It provides an interface between the data and the software application.

3.2.1 Database Application

Database application is a computer program whose primary purpose is retrieving information from a computerized database. From here, information can be inserted,

modified or deleted which is subsequently conveyed back into the database. A characteristic of modern database applications is that they facilitate simultaneous updates and queries from multiple users.

A database application with a Web interface had the advantage that it could be used on devices of different sizes, with different hardware, and with different operating systems.

3.2.2 Data Model

Complete freedom in the database modeling, with information of interest:

- Types of assets
- Functional data card contents
- Technical data card contents
- Administrative data card contents
- Usable "templates" for the creation of new types of objects.

3.2.3 Modules

- (i) Space and Asset Inventory
- (ii) GIS Support
- (iii) Energy & Environment
- (iv) Facility Maintenance
- (v) Logistic Management

3.2.4 Document Management & Import

Document archive management:

- Basic documents associated with the properties
- Building practices for works
- Technical manuals
- Accounting and administrative documents
- Images and photographs
- Files of any other type

3.2.5 Deployment & Testing

Deployment in software and web development means pushing changes or updates from one deployment environment to another. When setting up a website you will always have your live website, which is called the live environment or production environment.

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system to identify any gaps, errors, or missing requirements contrary to the actual requirements. The extent of testing varies greatly between organizations, developers, and individual sites or applications.

3.3 DEVELOPMENTAL TOOLS

A suitable programming environment needs to be selected for system development. The chosen languages should be capable of developing a graphical user interface to display suitable output to the end users. The chosen language should include support for object-oriented implementations. Booch (1998) stated that 'Each object can be viewed as an independent little machine with a distinct role or responsibility'. As the system will comprise of several components implementing via an object-oriented language would allow for more efficient data parsing through each component. Developing an application using a high-level object-oriented language enables more flexibility, making complex programming simpler than coding in low level languages. This would fulfill the extensibility requirement of the framework, making it easier to adapt and modify in the future. There are many programming and scripting languages available which meet these criteria.

3.3.1 Front-end Design – HTML, CSS, JS

HTML is at the core of every web page, regardless the complexity of a site or number of technologies involved. It's an essential skill for any web professional. It's the starting point for anyone learning how to create content for the web. CSS stands for Cascading Style Sheets. This programming language dictates how the HTML elements of a website should actually appear on the frontend of the page. JavaScript is a more complicated language than HTML or CSS, and it wasn't released in beta

form until 1995. Nowadays, JavaScript is supported by all modern web browsers and is used on almost every site on the web for more powerful and complex functionality.

3.3.2 Back-end Development

Backend is the server-side of the website. It stores and arranges data, and makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

The back-end portion is built by using My SQL and XAMPP Server.

3.3.3 Security Feature – SMS / Call API

SMS Alert

To send a SMS from Php we would need a Twilio account, and a trial account would suffice. We just need to register and verify our phone number in Twilio and get our own Twilio number. Now, whenever there is an insertion deletion or an update in a table a SMS will be send from our Twilio account to the registered mobile number. What we need is to create a stored PROCEDURE and make a post request to Twilio account from the registered account credentials. Then create a TRIGGER to execute the procedure whenever there is an alteration in a particular table.

Call Alert

With the registered Twilio account we can also make a voice call. We need a C# program which connects to Twilio using the accounts credentials. Then using Visual Studio, we convert the program to an executable file. Then we place the executable file in the path of SQL server. When there is an insertion, deletion, or an update in a table a trigger called `voice` is triggered and a job called `call` job activates the executable file. The executable file makes the voice call to the registered mobile number.

3.4 HARDWARE REQUIREMENTS

Building an asset management system has become a much more feasible task for the average user with the availability of free creation software and services. With the software readily available, the only other things required are a few key hardware components. Luckily, highly specialized, and expensive equipment is not needed to create one.

3.4.1 Computer

Most current computers and laptops have high enough specifications to be used to create an asset management system. It will be used for prototype development and will run both the graphical front-end and database end.

3.4.2 Internet Access

The computer must have Internet access for connection to the remote MS SQL database back-end.

3.4.3 Development Platform

The development platform is Windows operating system with internal Bluetooth module with a local testing server installed (Apache).

3.4.4 Database Server

MS SQL server is used to create the database end and connect. It is more commonly used by businesses for small to medium sized databases. MS SQL Server is a powerful database with a lot of functionality, and it also has built in security features.

3.4.5 GUI Development Platform

Visual Studio is used to design the front-end GUI for the website to which the databases are connected to make sure that it is available all the time to the end user who is trying to access it.

3.4.6 Software

To create the database and host on server - XAMPP

To create the web application front end - Visual Studio

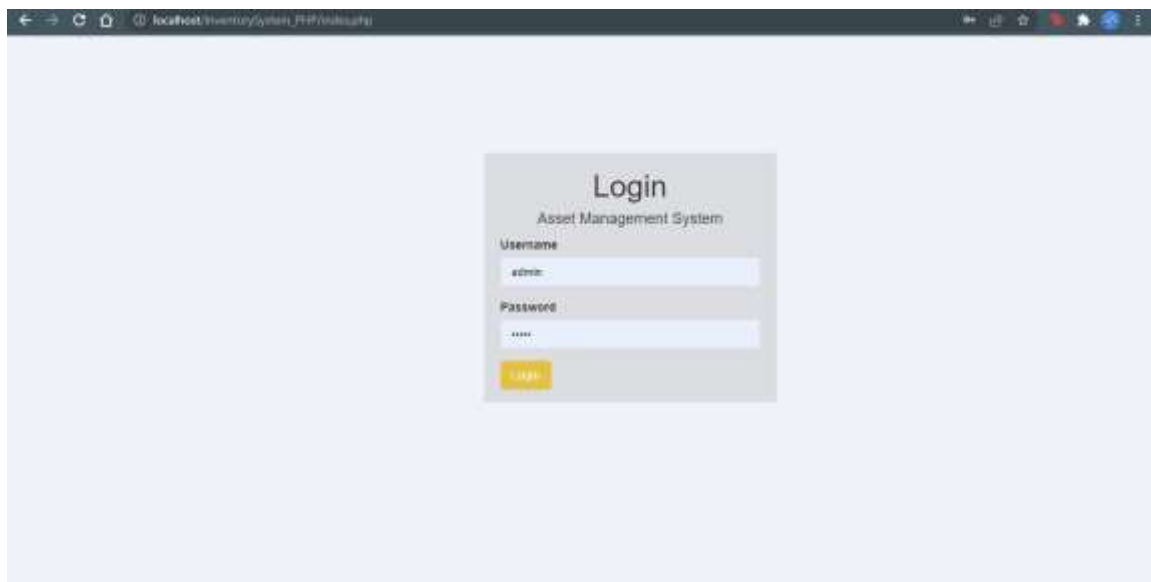
CHAPTER 4

RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS

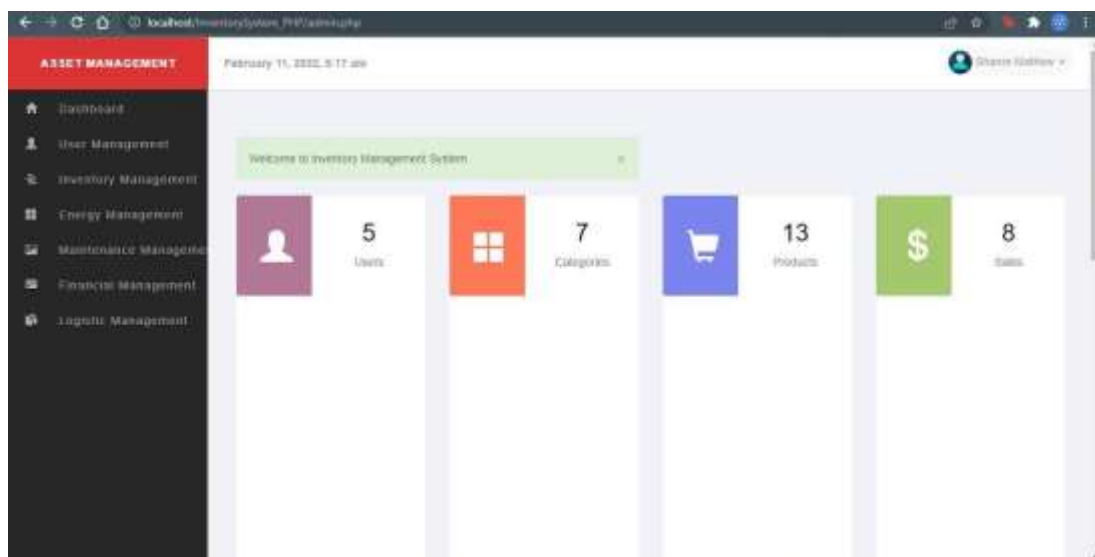
It is an Asset Management System with database created using a MySQL and XAMPP server and GUI of the web application scripted in HTML, CSS, JS for the management of mobile assets and inventory of an organization as per their requirements.

DEMO SCREENSHOTS:

4.1 Login Page



4.2 Dashboard



4.3 Sites

Sites Hierarchy									
Add new building									
Search...									
Code	Name	Main use	Availability	Address	ZIP Code	City	Service status	Calendar entry	
TT01	Terronella headquarters	Office	In use	Via L. Aquila, 1/A		Tevegnazzo			
WB01	Office Building 01	Office	In use	Murazzano 12/1	38100	Rovereto			
WB02	Office Building 02	Office	In use	Matteotest 12/2	38100	Rovereto			
CH01	Chaletan House	Residential	In use	3000 Granville Lane	07701	Red Bank			
WB14	Duplex Apartment 14	Residential	Not	77A Long Lane	08128	Bene			

4.3.1 WB02

Floor of WB02 - Office Building 02						
Add new floor						
Search...						
Code	Name	Level	Building	Service status	Calendar entry	
SB	Ground floor	0	WB02 - Office Building 02			
S1E	1 - First floor	1	WB02 - Office Building 02			
S0B	First floor base	1.5	WB02 - Office Building 02			
S2E	Second floor	2	WB02 - Office Building 02			
S2E	Second floor base	2.5	WB02 - Office Building 02			
S3E	Third floor	3	WB02 - Office Building 02			
S3M	Third floor covering	3.5	WB02 - Office Building 02			
S3B	Third floor base	3.5	WB02 - Office Building 02			

4.4 Basic Archives

4.4.1 Company

Cards Company					
Add new Company					
Search...					
Subtype	VAT	Business name	Country	Phone	E-mail
Supplier	9954233	myco1.com			info@myco1.com
Supplier	9575463	myco2			info@myco2.com
Corporate group	0123456789	MyCompany - Branch 1			mycompany.branch1@example.com
Corporate group		MyCompany - Branch 2			mycompany.branch2@example.com
Supplier	45341235	MyCompany			mycompany@example.com
Supplier	1234567	myco3			info@myco3.com
Supplier	7890123	Powerly			info@powerly.com
Supplier	5678901	WebDev			info@webdev.com

4.4.2 Employee

Cards: Employee										
Add card Employee										
Search...										
Subtype	Code	Number	Last name	First name	Email	Phone	State	Company	Service sta...	Calendar entry
Employee...		81729832	Sam	john	john.sam@e...		Active	MyCompany		
Internal e...	798241984	81724021	Gamer	Damian	damian.gam...		Active	MyCompany		
Internal e...	908012122	81724023	Hayes	Julia	julia.hayes@...		Active	MyCompany		
Internal e...		81729831	Howard	Ashley	ashley.howar...		Active	MyCompany		
Internal e...	888802299	81724020	Robins	Matthew	matthew.robi...		Active	MyCompany		

4.4.3 Corporate Group

Cards: Corporate group					
Add card Corporate group					
Search...					
id	Business cards	Country	Phone	E-mail	
8122444789	MyCompany - Branch 1			mycompany.branch1@example.com	
	MyCompany - Branch 2			mycompany.branch2@example.com	

4.4.4 Internal Employee

Cards: Internal employee									
Add card Internal employee									
Search...									
Code	Number	Last name	First name	Email	Phone	State	Company	Service status	Calendar entry
798241984	81724021	Gamer	Damian	damian.gam@e...		Active	MyCompany - Br...		
908012122	81724023	Hayes	Julia	julia.hayes@e...		Active	MyCompany - Br...		
888802299	81724020	Robins	Matthew	matthew.robi...		Active	MyCompany - Br...		

4.4.5 External Employee

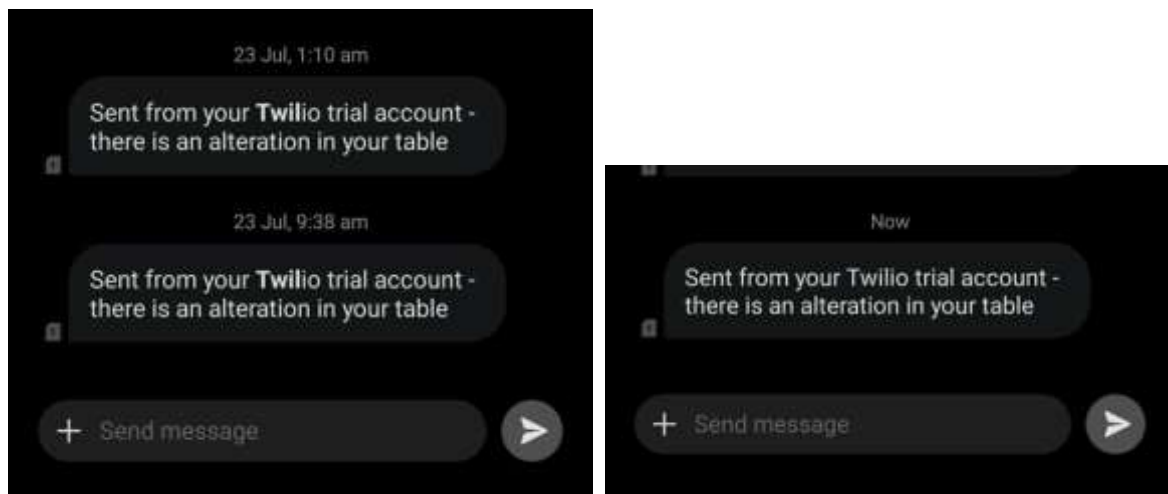
Cards External employee								
+ Add card External employee		Search...						2 items
Code	Number	Last name	First name	Email	Phone	State	Company	Service status
	EKT299832	Bauer	Jason	jason.bauer@ext...		Active	MyConsultants	
	EKT299831	Howard	Adelaide	adelaide.howard@...		Active	MyConsultants	

4.4.6 Supplier

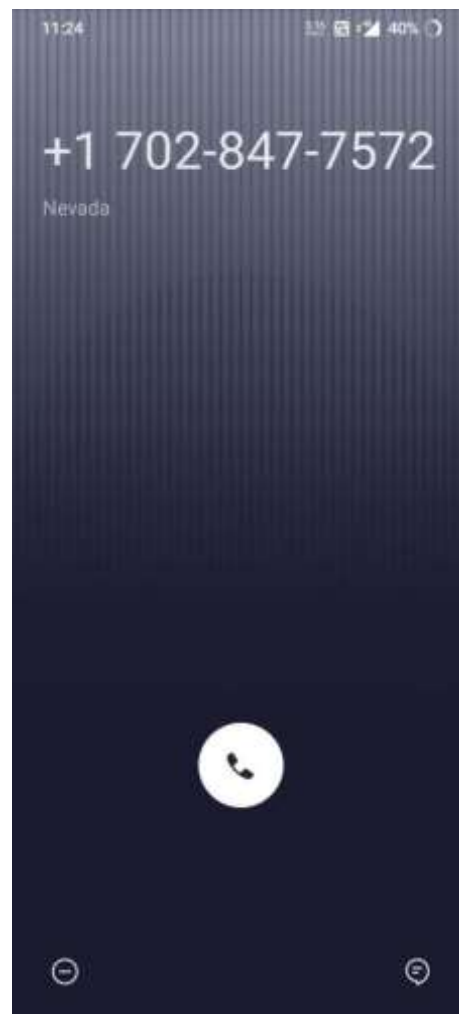
Carts Supplier				
+ Add cart supplier		Search...	    	6 Items
VAT	Name	Country	Phone	E-mail
99564233	HydroCluster			info@hydrocluster.com
99754462	InterSite			info@intersite.com
45561235	MyConsultants			myconsultants@ext.example.com
12557466	Nearability			info@nearability.com
73922672	Powerly			info@powerly.com
56744255	Voltfuel			info@voltfuel.com

4.5 TWILIO Call / SMS API

i) SMS alert



ii) Call alert



CHAPTER 5

SUMMARY AND CONCLUSION

5.1 OVERVIEW

5.1.1 BROWSER

- Chrome* 36+
- Edge* 20+
- Mozilla Firefox 31+
- Internet Explorer 11+ (Windows only)
- Safari 6+ (MacOS only)

5.1.2 BROWSER CONFIGURATION

Your browser must be configured as follows:

- Cookies must be enabled.
- Pop-up windows must be enabled.

5.1.3 OPERATING SYSTEM

- Windows 7, Windows 8 or Windows 10
- Mac OSX 10.8, 10.9, 10.10 or 10.11

5.2 CONCLUSION

In this paper, we explored the potential of what front-end (HTML, CSS, JavaScript, React), and client-side rendering coupled with a Real-time MySQL, PHP database hosted on XAMPP Server can do. By this way we can create a powerful and secured management system which will never glitch even when there are hundreds of users concurrently.

The Asset Management System was designed to decrease human error and boost efficiency. The primary goal of this study is to reduce human effort while increasing efficiency. The maintenance of records is made more efficient since all records are saved in a database, from which data can be quickly retrieved.

The navigation control given in all interfaces enables users to simply traverse through the enormous number of records. This system will enable automation of the Assets Inventory Management process by moving all activities online for convenience. This solution eliminates the problems that existed in the manual system, such as inconsistent asset reporting, spreadsheets, the appearance of ghost assets, and unwanted time consumption.

The automation of Assets Inventory Management will not only increase efficiency, but will also reduce human stress, and create an increase in resources available for effective management of inventory.

REFERENCES:

- Demo website of Asset Management System previously created using Java & Python: <https://demo.openmaint.org/>
- Sample screenshots of existing application: <https://www.openmaint.org/en/resources/screenshots>
- <https://www.openmaint.org/en/download/latest-version>
- <https://archive.codeplex.com/?p=sqlserversamples>
- <https://www.codeproject.com/Questions/308591/Send-an-email-from-SQL-Server-Trigger>
- <https://forums.asp.net/t/2024897.aspx?Create+insert+trigger+to+call+URL>
- Elizaveta Gavrikova , Irina Volkova and Yegor Burda, "Strategic Aspects of Asset Management: An Overview of Current Research" in The Future of Facilities Management and Sustainable Development, 2020.
- Dwi Puspita Sari and Alex Elentukh, "Systematic Implementation of ASM (Asset Management System)" in The 6th International Conference on Cyber and IT Service Management (CITSM), 2018.
- Sanyapong Petchrompo, "A Review of Asset Management Literature on Multi-Asset Systems" in Reliability Engineering and System Safety, 2018.
- Afifah Muftinisa, Rosalina, Rikip Ginanjar, RB. Wahyu and Nur Hadisukmana, "Development and Implementation of Fixed Asset Management System" in Second International Conference on Informatics and Computing (ICIC), 2017.
- C.B.H. Nel and J.L. Jooste, "A TECHNOLOGICALLY-DRIVEN ASSET MANAGEMENT APPROACH TO MANAGING PHYSICAL ASSETS - A LITERATURE REVIEW AND RESEARCH AGENDA FOR 'SMART' ASSET MANAGEMENT", in South African Journal of Industrial Engineering December 2016 Vol 27(4), 2016.

APPENDIX:

(A) SOURCE CODE

MySQL Code for Database

(i) *Sites Table*

```
CREATE TABLE Sites (  
    Code varchar(100),  
    Name varchar(100),  
    Availability varchar(100),  
    Main_use varchar(100),  
    Complex varchar(100),  
    Address varchar(200),  
    City varchar(100)  
);
```

```
insert into Sites (Code, Name, Availability, Main_use, Complex, Address, City)  
values('TT01','Tecnoteca headquarters','In use','Office','CPX01 - Friuli-  
Venezia Giulia Center','Via LAquila, 1/A','Tavagnacco'),  
( '01','STO Gambir','In use','Office','OB - Office Buildings','PT Administrasi  
Medika Telkom Gambir, C Building. Jl. Medan Merdeka Selatan No. 12  
Jakarta Pusat DKI Jakarta 10110 Indonesia','Jakarta'),  
( 'WB01','Office Builing 01','In use','Office','OB - Office Buildings','Mainstreet  
12/1','Florence'),  
( 'WB02','Office Builing 02','In use','Office','OB - Office Buildings','Mainstreet  
12/2','Florence');
```

(ii) *WB02 Table*

```
CREATE TABLE WB02OfficeBuilding02 (  
    Code varchar(255),  
    Name varchar(255),  
    Level varchar(255),  
    Building varchar(255)  
);
```

```
insert into WB02OfficeBuilding02 (Code, Name, Level, Building)  
values ('55A','Ground Floor','0','WB02-Office Building 02'),  
( '55E','1- First Floor','1','WB02-Office Building 02'),  
( '55B','First Floor Base','1.5','WB02-Office Building 02'),  
( '55F','Second Floor','2','WB02-Office Building 02');
```

(iii) *Company Table*

```
CREATE TABLE Company(  
    Subtype varchar(100),  
    VAT varchar(200),  
    Business varchar(100),  
    Country varchar(100),  
    Phone varchar(100),  
    Email varchar(200)  
);
```

```
insert into Company (Subtype, VAT, Business, Country, Phone, Email)  
values ('Supplier', '99564233', 'HydroCluster', '', '', 'info@hydrocluster.com'),  
('Supplier', '99754462', 'InterSite', '', '', 'info@intersite.com'),  
('Corporate Group', '0123456789A', 'MyCompany - Branch 1', '', '',  
'mycompany-branch1@example.com');
```

(iv) *Employee Table*

```
CREATE TABLE Employee(  
    Subtype varchar(100),  
    Code varchar(200),  
    Number varchar(200),  
    Last_name varchar(100),  
    First_name varchar(100),  
    Email varchar(200),  
    Phone varchar(100),  
    State varchar(100),  
    Company varchar(200),  
    Service_status varchar(100),  
    Calendar_entity varchar(100)  
);
```

```
insert into Employee(Subtype, Code, Number, Last_name, First_name, Email,  
Phone, State, Company, Service_status, Calendar_entity)  
values ('External employee', '', 'EXT299832', 'Bauer', 'Jason',  
'jason.bauer@ext.example.com', '', 'Active', 'MyConsultants', '', ''),  
('Internal employee', '7982419964', 'INT240221', 'Garret', 'Damian',  
'damian.garret@example.com', '', 'Active', 'MyCompany - Branch 1', '', ''),  
('Internal employee', '9285521262', 'INT240223', 'Hayes', 'Julia',  
'julia.hayes@example.com', '', 'Active', 'MyCompany - Branch 1', '', '');
```

(v) *Corporate Group*

```
CREATE TABLE Corporategroup(  
    VAT varchar(200),
```

```

    Business_name varchar(200),
    Country varchar(100),
    Phone varchar(200),
    Email varchar(200)
);

```

```

insert into Corporategroup(VAT, Business_name, Country, Phone, Email)
values ('0123456789A', 'MyCompany - Branch 1', '', 'Bauer', '', 'mycompany-branch1@example.com'), ('', 'MyCompany - Branch 2', '', '', 'mycompany-branch2@example.com');

```

(vi) *Internal Employee*

```

CREATE TABLE InternalEmployee(
    Code varchar(200),
    Number varchar(200),
    Last_name varchar(100),
    First_name varchar(100),
    Email varchar(200),
    Phone varchar(100),
    State varchar(100),
    Company varchar(200),
    Service_status varchar(100),
    Calendar_entity varchar(100)
);

```

```

insert into InternalEmployee(Code, Number, Last_name, First_name, Email,
Phone, State, Company, Service_status, Calendar_entity)
values ('7982419964', 'INT240221', 'Garret', 'Damian',
'damian.garret@example.com', '', 'Active', 'MyCompany - Branch 1', '', ''),
('9285521262', 'INT240223', 'Hayes', 'Julia', 'julia.hayes@example.com', '',
'Active', 'MyCompany - Branch 1', '', ''),
('3558262288', 'INT240225', 'Rollins', 'Matthew',
'matthew.rollins@example.com', '', 'Active', 'MyCompany - Branch 1', '', '');

```

(vii) *External Employee*

```

CREATE TABLE externalEmployee (
    Code varchar(255),
    Number varchar(255),
    LastName varchar(255),
    FirstName varchar(255),
    email varchar(255),
    phone varchar(255),
    state varchar(255),

```

```

        company varchar(255)
    );

insert into externalEmployee(Code,
Number,LastName,FirstName,email,phone,state,company)
values
('','EXT299382','Bauer','Jason','jason.bauer@ext.example.com','','Active','MyC
onsultants'),
('','EXT299381','Howard','Adelaide','howard.adelaide@ext.example.com','','Act
ive','MyConsultants');

```

(viii) *Supplier Database*

```

CREATE TABLE supplier (
    VAT varchar,
    Name varchar(255),
    Country varchar(255),
    phone varchar(255),
    email varchar(255)
);

insert into supplier(VAT,Name,Country,phone,email)
values('99564233','HydroCluster','','','info@hydrocluster.com'),
('99754462','InterSite','','','info@intersite.com'),
('45561543','MyConsultants','','','info@myconsultants.com');

```

(ix) *Supplier Employee Database*

```

CREATE TABLE supplier_Employee (
    Code varchar(255),
    Number varchar(255),
    LastName varchar(255),
    FirstName varchar(255),
    email varchar(255),
    phone varchar(255),
    state varchar(255),
    company varchar(255)
);

```

(x) *Customer Table*

```

CREATE TABLE Customer (
    VAT varchar(255),
    BusinessName varchar(255),
    Country varchar(255),

```

```
        phone varchar(255),
        email varchar(255)
    );
```

(xi) *Customer Employee Table*

```
CREATE TABLE customer_Employee (
    Code varchar(255),
    Number varchar(255),
    LastName varchar(255),
    FirstName varchar(255),
    email varchar(255),
    phone varchar(255),
    state varchar(255),
    company varchar(255)
);
```

HTML / CSS

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1" />
```

```
    <title>Asset Management</title>
```

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css"/>
```

```
<style>
```

```
    body {
```

```
        font-family: "Lato", sans-serif;
```

```
        background-color: rgb(228, 218, 218);
```

```
    }
```

```
    header
```

```
    {
```

```
        background-color: #cf8319;
```

```
padding:5px;
font-size: 10px;
color: white;
overflow: hidden;
}
#main h1{
margin-left: 5px;
}
#main a{
float: right;
color: white;
font-size: 15px;
padding: 5px;
text-decoration: none;
justify-content: center;
margin-left: 45px;
}
.sidenav {
height: 100%;
width: 200px;
position: fixed;
z-index: 1;
top: 0;
left: 0;
background-color:#818181;
overflow-x: hidden;
padding-top: 20px;
}
.sidenav a {
padding: 6px 8px 6px 16px;
text-decoration: none;
```

```

font-size: 15px;
color: white;
display: block;
}
.sidenav a:hover {
color: #f1f1f1;
font-size: 17px;
}

#main {
margin-left: 200px;
padding: 3px;
}

@media screen and (max-height: 450px) {
.sidenav {padding-top: 15px;}
.sidenav a {font-size: 18px;}
}
</style>
</head>
<body>
<form id="form1" runat="server">
<header>
<div class="sidenav">
<a href="#about"><i class="fa fa-folder"></i> Facilities and Assets</i></a>
<a href="#services"><i class="fa fa-folder"></i> Basic Archives</a>
<a href="#clients"><i class="fa fa-folder"></i> Maintenance
Management</a>
<a href="#contact"><i class="fa fa-folder"></i> Financial Management</a>
</div>

<div id="main">

```

<h1>Asset Management</h1>

<p> </p>

```
<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" BackColor="LightGoldenrodYellow"
BorderColor="Tan" BorderWidth="1px" CaptionAlign="Left" CellPadding="2"
DataSourceID="SqlDataSource1" ForeColor="Black" GridLines="None"
Height="436px" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting" OnRowEditing="GridView1_RowEditing"
OnRowUpdating="GridView1_RowUpdating" ShowFooter="True" style="margin-left:
32px; margin-right: 34px" Width="846px">
```

```
<AlternatingRowStyle BackColor="PaleGoldenrod" />
```

```
<Columns>
```

```
<asp:TemplateField HeaderText="Code">
```

```
<EditItemTemplate>
```

```
<asp:Label ID="Label5" runat="server" Text='<%# Eval("Code")
%>'></asp:Label>
```

```
</EditItemTemplate>
```

```
<FooterTemplate>
```

```
<asp:TextBox ID="TextBox5" runat="server"
Width="34px"></asp:TextBox>
```

```
</FooterTemplate>
```

```
<ItemTemplate>
```

```
<asp:Label ID="Label1" runat="server" style="text-align: center"
Text='<%# Eval("Code") %>'></asp:Label>
```

```
</ItemTemplate>
```

```
</asp:TemplateField>
```

```
<asp:TemplateField HeaderText="Name">
```

```
<EditItemTemplate>
```

```
<asp:TextBox ID="TextBox2" runat="server" Text='<%#
Eval("Name") %>'></asp:TextBox>
```

```
</EditItemTemplate>
```

```
<FooterTemplate>
```

```
<asp:TextBox ID="TextBox6" runat="server"
Width="41px"></asp:TextBox>
```

```
</FooterTemplate>
```

```

        <ItemTemplate>
            <asp:Label ID="Label2" runat="server" style="text-align: center"
Text='<%# Eval("Name") %>'></asp:Label>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Level">
        <EditItemTemplate>
            <asp:TextBox ID="TextBox3" runat="server" Text='<%#
Eval("Level") %>'></asp:TextBox>
        </EditItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="TextBox7" runat="server"
Width="46px"></asp:TextBox>
        </FooterTemplate>
        <ItemTemplate>
            <asp:Label ID="Label3" runat="server" style="text-align: center"
Text='<%# Eval("Level") %>'></asp:Label>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="Building">
        <EditItemTemplate>
            <asp:TextBox ID="TextBox4" runat="server" Text='<%#
Eval("Building") %>'></asp:TextBox>
        </EditItemTemplate>
        <FooterTemplate>
            <asp:TextBox ID="TextBox8" runat="server"
Width="49px"></asp:TextBox>
        </FooterTemplate>
        <ItemTemplate>
            <asp:Label ID="Label4" runat="server" style="text-align: center"
Text='<%# Eval("Building") %>'></asp:Label>
        </ItemTemplate>
    </asp:TemplateField>

```

[illegible]

```

        <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:college %>" SelectCommand="SELECT
* FROM [officebuilding]"></asp:SqlDataSource>

        <span style="font-size:30px;cursor:pointer" onclick="openNav()">

            <br />

        </span>

        <a class="active" href="#home"><i class="fa fa-group"></i> Maintenance
Office</a>

    </div>

</header>

</form>

</body>

</html>

```

Backend Code

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection("Data Source=LAPTOP-
64BK0C5D\\SQLDEV;Initial Catalog=college;Integrated Security=True");
        if (con.State==ConnectionState.Open)

```

```
{
    con.Close();
}
con.Open();
}
```

```
protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    GridView1.DataSource = SqlDataSource1;
    GridView1.DataBind();
    GridView1.EditRowStyle.BackColor = System.Drawing.Color.Black;
}
```

```
protected void GridView1_RowCancelingEdit(object sender,
GridViewCancelEventArgs e)
{
    GridView1.EditIndex = -1;
    GridView1.DataSource = SqlDataSource1;
    GridView1.DataBind();

}
```

```
protected void GridView1_RowUpdating(object sender,
GridViewUpdateEventArgs e)
{
    Label code = GridView1.Rows[e.RowIndex].FindControl("Label5") as Label;
    TextBox name = GridView1.Rows[e.RowIndex].FindControl("TextBox2") as
TextBox;
    TextBox level = GridView1.Rows[e.RowIndex].FindControl("TextBox3") as
TextBox;
```

```
        TextBox building = GridView1.Rows[e.RowIndex].FindControl("TextBox4") as
        TextBox;
```

```
        String mycon = "Data Source=LAPTOP-64BK0C5D\\SQLDEV;Initial
        Catalog=college;Integrated Security=True";
```

```
        String updatedata="Update officebuilding set
        Name='"+name.Text+"',Level='"+level.Text+"',Building='"+building.Text+" where
        Code='"+code.Text+""";
```

```
        SqlConnection con = new SqlConnection(mycon);
```

```
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand();
```

```
        cmd.CommandText = updatedata;
```

```
        cmd.Connection = con;
```

```
        cmd.ExecuteNonQuery();
```

```
        GridView1.EditIndex = -1;
```

```
        SqlDataSource1.DataBind();
```

```
        GridView1.DataSource = SqlDataSource1;
```

```
        GridView1.DataBind();
```

```
    }
```

```
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs
    e)
```

```
    {
```

```
        Label code = GridView1.Rows[e.RowIndex].FindControl("Label5") as Label;
```

```
        String mycon = "Data Source=LAPTOP-64BK0C5D\\SQLDEV;Initial
        Catalog=college;Integrated Security=True";
```

```
        String updatedata = "delete from officebuilding where Code=" + code.Text + """;
```

```
        SqlConnection con = new SqlConnection(mycon);
```

```
        con.Open();
```

```
        SqlCommand cmd = new SqlCommand();
```

```
        cmd.CommandText = updatedata;
```

```
        cmd.Connection = con;
```

```
        cmd.ExecuteNonQuery();
```

```
        GridView1.EditIndex = -1;
```

```
        SqlDataSource1.DataBind();
```

```

        GridView1.DataSource = SqlDataSource1;
        GridView1.DataBind();
    }

protected void LinkButton5_Click(object sender, EventArgs e)
{
    TextBox code = GridView1.FooterRow.FindControl("TextBox4") as TextBox;
    TextBox name = GridView1.FooterRow.FindControl("TextBox5") as TextBox;
    TextBox level = GridView1.FooterRow.FindControl("TextBox6") as TextBox;
    TextBox building = GridView1.FooterRow.FindControl("TextBox7") as TextBox;

    String mycon = "Data Source=LAPTOP-64BK0C5D\\SQLDEV;Initial
Catalog=college;Integrated Security=True";

    String query = "insert into officebuilding values('"+code.Text+"','"+name.Text+
"','"+level.Text+"','"+building.Text+"')";

    SqlConnection con = new SqlConnection(mycon);
    con.Open();

    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = query;
    cmd.Connection = con;
    cmd.ExecuteNonQuery();

    SqlDataSource1.DataBind();
    GridView1.DataSource = SqlDataSource1;
    GridView1.DataBind();
}
}

```

Call / SMS API

i) Twilio Call API

```

using System;
using Twilio;

```

```

using Twilio.Rest.Api.V2010.Account;

class Program
{
    static void Main(string[] args)
    {
        // Find your Account SID and Auth Token at twilio.com/console
        // and set the environment variables. See http://twil.io/secure
        string accountSid =
Environment.GetEnvironmentVariable("TWILIO_ACCOUNT_SID");
        string authToken =
Environment.GetEnvironmentVariable("TWILIO_AUTH_TOKEN");

        TwilioClient.Init(accountSid, authToken);

        var call = CallResource.Create(
            twiml: new Twilio.Types.Twiml("<Response><Say>Ahoy,
World!</Say></Response>"),
            to: new Twilio.Types.PhoneNumber("+14155551212"),
            from: new Twilio.Types.PhoneNumber("+15017122661")
        );
        Console.WriteLine(call.Sid);
    }
}

```

ii) Twilio SMS API

```

using System;
using Twilio;
using Twilio.Rest.Api.V2010.Account;

class Program
{

```

```
static void Main(string[] args)
{
    // Find your Account SID and Auth Token at twilio.com/console
    // and set the environment variables. See http://twil.io/secure
    string accountSid =
Environment.GetEnvironmentVariable("TWILIO_ACCOUNT_SID");
    string authToken =
Environment.GetEnvironmentVariable("TWILIO_AUTH_TOKEN");

    TwilioClient.Init(accountSid, authToken);

    var message = MessageResource.Create(
        body: "McAvoy or Stewart? These timelines can get so confusing.",
        from: new Twilio.Types.PhoneNumber("+15017122661"),
        statusCallback: new Uri("http://postb.in/1234abcd"),
        to: new Twilio.Types.PhoneNumber("+15558675310")
    );
    Console.WriteLine(message.Sid);
}
}
```