# Event Handler

Web site for book event and manage event by admin and manage admin by super admin

This web site is develop on asp.net, c#

|  |  |
| --- | --- |
| Author |  |
| Title | Event Handler |
| Degree | Computer Science |
| Instructors |  |
| Technologies | Asp .net , c#, SQL server |

### ABSTRACT

This paper describes a new monitoring and event management concept. Event management is the application of the management science of project management to the creation and development of festivals and events. Event Handler Event Management System lets clients and staffs register venues and events in a simple manner.

From registering of venue to informing attendees, Event Handler Event Management System simplifies the reservations process, enabling staff to operate at peak efficiency. And, when everyone is working together with complete, accurate, real-time information, customers receive the best possible service.

Generally, the project is about registering venues by vendors for events like meeting, exhibition, convocation etc and for clients to book venue for events. The staff will exist in registering attendees for particular events and send emails days before the event as reminder.

The design of the system was illustrated using several types of diagrams, namely Entity Relationship Diagram (ERD), Data Flow Diagram (DFD), and also Data Dictionary and Data Normalization. This project was written on c# and ASP.NET.

## 1.0 Introduction

Event management is the process by which an event is planned, prepared, and produced. I

I As with any other form of management, it encompasses the assessment, definition, acquisition, allocation, direction, control, and analysis of time, finances, people, products, services, and other resources to achieve objectives.

An event manager's job is to oversee and arrange every aspect of an event, including researching, planning, organizing, implementing, controlling, and evaluating an event's design, activities, and production

#### Table 02. Operation Environment for the Design and Sign System

|  |  |
| --- | --- |
| PROCESSOR | Intel Core Processor |
| OPERATING SYSTEM | WINDOWS7 Or Higher |
| MEMORY | 1GB RAM Or More |
| HARD DISK SPACE | Minimum 30 GB for Database Usage for future |
| DATABASE | SQL Server 2019 |
| TECHNICAL SKILLS | JavaScript, CSS, HTML, ASP.NET |
| IDE | Visual Studio 2019 |
| WEB SERVER | IIS 7.0 |

### ****Module Specification:****

* ****Admin****
  + Manage Admin User.
  + Mange Customer.
  + Manage Event
  + Manage Package.
  + Manage Employee.
  + Manage Booking.
  + View Feedback.
  + Generate Various Report.
* ****Customer****
  + Manage Profile.
  + View Event.
  + View Package.
  + Book Package.
  + Cancel Package.
  + Give Feedback.
* ****Guest****
  + View Event.
  + View Package.
  + Give Feedback.

# Development Environment

### Asp.net

ASP.NET is an open source web framework, created by Microsoft, for building modern web apps and services with .NET.

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

The ASP.NET application codes can be written in any of the following languages:

* C#
* Visual Basic.Net
* Jscript
* J#

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

## ASP.NET Web Forms Model

ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

* Page state
* Session state

The page state is the state of the client, i.e., the content of various input fields in the web form. The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields.

This way, the server becomes aware of the overall application state and operates in a two-tiered connected way.

## The ASP.NET Component Model

The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

Server side counterparts of almost all HTML elements or tags, such as <form> and <input>.

Server controls, which help in developing complex user-interface. For example, the Calendar control or the Gridview control.

ASP.NET is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.NET web application is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.

The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

## Components of .Net Framework 3.5

Before going to the next session on Visual Studio.Net, let us go through at the various components of the .Net framework 3.5. The following table describes the components of the .Net framework 3.5 and the job they perform:

|  |
| --- |
| **Components and their Description** |
| **(1) Common Language Runtime or CLR**  It performs memory management, exception handling, debugging, security checking, thread execution, code execution, code safety, verification, and compilation. The code that is directly managed by the CLR is called the managed code. When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just In Time(JIT) compiler compiles the IL code into native code, which is CPU specific. |
| **(2) .Net Framework Class Library**  It contains a huge library of reusable types. classes, interfaces, structures, and enumerated values, which are collectively called types. |
| **(3) Common Language Specification**  It contains the specifications for the .Net supported languages and implementation of language integration. |
| **(4) Common Type System**  It provides guidelines for declaring, using, and managing types at runtime, and cross-language communication. |
| **(5) Metadata and Assemblies**  Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory. Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files. |
| **(6) Windows Forms**  Windows Forms contain the graphical representation of any window displayed in the application. |
| **(7) ASP.NET and ASP.NET AJAX**  ASP.NET is the web development model and AJAX is an extension of ASP.NET for developing and implementing AJAX functionality. ASP.NET AJAX contains the components that allow the developer to update data on a website without a complete reload of the page. |
| **(8) ADO.NET**  It is the technology used for working with data and databases. It provides access to data sources like SQL server, OLE DB, XML etc. The ADO.NET allows connection to data sources for retrieving, manipulating, and updating data. |
| **(9) Windows Workflow Foundation (WF)**  It helps in building workflow-based applications in Windows. It contains activities, workflow runtime, workflow designer, and a rules engine. |
| **(10) Windows Presentation Foundation**  It provides a separation between the user interface and the business logic. It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations, and more. |
| **(11) Windows Communication Foundation (WCF)**  It is the technology used for building and executing connected systems. |
| **(12) Windows CardSpace**  It provides safety for accessing resources and sharing personal information on the internet. |
| **(13) LINQ**  It imparts data querying capabilities to .Net languages using a syntax which is similar to the tradition query language SQL. |

## The .NET platform

.NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications.

The base platform provides components that apply to all different types of apps. Additional frameworks, such as ASP.NET, extend .NET with components for building specific types of apps.

Here are some things included in the .NET platform:

* ****The C#, F#, and Visual Basic programming languages****
* ****Base libraries**** for working with strings, dates, files/IO, and more
* ****Editors and tools**** for Windows, Linux, macOS, and Docker

**.NET Framework**

.NET is a software development framework and ecosystem designed and supported by Microsoft to allow for easy desktop and web application engineering. It’s a popular free platform currently used for a lot of different types of applications as it provides the programming environment for most software development phases.

.NET best suits businesses that look for a wide range of features like web-based services, desktop software, and cloud infrastructure support.

Microsoft started working on the .NET framework in the late 90s. The idea was to create a platform based on so-called managed code, code that can be executed under a runtime environment. This was needed to improve the development experience and relieve engineers from handling security operations, active memory management, and other low-level efforts that C/C++ developers had to bother with.

The .NET Framework era

The first release of .NET Framework in 2002 introduced C#, a language for writing managed code that had a design similar to C++. The framework itself was aimed at Windows-based computers and servers. It had WinForms, a GUI library for desktop applications; ASP.NET, a framework for Web; and ADO.NET for data access. All these elements were driven by Common Language Runtime (CLR) to compile and execute managed code.

To unite various functions, .NET offered a framework class library (FCL) that included the base class library (BCL), network library, a numerics library, and others.

Since that time, the framework has undergone multiple iterations spanning runtime updates, new desktop graphical systems (WPF), APIs for service-oriented applications (WCF), and more.

The .NET CORE era

In 2014, Microsoft announced a dramatic shift in the way .NET exists by presenting .NET Core, a new cross-platform, cloud-friendly, and open-source version of the framework. .NET Core made it to a release in 2016, becoming the main technology to consider for new .NET projects. Gradually, Microsoft started porting existing services to work with Core. Some that didn’t receive official ports, like Windows Communication Foundation (WCF), were substituted by alternatives sourced from the community.

**ASP.NET extends .NET**

ASP.NET extends the .NET platform with tools and libraries specifically for building web apps.

These are some things that ASP.NET adds to the .NET platform:

Base framework for processing web requests in C# or F#

Web-page templating syntax, known as Razor, for building dynamic web pages using C#

Libraries for common web patterns, such as Model View Controller (MVC)

Authentication system that includes libraries, a database, and template pages for handling logins, including multi-factor authentication and external authentication with Google, Twitter, and more.

Editor extensions to provide syntax highlighting, code completion, and other functionality specifically for developing web pages

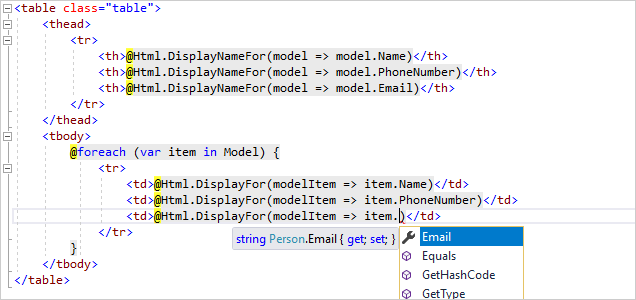
When using ASP.NET your back-end code, such as business logic and data access, is written using C#, F#, or Visual Basic.

Because ASP.NET extends .NET, you can use the large ecosystem of packages and libraries available to all .NET developers. You can also author your own libraries that are shared between any applications written on the .NET platform.

Dynamic pages using C#, HTML, CSS, and JavaScript

Razor provides a syntax for creating dynamic web pages using HTML and C#. Your C# code is evaluated on the server and the resulting HTML content is sent to the user.

Code that executes client-side is written in JavaScript. ASP.NET integrates with JavaScript frameworks and includes pre-configured templates for single page app (SPA) frameworks like React and Angular.



**What is ASP.NET Core?**

If you use ASP.NET, you'll soon come across the term ASP.NET Core.

ASP.NET Core is the open-source and cross-platform version of ASP.NET. You should use ASP.NET Core for all new applications. The tutorials on this site all use ASP.NET Core.

**Entities Framework**

Entity Framework was first released in 2008, Microsoft's primary means of interacting between .NET applications and relational databases. Entity Framework is an Object Relational Mapper (ORM) which is a type of tool that simplifies mapping between objects in your software to the tables and columns of a relational database.

Entity Framework (EF) is an open source ORM framework for ADO.NET which is a part of .NET Framework.

An ORM takes care of creating database connections and executing commands, as well as taking query results and automatically materializing those results as your application objects.

An ORM also helps to keep track of changes to those objects, and when instructed, it will also persist those changes back to the database for you.

## **Why Entity Framework?**

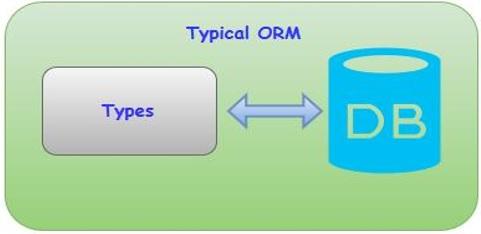
Entity Framework is an ORM and ORMs are aimed to increase the developer’s productivity by reducing the redundant task of persisting the data used in the applications.

Entity Framework can generate the necessary database commands for reading or writing data in the database and execute them for you.

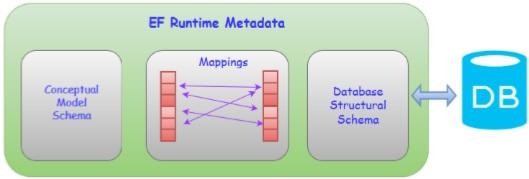
If you're querying, you can express your queries against your domain objects using LINQ to entities.

Entity Framework will execute the relevant query in the database and then materialize results into instances of your domain objects for you to work within your app.

There are other ORMs in the marketplace such as NHibernate and LLBLGen Pro. Most ORMs typically map domain types directly to the database schema.



Entity Framework has a more granular mapping layer so you can customize mappings, for example, by mapping the single entity to multiple database tables or even multiple entities to a single table.



Entity Framework is Microsoft's recommended data access technology for new applications.

 ADO.NET seems to refer directly to the technology for data sets and data tables.

Entity Framework is where all of the forward moving investment is being made, which has been the case for a number of years already.

Microsoft recommends that you use Entity Framework over ADO.NET or LINQ to SQL for all new development.

## **Conceptual Model**

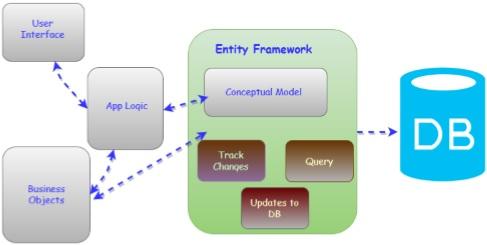
For developers who are used to database focused development, the biggest shift with Entity Framework is that it lets you focus on your business domain. What it is that you want your application to do without being limited by what the database is able to do?

With Entity Framework, the focal point is referred to as a conceptual model. It's a model of the objects in your application, not a model of the database you use to persist your application data.

Your conceptual model may happen to align with your database schema or it may be quite different.

You can use a Visual Designer to define your conceptual model, which can then generate the classes you will ultimately use in your application.

You can just define your classes and use a feature of Entity Framework called Code First. And then Entity Framework will comprehend the conceptual model.



Either way, Entity Framework works out how to move from your conceptual model to your database. So, you can query against your conceptual model objects and work directly with them.

## **Features**

Following are the basic features of Entity Framework. This list is created based on the most notable features and also from frequently asked questions about Entity Framework.

Entity Framework is a Microsoft tool.

Entity Framework is being developed as an Open Source product.

Entity Framework is no longer tied or dependent to the .NET release cycle.

Works with any relational database with valid Entity Framework provider.

SQL command generation from LINQ to Entities.

Entity Framework will create parameterized queries.

Tracks changes to in-memory objects.

Allows to insert, update and delete command generation.

Works with a visual model or with your own classes.

Entity Framework has stored Procedure Support.

**The architecture of Entity Framework, from the bottom up, consists of the following −**

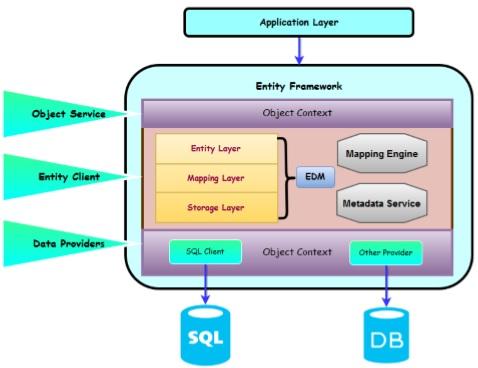
## **Data Providers**

These are source specific providers, which abstract the ADO.NET interfaces to connect to the database when programming against the conceptual schema.

It translates the common SQL languages such as LINQ via command tree to native SQL expression and executes it against the specific DBMS system.

## **Entity Client**

This layer exposes the entity layer to the upper layer. Entity client provides the ability for developers to work against entities in the form of rows and columns using entity SQL queries without the need to generate classes to represent conceptual schema. Entity Client shows the entity framework layers, which are the core functionality. These layers are called as Entity Data Model.



The **Storage Layer** contains the entire database schema in XML format.

The **Entity Layer** which is also an XML file defines the entities and relationships.

The **Mapping layer** is an XML file that maps the entities and relationships defined at conceptual layer with actual relationships and tables defined at logical layer.

The **Metadata services** which is also represented in Entity Client provides centralized API to access metadata stored Entity, Mapping and Storage layers.

## **Object Service**

Object Services layer is the Object Context, which represents the session of interaction between the applications and the data source.

The main use of the Object Context is to perform different operations like add, delete instances of entities and to save the changed state back to the database with the help of queries.

It is the ORM layer of Entity Framework, which represents the data result to the object instances of entities.

This services allow developer to use some of the rich ORM features like primary key mapping, change tracking, etc. by writing queries using LINQ and Entity SQL.

## **What’s New in Entity Framework 6?**

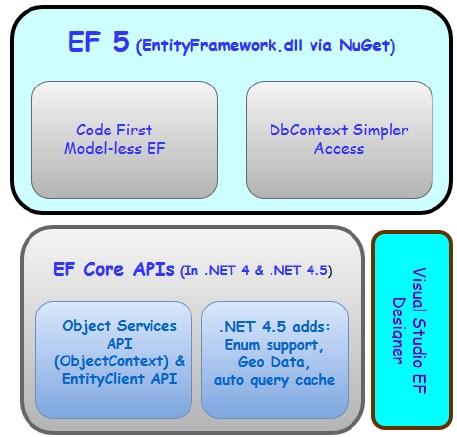
Framework has a complex API that lets you have granular control over everything from its modeling to its runtime behavior. Part of Entity Framework 5 lives inside of .NET. And another part of it lives inside of an additional assembly that's distributed using NuGet.

The core functionality of Entity Framework is built into the .NET Framework.

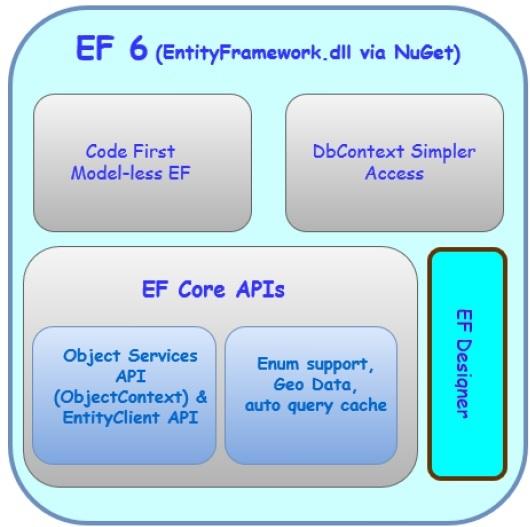
The Code First support, that's what lets Entity Framework use classes in lieu of a visual model, and a lighter way API for interacting with EF are in the NuGet package.

The core is what provides the querying, change tracking and all of the transformation from your queries to SQL queries as well as from data return into the objects.

You can use the EF 5 NuGet package with both .NET 4 and with .NET 4.5.One big point of confusion - .NET 4.5 added support for enums and spatial data to the core Entity Framework APIs, which means if you're using EF 5 with .NET 4, you won't get these new features. You'll only get them when combining EF5 with .NET 4.5.



Let us now take a look at Entity Framework 6. The core APIs which were inside of .NET in Entity Framework 6 are now a part of NuGet package.



It means −

All of the Entity Framework lives insides this assembly that's distributed by NuGet

You won't be dependent on .NET to provide specific features like the Entity Framework enum support and special data support.

You'll see that one of the features of EF6 is that it supports enums and spatial data for .NET 4

To start working on Entity Framework you need to install the following development tools −

Visual Studio 2013 or above

SQL Server 2012 or above

Entity Framework updates from NuGet Package

**8.       Advantages**

**Data entry in a simple way** – fast and very quickly data import, in a simple way it carried out once

**Speedily process** – in between making a sale and generating an invoice it reducing delays

**Reports automatically** - debtors and creditors, on profit and loss, customer accounts, inventory counts, forecasting, etc.

**Automating tasks** – such as calculating pay, producing pay slips, automatically calculating etc.

**Reducing the chance of errors** – by computerizing calculations that would be very low chance of errors

#### **Time Saving**

In accounting package the major thing and the most important advantage is that it saves a lot of time over manual bookkeeping. With a computerized accounting package, front end business documents and back end transaction recordings are completed simultaneously.

#### **Cash Flow Management:**

In accounting there is a great advantage that it allows you to record receivables and payable in every system. This allows the system to project your cash flow position into the future.

#### **Minimize Errors:**

When you use a computerized system, you make some mistakes in calculating totals or in typing, decreasing confusion. Invoice software usually allows you to include a tax percentage when appropriate, with the system performing all calculations. This is a useful advantage, especially when you have detailed invoices where hours or specific items are billed at different rates.

#### **Hybrid solutions:**

As technology improves, software vendors have been able to offer increasingly advanced software at lower prices. This software is suitable for companies at multiple stages of growth. Many of the features of mid-market and high-end software (including advanced customization and extremely salable databases) are required even by small businesses as they open multiple locations or grow in size.

#### **Computerized invoice:**

A major advantage of using a computerized invoice system is the capability to run reports to determine which customers have paid. An aging report shows you who owes you, how much and for how long. This is done easily and puts you in control of your finances.

#### **Organization:**

It is easy to lose or forget about invoices, especially when they are prepared manually. They can get lost in the mail, or they may be misspelled in a manual system. With an accounting software package, the data is there--all organized in one place to be reviewed, reprinted and resent in case of problems. You usually can look in the system for invoices based on names, amounts or invoice numbers. It is a fast and easy process because the software organizes data as it is entered.

#### **Information Accuracy:**

Organizations huge and little should keep exact records of their funds. In a few circumstances, there might be lawful implications for anything besides the most points by point and particular records of bookkeeping practices.

**9.       Disadvantages**

Not all businesses will benefit equally from using accounting software. If you’re thinking of switching from manual to computerized accounting, you should consider these

possible drawbacks: Cost: the bundle cost, albeit little in connection with your different expenses, is higher than a paper-based framework.

**Usage:** you will presumably require some underlying help setting up a records bundle. This will more often than not be a chargeable administration, maybe acquired from your bookkeeper or the framework supplier.

**Support:** you may need to buy yearly upkeep and backing for your bundle.

**SQL Server**

SQL Server is a database server by Microsoft.

The Microsoft relational database management system is a software product which primarily stores and retrieves data requested by other applications. These applications may run on the same or a different computer.

Going more in-depth, in order to understand what a SQL Server is, you must first understand what SQL is.

SQL is a special-purpose programming language designed to handle data in a relational database management system.

A database server is a computer program that provides database services to other programs or computers, as defined by the client-server model. Therefore, a SQL Server is a database server that implements the Structured Query Language (SQL).

SQL SERVER is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed and developed to compete with MySQL and Oracle database.

SQL Server supports ANSI SQL, which is the standard SQL (Structured Query Language) language. However, SQL Server comes with its own implementation of the SQL language, T-SQ

(Transact-SQL).

T-SQL is a Microsoft propriety Language known as Transact-SQL. It provides further capabilities of declaring variable, exception handling, stored procedure, etc.

SQL Server Management Studio (SSMS) is the main interface tool for SQL Server, and it supports both 32-bit and 64-bit environments.

## **Version History SQL Server**

* Microsoft and Sybase released version 1.0 in 1989.
* However, the partnership between these two ended in the early 1990s.
* Microsoft maintained ownership rights to the name SQL Server.
* Since the 1990s, subsequent versions of SQL Server have been released including SQL Server 2000, 2005, 2008, 2012, 2014, 2016, 2017, and 2019

## **SQL Server Editions**

****SQL Server Enterprise:****It is used in the high end, large scale and mission Critical business. It provides High-end security, Advanced Analytics, Machine Learning, etc.

****SQL Server Standard:****Itis suitable for Mid-Tier Application and Data marts. It includes basic reporting and analytics.

****SQL Server WEB:****It is designed for a low total-cost-of-ownership option for Web hosters. It provides scalability, affordability, and manageability capabilities for small to large scale Web properties.

****SQL Server Developer:****It is similar to an enterprise edition for the non-production environment. It is mainly used for build, test, and demo.

****SQL Server Express:****It is for small scale applications and free to use

## **Key Components and Services of SQL Server**

****Database Engine:****This component handle storage, Rapid transaction Processing, and Securing Data.

****SQL Server:****This service starts, stops, pauses, and continues an instance of Microsoft SQL Server. Executable name is sqlservr.exe.

****SQL Server Agent:****It performs the role of Task Scheduler. It can be triggered by any event or as per demand. Executable name is sqlagent.exe.

****SQL Server Browser:****This listens to the incoming request and connects to the desired SQL server instance. Executable name is sqlbrowser.exe.

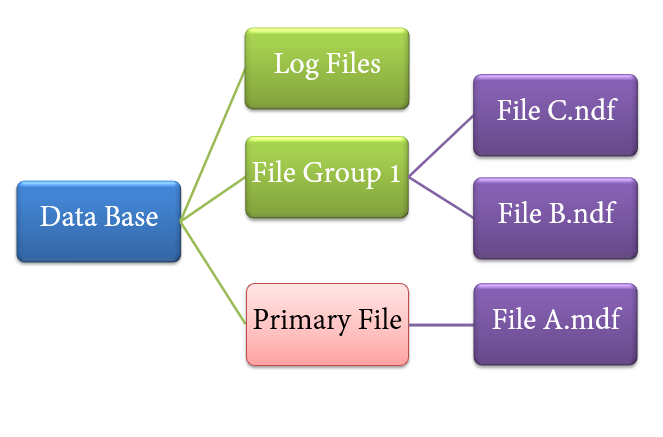
****SQL Server Full-Text Search:****This lets user running full-text queries against Character data in SQL Tables.Executable name is fdlauncher.exe.

****SQL Server VSS Writer:****This allows backup and restoration of data files when the SQL server is not running.Executable name is sqlwriter.exe.

****SQL Server Analysis Services (SSAS):****Provide Data analysis, Data mining and Machine Learning capabilities. SQL server is integrated with R and Python language for advanced analytics. Executable name is msmdsrv.exe.

****SQL Server Reporting Services (SSRS):****Provides reporting features and decision-making capabilities. It includes integration with Hadoop. Executable name is ReportingServicesService.exe

****SQL Server Integration Services (SSIS):****Provided Extract-Transform and Load capabilities of the different type of data from one source to another. It can be view as converting raw information into useful information. Executable name is MsDtsSrvr.exe



**10.      System Design**

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. System development is the process of creating or altering systems, along with the processes, practices, models, and methodologies used to develop them.

## **Elements of a System**

**Architecture**- This is the conceptual model that defines the structure, behavior and more views of a system. We can use flowcharts to represent and illustrate the architecture.

**Modules -**This are components that handle one specific tasks in a system. A combination of the modules make up the system.

**Components -**This provides a particular function or group of related functions. They are made up of modules.

**Interfaces -**This is the shared boundary across which the components of a the system exchange information and relate.

**Data -**This the management of the information and data flow.

A data flow diagram shows the way information flows through a process or system.

It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. You can use these diagrams to map out an existing system and make it better or to plan out a new system for implementation. Visualizing each element makes it easy to identify inefficiencies and produce the best possible system.

Physical and logical data flow diagrams

Logical data flow diagrams focus on what happens in a particular information flow:

what information is being transmitted, what entities are receiving that info, what general processes occur, etc.

The processes described in a logical DFD are business activities—a logical DFD doesn’t delve into the technical aspects of a process or system. Non-technical employees should be able to understand these diagrams.

Physical data flow diagrams focus on how things happen in an information flow.

These diagrams specify the software, hardware, files, and people involved in an information flow. A detailed physical data flow diagram can facilitate the development of the code needed to implement a data system.

* **Entity Symbols**

|  |  |
| --- | --- |
| Entity |  |
| Relationship |  |
| process |  |
| Data store |  |
| Data Flow |  |
| Attributes |  |
| Repeated Entity |  |

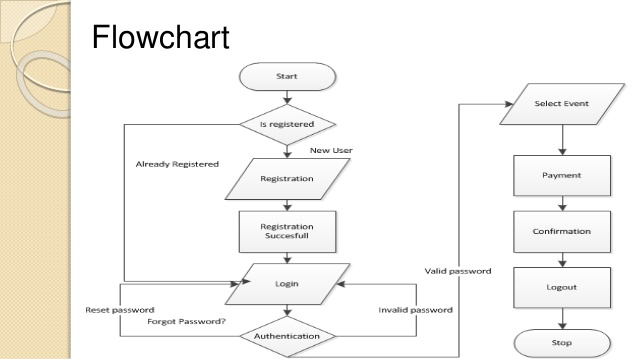
|  |  |  |
| --- | --- | --- |
| IMG_256 | Strong entity | These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity. |

|  |  |  |
| --- | --- | --- |
| IMG_256 | Weak entity | Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity. |

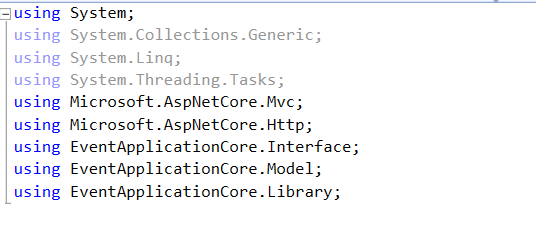
|  |  |  |
| --- | --- | --- |
| IMG_256 | Associative entity | Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances. |

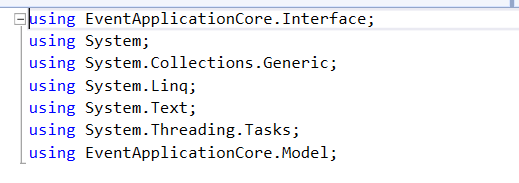
## Event Handler Architecture

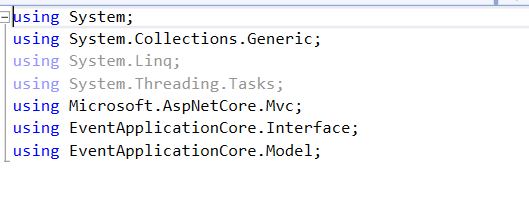
1. Flow Chart



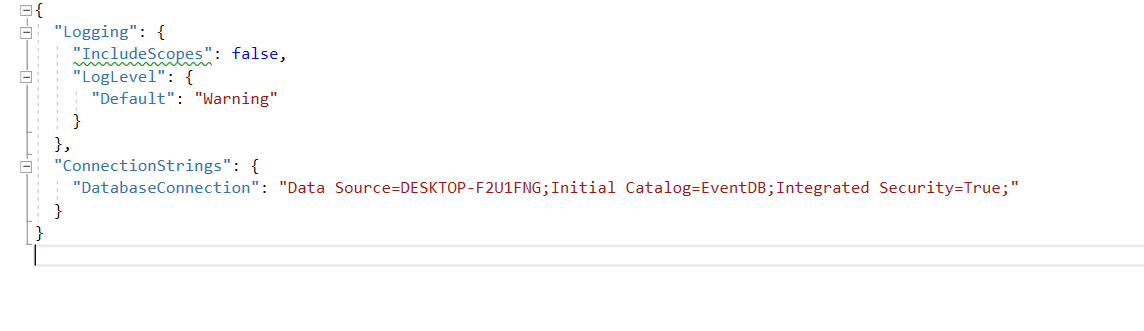
* **Imported Library**

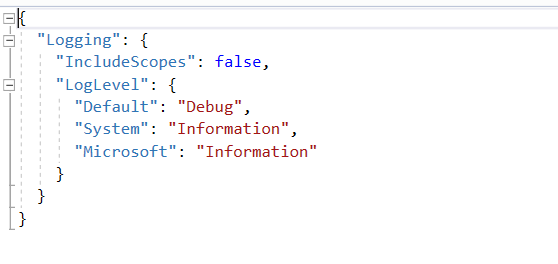






* **Applications Settings**



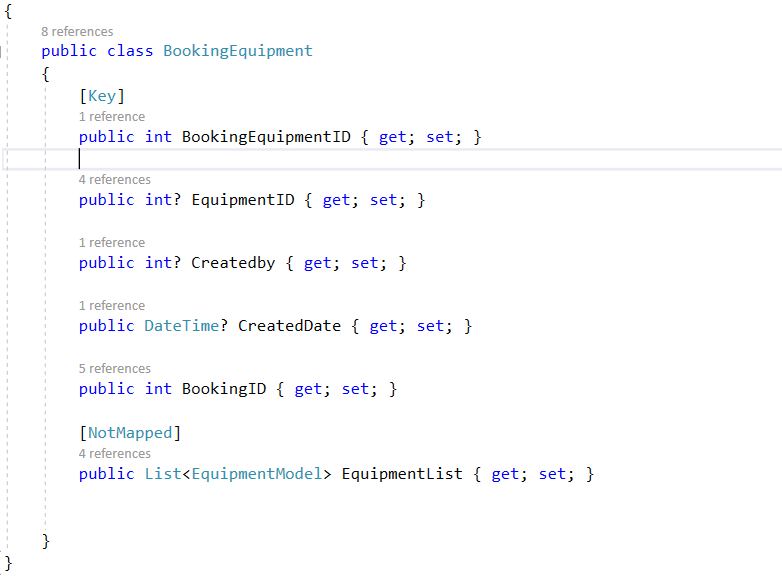


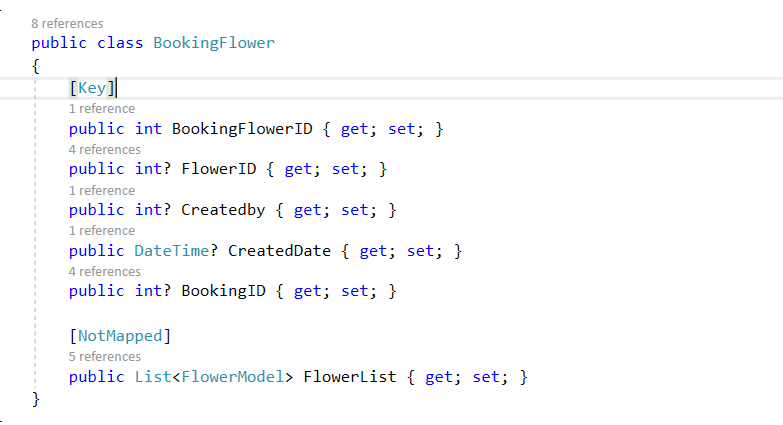
* **Controllers**





* **Entity Data Modal**

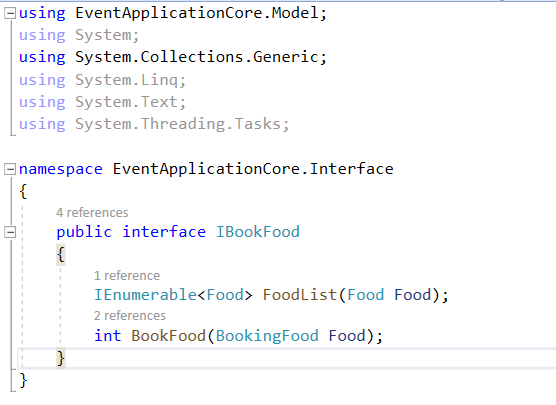


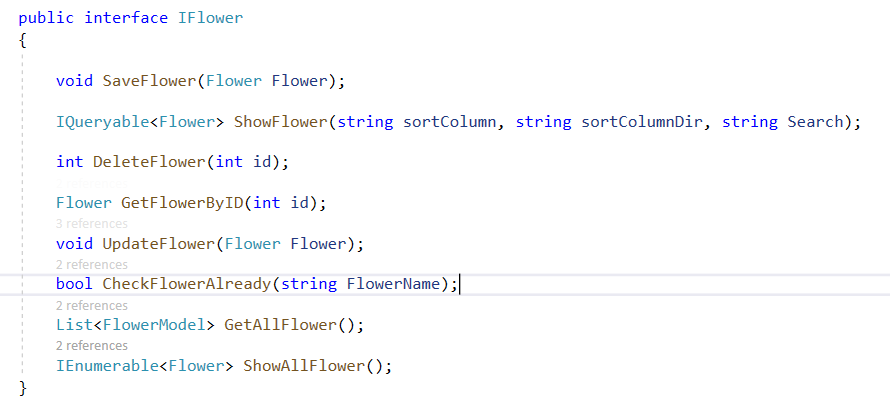


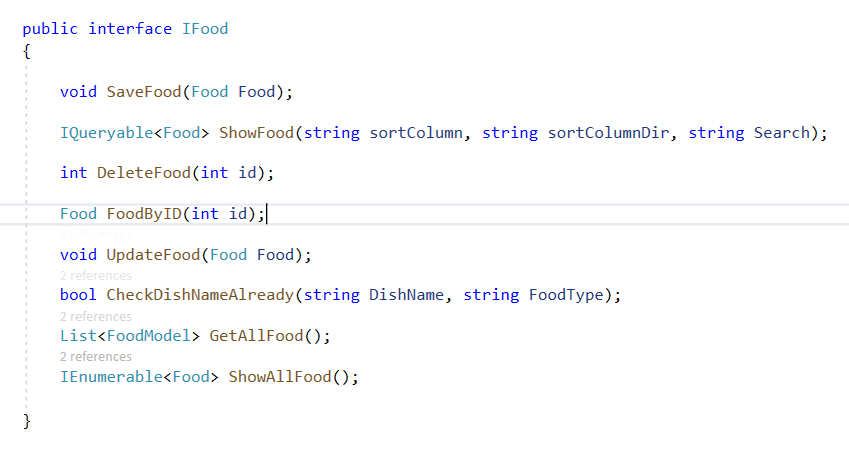


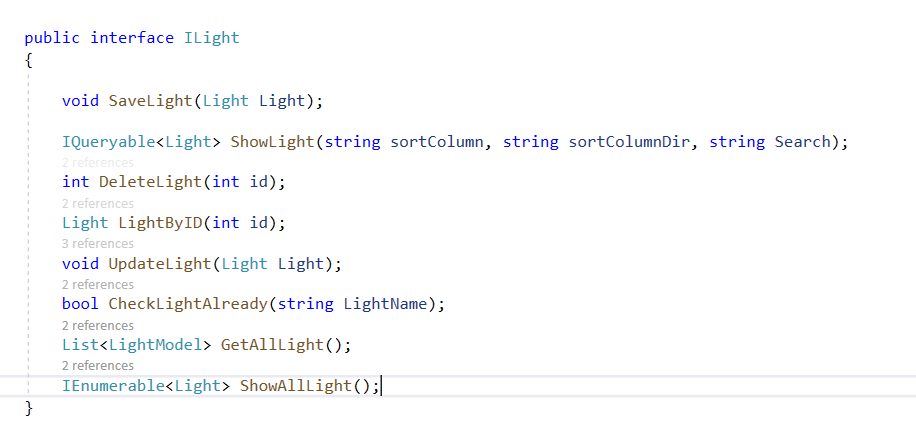


* **Interfaces**

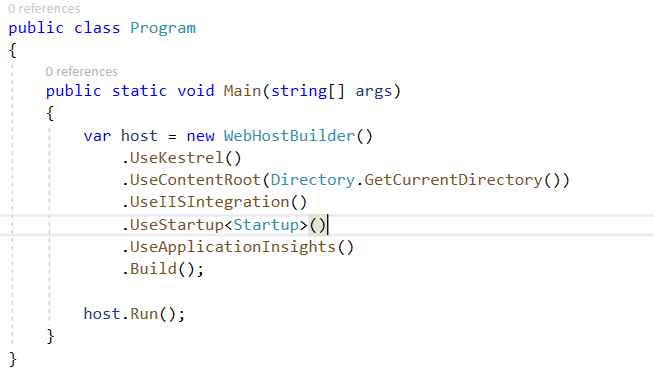








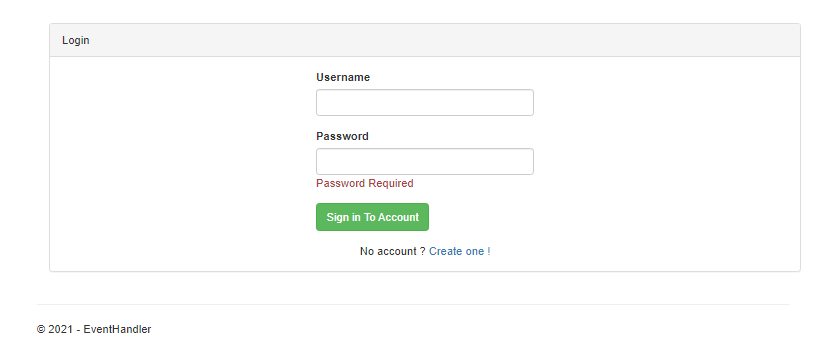
* **Program.cs**



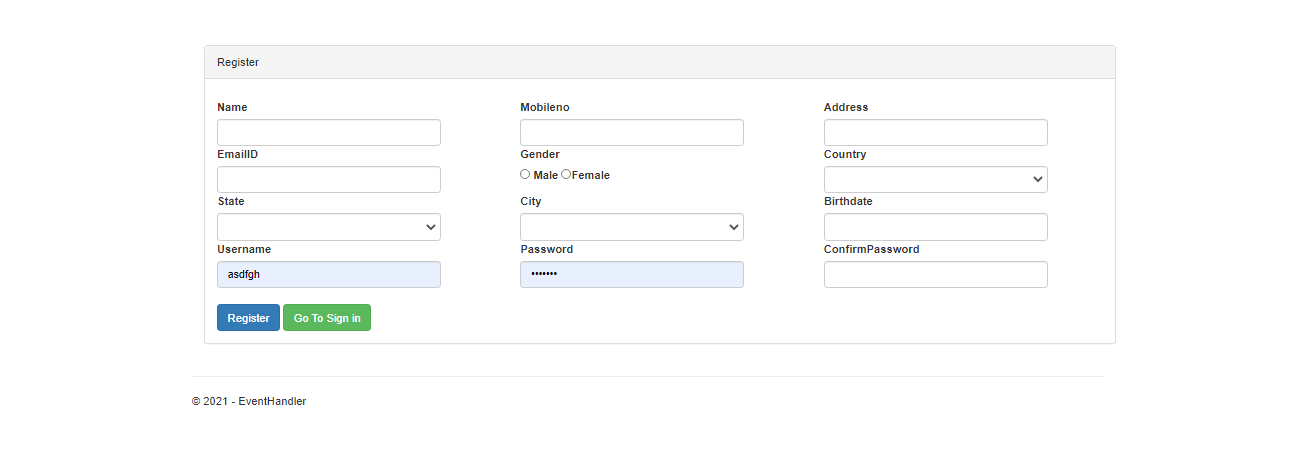


**Event Handler Front-end :-**

1. **Log in Page**

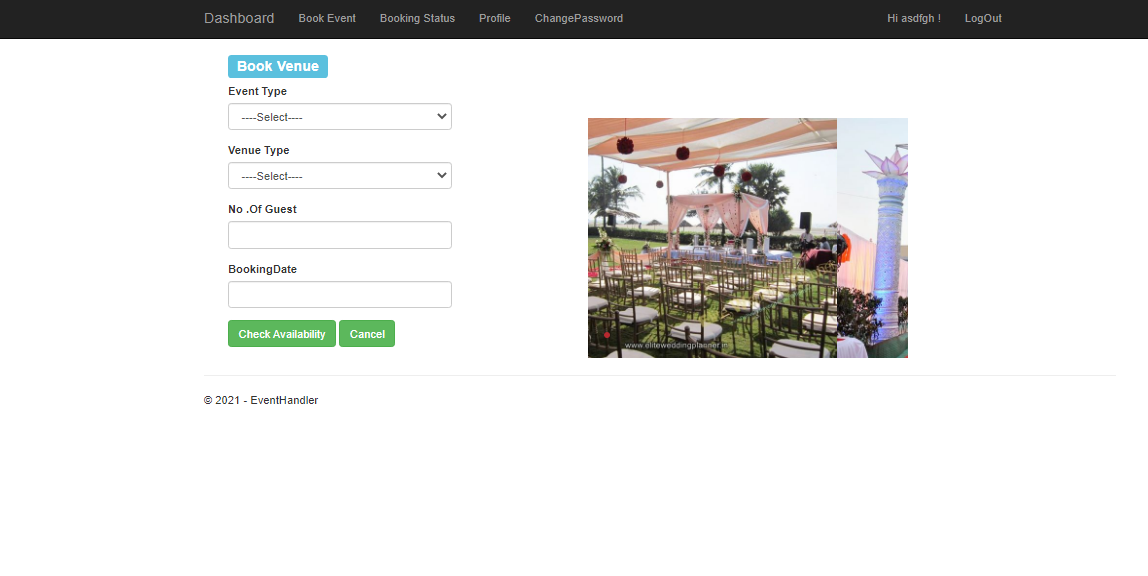


1. **Register Page**

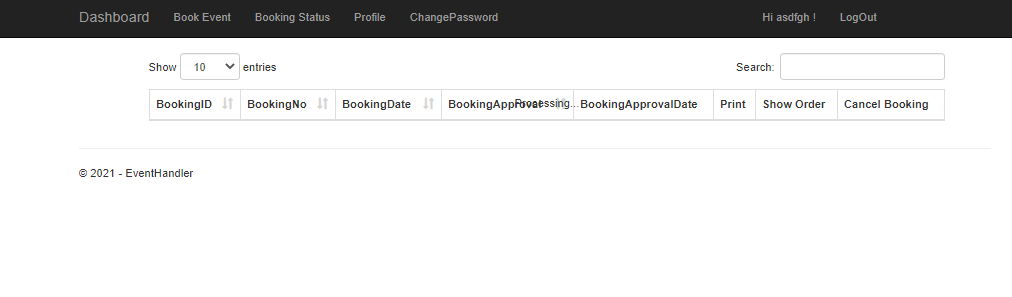


#### End user Interface

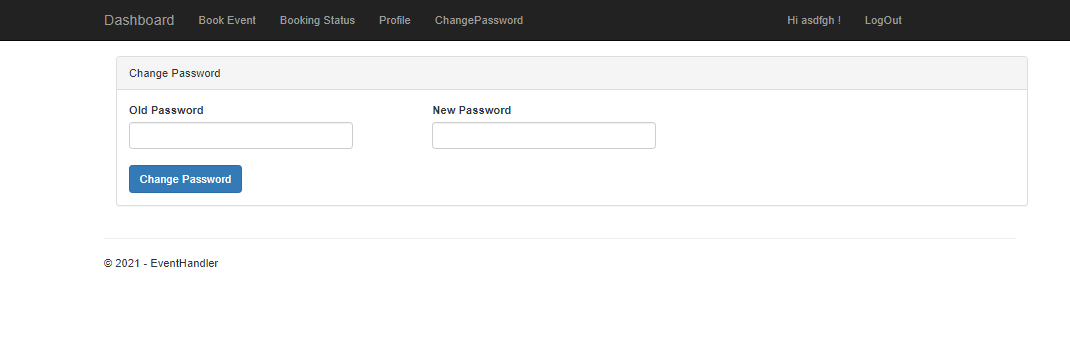
1. Event Booking Page



1. Booking status page

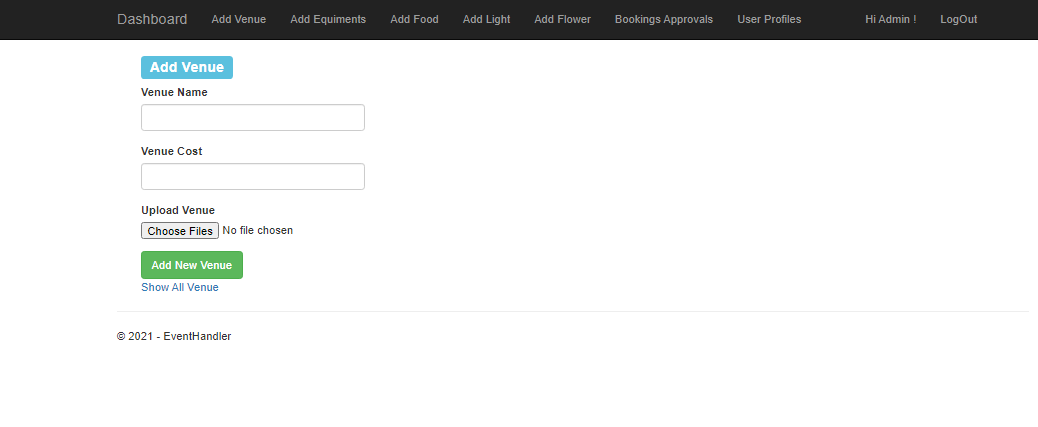


1. For changing Password

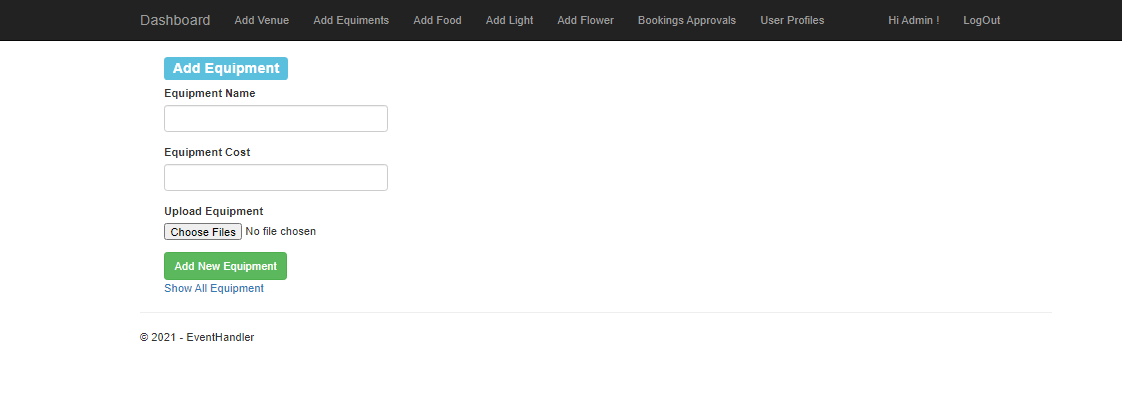


* Admin User Interface

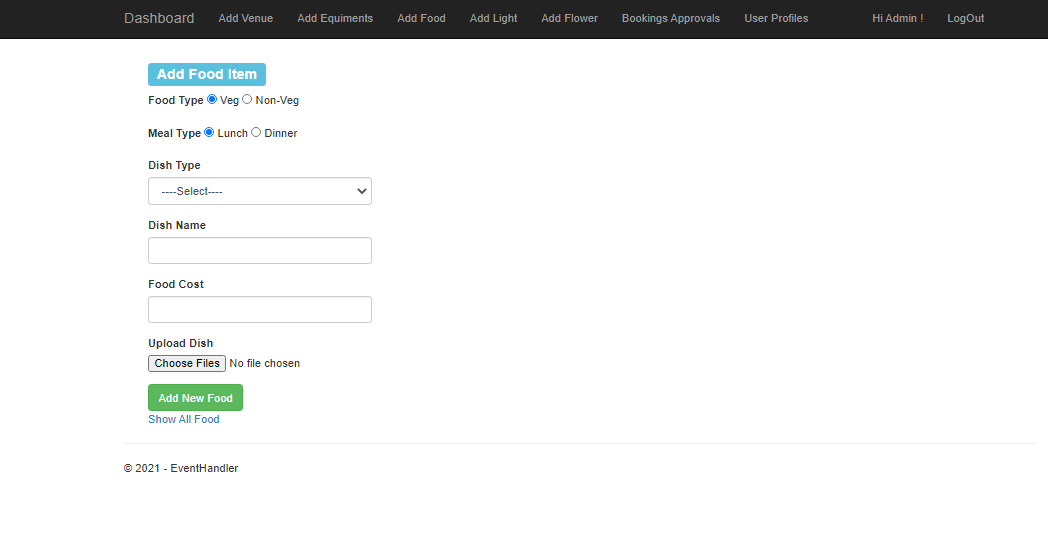
1. For adding new venue



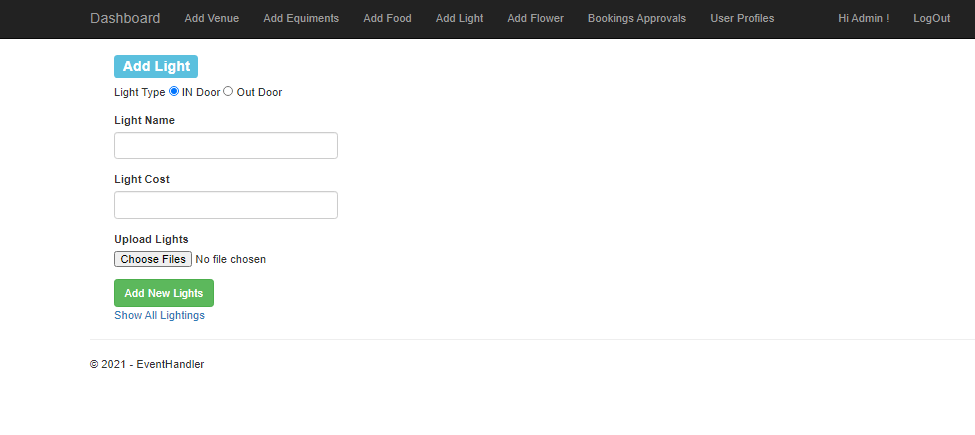
1. For adding Equipment



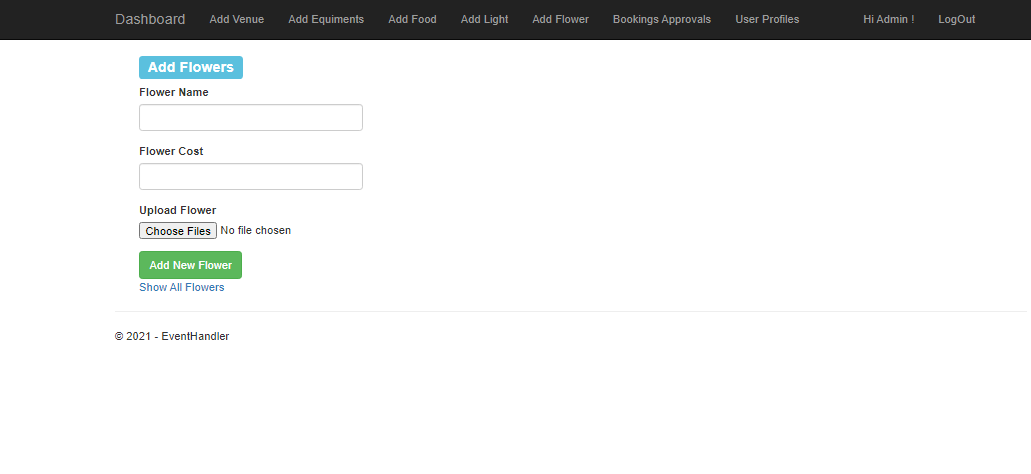
1. For adding Food Item



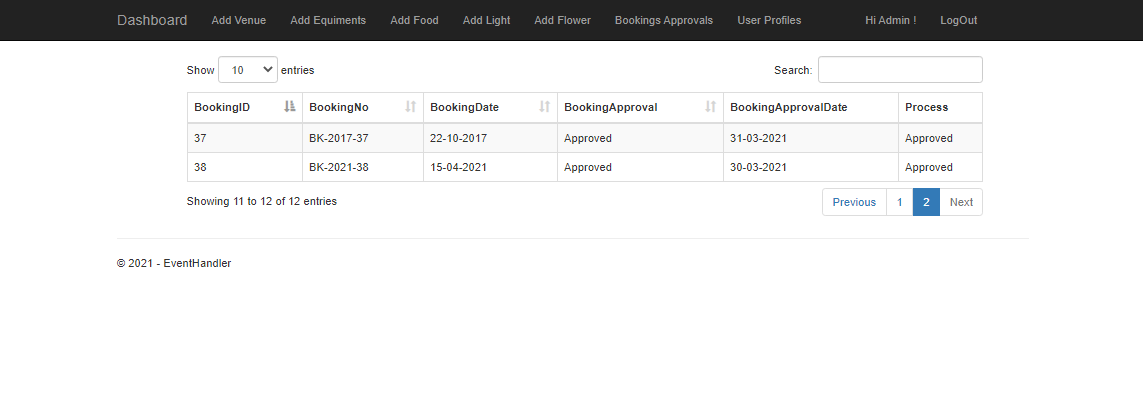
1. For adding new light



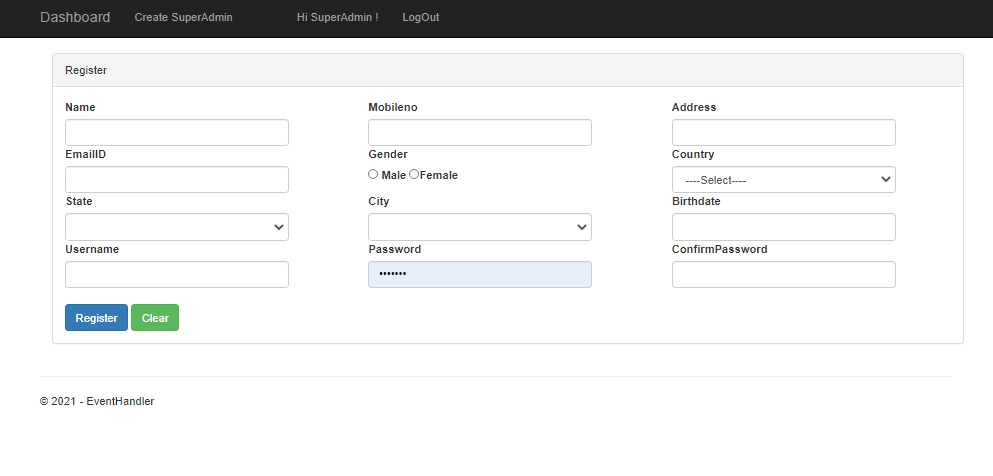
1. For adding new Flowers



1. Pending approval

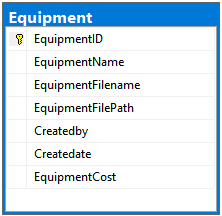


* Super Admin user interface

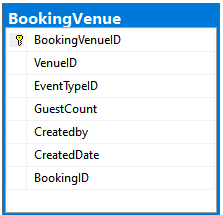


## Back- end Design

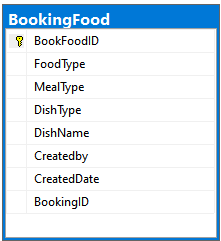
1. Equipment table



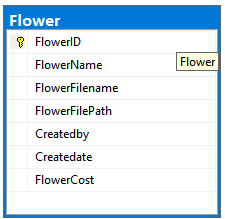
1. Booking Venue



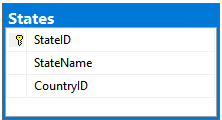
1. Booking Food



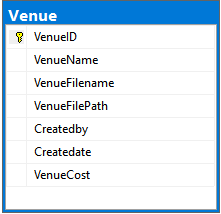
1. Flower



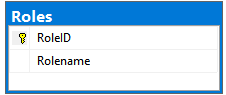
1. States



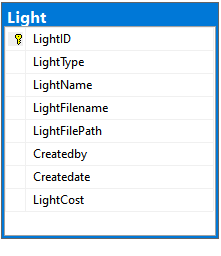
1. Venue



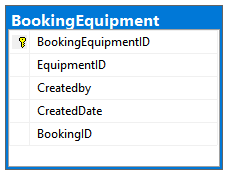
1. Roles



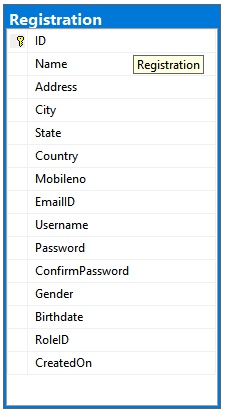
1. Light



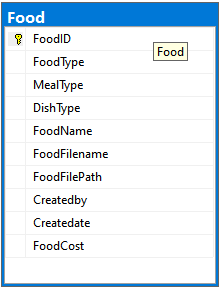
1. Booking Equipment



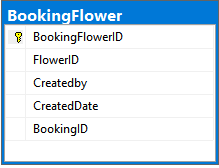
1. Registration



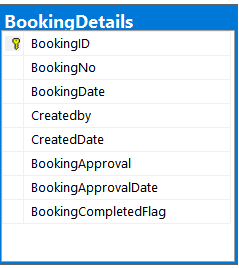
1. Food



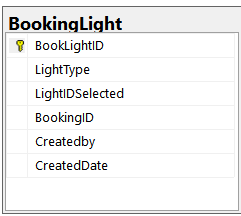
1. Booking Flower



1. Booking Details



1. Booking Light



# Conclusion

The study resulted in several interesting findings. The predominant reasons why enterprises switched from manual to computer based reporting of accounting information were as follows:

For automating clerical operations and for speedy, timely and more accurate reports to management.

Other reasons Qited were: Better more effective decisions aid, better and current financial information to control costs; better and current corrective action improving reaction time.

Better timeliness and more confidence for meeting schedules, More effective management action and planning; More planning and thinking time for managers;

Improved customer service; Management by exception and for reducing errors due to human failure.