

“WordMan”

*Dissertation submitted in partial fulfillment of the requirement for the degree
of*

BACHELOR OF TECHNOLOGY

IN

Computer Science and Engineering

By

Muskan (151369)

Anirudh Sharma (151363)

Vasudha Pandey (151404)

Himanshu Chauhan (151414)

Ayushmat Bhardwaj Soni (151420)

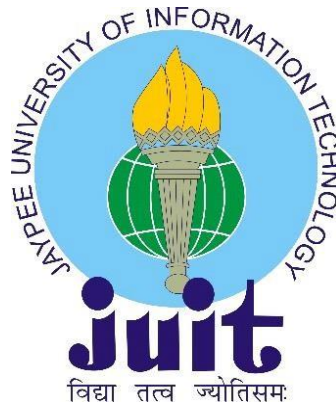
Prateek Ramani (151464)

UNDER THE GUIDANCE OF

Mr. Dillibabu R.

System Engineer - Education, Training & Assessment, ETA, Infosys Mysore

To



Department of Computer Science & Engineering and Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, SOLAN

CERTIFICATE

I hereby declare that the work presented in this report entitled “**WordMan**” submitted in the department of Computer Science & Engineering and Information Technology, **Jaypee University of Information Technology, Wagnaghat** is an authentic record of our work carried out under the supervision of **Mr. Dillibabu R.**, System Engineer - Education, Training & Assessment, ETA, Infosys Mysore. We have not submitted this work elsewhere for any degree or diploma.

Muskan (151369)
Anirudh Sharma (151363)
Vasudha Pandey (151404)
Himanshu Chauhan (151414)
Ayushmat Bhardwaj Soni (151420)
Prateek Ramani (151464)

Department of Computer Science & Engineering
Jaypee University of Information Technology, Wagnaghat, Solan, India

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Dillibabu R.
System Engineer - Education, Training & Assessment, ETA Infosys
Mysore

Date: May, 2019

ACKNOWLEDGEMENT

We are highly indebted to all the members of **ETA department of Infosys, Mysore** for their guidance and constant supervision as well as providing necessary information regarding the project and also for their support in completing the project.

We would like to express our gratitude towards **Mr. Dillibabu R.**, System Engineer, ETA, for his kind co-operation and encouragement which helped us in completion of this project and for giving us such attention and time.

Our thanks and appreciation also goes to the faculty of Computer Science Department of **Jaypee University of Information Technology, Waknaghat** for their constant support and motivation.

CONTENTS

Chapter 1: Introduction

1.1 Introduction.....	1
1.2 Problem Statement.....	2
1.3 Objective.....	3
1.4 Organization.....	4
1.5 Methodology.....	5

Chapter 2: Literature Survey

2.1 Entity Data Models.....	10
2.2 HTML.....	12
2.3 Cascading Style Sheet(CSS).....	12
2.4 Java Script.....	12
2.5 Asp.NET MVC.....	13
2.6 SQL Server.....	14
2.7 MySQL Features.....	14

Chapter 3: System Development

3.1 Hardware Interface.....	17
3.2 Software Interface.....	17
3.2.1 Functional Requirements.....	17
3.2.2 Non-Functional Requirements.....	18
3.2.3 Performance Requirements.....	18
3.2.4 Other Requirements.....	18

3.3 Project Development.....	19
3.3.1 Implementation of Database.....	19
3.3.2 MVC Implementation.....	31
3.3.2.1 Controllers.....	31
3.3.2.2 Views.....	35

Chapter 4: Performance Analysis

4.1 Test Environment.....	51
4.2 Unit Test Cases.....	51

Chapter 5: Conclusion

5.1 Conclusion.....	57
5.2 Future Scope.....	57

LIST OF FIGURES

S. No.	TITLE
1	Use Case of Guest User
2	Use Case of Registered User
3	Use Case of Admin
4	Components Of MVC
5	Class Diagram
6	Home Page 1
7	Home Page 2
8	Home Page 3
9	About Us
10	Contact Us 1
11	Contact Us 2
12	Feedback

13	Login Page Of User
14	Home Page Of User
15	Messages of Category Birthday
16	Messages of Category Farewell
17	Messages of Category Joining Anniversary

18	Edit Details
19	Change password Pop Up
20	Edit Message
21	Top Rated Messages
22	Sort By downloads
23	View Message By Template 1
24	View Message By Template 2
25	View Message By Template 3
26	Home Page Of Admin
27	Add Category
28	Add Message
29	View Categories
30	Register New User
31	Payment Portal Pop Up
32	Error Page 1
33	Error Page 2
34	Error Page 3

LIST OF TABLES

S. No.	TITLE
1	Users
2	Roles
3	Card Details
4	Message
5	Feedback
6	Categories
7	Rating
8	Template
9	Test Data 1
10	Test Data 2
11	Test Data 3
12	Test Data 4
13	Test Data 5

ABSTRACT

WordMan is focused on giving you the most ideal computerized welcome and wish your friends and family joy and love. In the present progressively rushed occasions, we get so fascinated in bringing home the bacon that we nearly neglect to welcome the genuine favors in our lives - our kin. Regardless of whether we talk about our mothers who courageous innumerable restless evenings after we are conceived or that closest companion who is dependably close by through various challenges, it is sheltered to state that our reality wouldn't be the equivalent if not for our friends and family. Right now is an ideal opportunity to tell them how you feel, without saying a word, through a scope of energizing web based welcome thoughts set up together by our group of specialists. Our group of specialists comprehend the difference of human feelings and spend innumerable hours in assembling keen messages that help you express your sentiments to your friends and family. Our website really trust that each human relationship is unique, with each individual in our life holding a fundamental spot that requires a novel gratefulness, and WordMan is here to get that going. Joy your family and companions with these new and fun online welcome.

Chapter 1: INTRODUCTION

1.1 INTRODUCTION

We all seem to be very curious about the occasions that we come across our daily life, excited about wishing our near and dear ones sometimes it is difficult to figure out the messages we need to send to them. For the sake of the same we have made a website which makes the user access different messages on different occasions to different relations, with use of their desired templates.

In this project we have used some concepts of .net a language generated by Microsoft. Using those concepts, we have developed a website for end user that provides with the appropriate greetings and a feature to download those greeting with templates of choice. Our project included the concepts of c#, Entity framework, MVC and WEB API.

We have made our searching algorithm using syntax and concept of C# language, entity framework concepts were used for making connections with the database and the UI was developed using MVC. We started with the entity framework of our website for which we used Database first approach that was done using scaffolding command, then performed different CRUD operations using the.

LINQ queries which are used to fetch data from the database. CRUD operations includes Create, read, update and delete queries, we used java script to set templates of our websites and MVC concepts. For homepage and other web pages. Our website makes all the search works easy for the end user without getting into much details. We have covered almost every keyword which may be used by the user for searching greetings for the specific occasions in our searching algorithm using C# as the coding language.

All the expectations of the user are kept in mind, it is a basic yet effective project that helps to spend a lot of time on for the sending greetings purpose, the search engine is made kept in mind all the keywords that can be used when the user searches for greetings, almost all the relations are kept in mind by making the algorithm and all the permutation a user can try.

As it is said nothing is perfect, we have tried to tend to perfections but there can be things that are not ensured in our project, already said it is a basic project. We have tried to make our UI as effective as we could be using all the concepts that we have learnt in our Internship. No concept that we have learnt is left, all the concepts are applied keeping in mind the effectiveness.

We have made a lot of efforts on each and every part whether its database or the algorithm or the front end or all the connections. Search any relation with the occasion and you will get the output, some exceptions may be there if the search statement is not in accordance to the algorithm made by us but we have kept almost everything in mind. This site is accessible 24/7 wherever and whenever you want to greet anyone, you can without a second thought access our website.

The way you want your template look, the way you want your message to be, we have thought to every extent and inserted all those messages in our database, all the main occasions are also kept in mind by us and are inserted in the database, we also have a paid download option if the user wants to download the message. You just have to be a registered user for that and should be logged in during the session.

1.2 PROBLEM STATEMENT

A web application which takes inputs from user and provides a required statement/paragraph. For example, if the user types input as 'This is my friend's birthday, suggest something', the output can be, 'Dear User, and a statement or paragraph appears which the user can use as a

birthday message or greetings to share with his friend. Not only for birthday, can it be used for all occasions. This app will include strong data algorithms.

We have used our knowledge about the things we mentioned in introduction for making this project as it clearly mentions that it takes user inputs on greetings for different occasions to different relationships and names and displays a message with desired templates. We made different user stories with respect to admin and user that is been implemented in the project we have made a very strong search algorithm that detects different occasions, relations using C#, we made different views that can be accessed by the user to sign up, login, payment, view message, download using java script and html.

We specified different actions related to respective buttons that maybe clicked by the user, most importantly we have maintained a proper database using Microsoft SQL Server as a tool that is used to accessed categories, messages by the user not only this but we have maintained feedback page and user is allowed to give ratings, see the messages according to the ratings purchase messages using download button and select the desired template for the messages, these all features the card details everything has been updated in a proper data base

1.3 OBJECTIVE

Our main objective is to make greetings for the user more easily accessible, anytime and anywhere. The user just has to search and he/she will get all the necessary details, also making our website 24/7 accessible with an UI pleasing to eyes of the users, so he can easily use all the functionalities we have provided such as register, login, views, payments just on a single click and typing some credentials.

The objective is also to provide user with all the necessary contents that is needed in the greetings, without him worrying about anything, to provide him/her with top rated and

downloaded messages for keeping transparency of our website, these are the main goals that we have concentrated on and making an easy and secure payment portal, just adding some important not all card credentials. We have looked on every mirror details of the project leading it to achieve the goal we thought of. With the vision of completion in the desired time limit and making on time delivery to the client (our mentor).

We have a User, admin and a Guest that can access the website in different ways. Admin performs different functionalities, User can access all the functionalities that can help in getting the greeting once by logging in and making payments, guest can only view some of the messages that we have allowed them to for accessing the site completely they have to be a User.

1.4 ORGANIZATION

Chapter1 contains the Introduction of our project, all the necessary details with the concepts used, the description of project, the objective we kept in mind while commencing the project, the methodology used by us.

Chapter 2 contains the literature survey in which we will define all the concepts that we have gone through and used in our project, including all the details of the concept with a brief description.

Chapter 3 contains the System development which contains the details about the making of database, models, views, controllers, framework, and all the things we have included in our website, these contents are well explained and told.

Chapter 4 contains the agile part, as we have used agile model in our project we have giving all our agile details like in the way we started, the Scrum master, time division of our project.

Chapter 5 contains all the future scope of the project and the conclusion of our project, and all the segregation of our project detailing. The way how we all our project and which new technologies we can use in future.

1.5 METHODOLOGY

We have three modules of our project, the admin, the guest user, the registered user, in this section those will be discussed one by one with the diagram with explaining the functionalities. Each use case is defined in the diagram prominently with the explanation.

Guest User: The following figure shows the functionalities user can if he/she is a guest and is not logged in. As the guest user cannot view as much as the registered user.

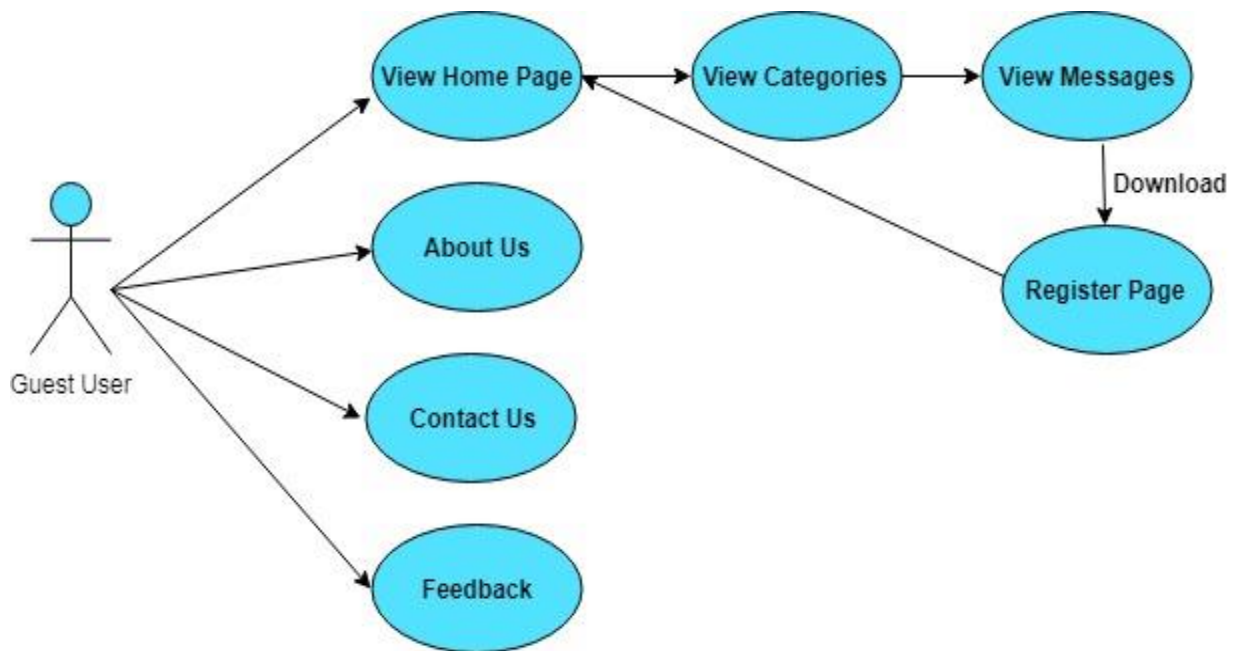


Figure 1: Use case Guest User

Once guest user accesses the website he is directed to the homepage with contains about us, Contact Us, Feedback actions. Once the guest is at the homepage he can view all the categories

of the occasions defined in the database such as birthday, Diwali etc. This category options helps him to redirect to the occasion he wishes to greet a person on. Then he can view the messages which he is allowed to and if he/she clicks on the download button he /she is redirected to the Register Page.

Homepage: Contains all the buttons that redirect the user to pages such as about us, contact us and the feedback. It also has a view category option which provides user with all the occasions that have been saved in our database.

About us: Gives the knowledge about what kind of services our website provides to the user and the purpose of making this site.

Contact Us: Gives all the details about how to and whom to approach regarding the quires about our website.

Feedback: It gives user to give the feedback related to website as how much he liked the service provided by us.

View messages: Helps to see all the messages related to the category and download the same but this download feature is not available for the guest user, only accessible once user becomes registered.

Register page: If the guest user will click on the download button after viewing the message he will be redirected to this page and he has to enter the desired credential to become a register user and avail the download feature.

Registered User: Our site allows more functionality usage once the user is registered, the diagram for the same is shown below

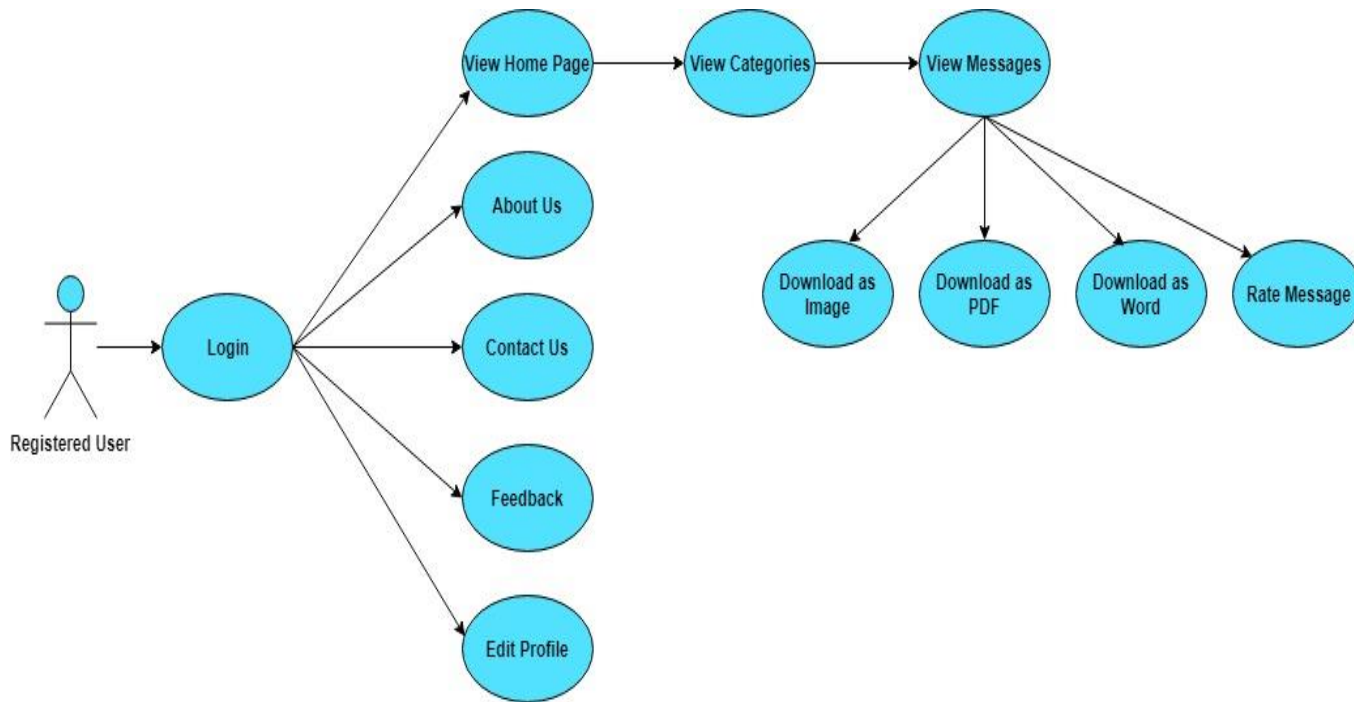


Figure 2: Use case registered user

Login: As the user is registered he/she will be directed to the login where he/she will be putting the required credentials for redirecting to the homepage.

View Homepage: the homepage will contain different actions like about us, contact us, feedback with the addition feature of edit profile provided to the registered user.

Edit profile: Used to view the profile and can change username or password if required by the user. He will be able to see all his necessary details on this page.

View categories: Used to view the occasion Category list.

View messages: Under categories we will have some messages which can be previewed by the register user. He can select the desired template according to his wish. After this if he wants to download he is provided with different options.

He can download as a PDF, Word and Image according to his desired requirement.

Rate Message: He is provided with an additional feature to rate message once he use the message so that we can display the top rated messages according to how user rates the message.

Admin: He is the one who can alter some things of the website and view the user details, ratings, feedback, he also has the excess to insert into database.

Login as Admin: We have a separate option for a person to login as admin as the functionalities of admin totally differ from that of user. So the login portal of admin is different.

View homepage: Homepage of admin contains Edit about us, edit contact us and view feedbacks. Alteration can be done by the admin.

Edit contact us: If admin wants to change the contact numbers or any contact details he is free to do that once he logs in with the correct credentials.

Edit about us: The purpose and services provided by the website can be altered by the admin according to the wish and time to time if the features get updated.

View Feedbacks: The admin can view all the feedback given by the user and make changes in the site accordingly.

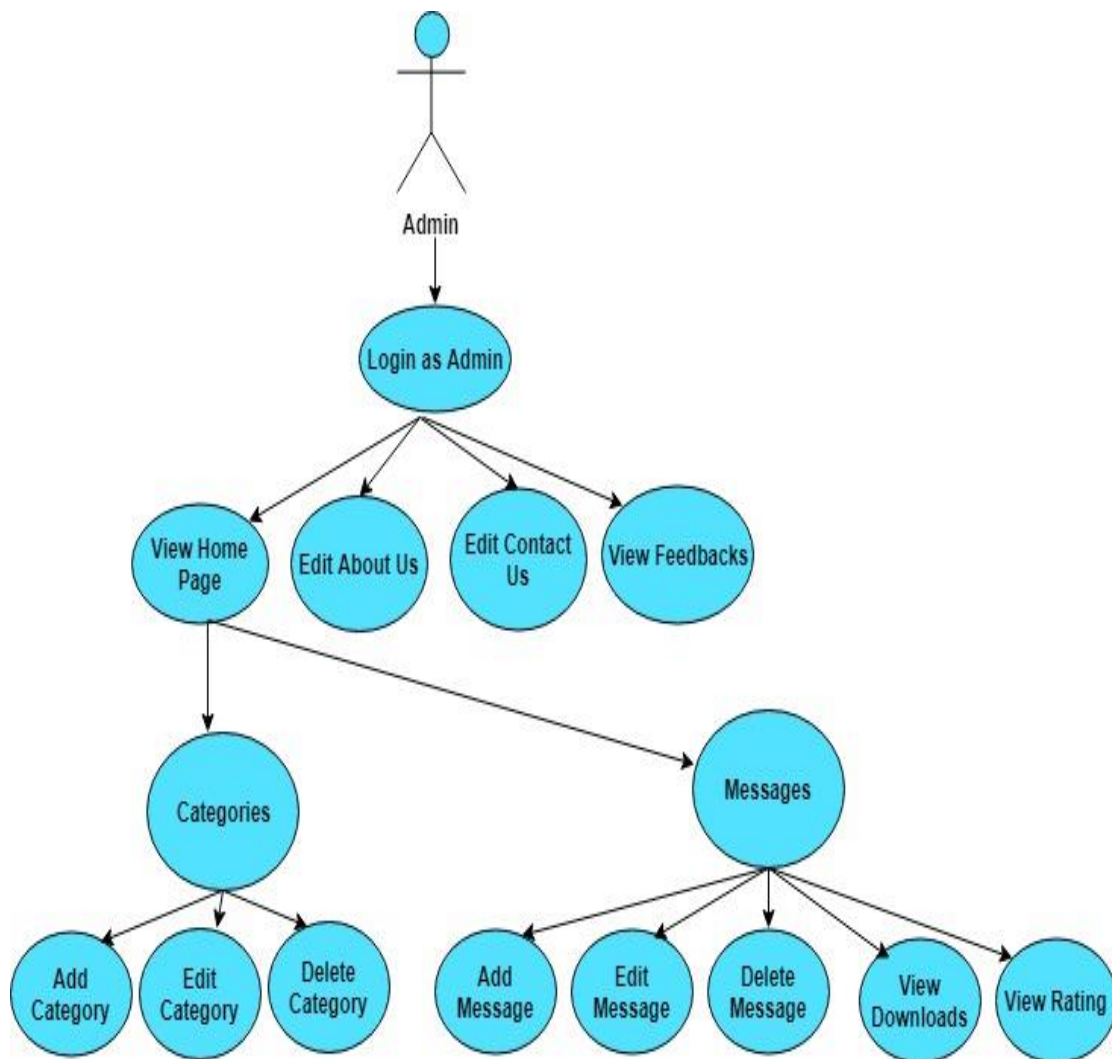


Figure 3: Use Case Admin

Categories: Include modifications of different categories by performing the CRUD operations admin can:

-Add a category

-Edit a Category

-Delete a Category

Messages: Admin can modify the messages table of the database also by performing different CRUD operations

-Add a Message

-Delete a Message

-Edit a Message

View Downloads: He can view the number of downloads updated recently to a particular message.

View Ratings: He can view the ratings given to a particular message.

Chapter 2: LITERATURE SURVEY

2.1 ENTITY DATA MODELS

Entities that is C# classes that represents our database. Data model that tells relationship between the different entities. The Scaffold command runs in the Package Manager Console which. There are some important parameters in the same command those are:

-Connection string - It is used to establish a connection with the database.

-Provider string - It is database provider name that has all the run-time libraries required to interact with a specific database.

LINQ Quires: **LINQ** is a short form for **Language Integrated Query** which was added by Microsoft in .NET Framework 3.5. This is a programming model for data access that integrates query support across .NET Languages.

It is introduced to reduce the amount of complexity and improving performance.

This can be used for querying in different data sources that are:

- Plain text
- XML (extensible Markup Language)
- Relational Database
- Datasets
- Business class objects

The two different syntax for writing queries is:

- Query syntax –This is similar to SQL queries
- Method syntax – This is equivalent to C# methods

The project contains different web pages and buttons like sign up, about, contact, login, logout all are made using views in different actions in controller class. Each action returns a respective view that takes us to the page we want and we have different mapping routes each according to the action performed by the controller and one default route. We have established all these pages using MVC and its controller classes.

Model-View-Controller - an architectural design pattern used for enterprise application development. It separates different aspects of the applications (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

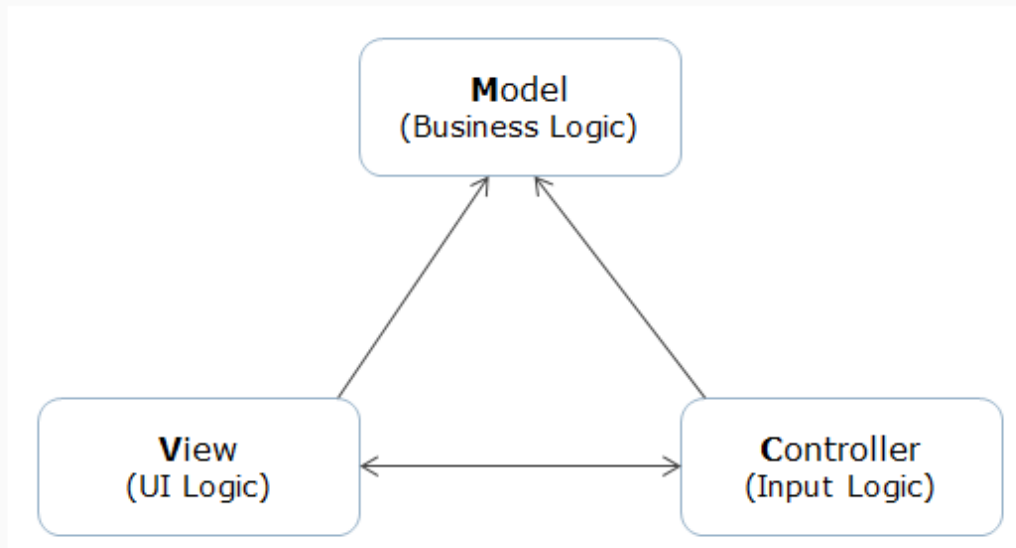


Figure 4: Components of MVC

This separation provides easier change management and high testability of the application.

2.2 HTML

HTML, which stands for Hypertext Markup Language, is the predominant markup language for web. HTML is not a programming language; it is a markup language. A markup language is a set of markup tags. HTML uses markup tags to describe web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc. as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can include or can load scripts in languages such as JavaScript which affect the behavior of HTML processors like Web browsers to define the appearance and layout of text and other material.

2.3 CSS

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

2.4 Java Script

JavaScript is not a programming language in strict sense. Instead, it is a scripting language because it uses the browser to do the dirty work. If you command an image to be replaced by another one, JavaScript tells the browser to go do it. Because the browser actually does

the work, you only need to pull some strings by writing some relatively easy lines of code. That's what makes JavaScript an easy language to start with.

But don't be fooled by some beginner's luck: JavaScript can be pretty difficult, too. First of all, despite its simple appearance it is a full-fledged programming language: it is possible to write quite complex programs in JavaScript. This is rarely necessary when dealing with web pages, but it is possible. This means that there are some complex programming structures that you'll only understand after protracted studies.

Secondly, and more importantly, there are the browser differences. Though modern web browsers all support JavaScript, there is no sacred law that says they should support exactly the same JavaScript. A large part of this site is devoted to exploring and explaining these browser differences and finding ways to cope with them.

So basic JavaScript is easy to learn, but when you start writing advanced scripts browser differences (and occasionally syntactic problems) will creep up.

2.5 Asp.NET MVC

The ASP.NET MVC is a web application framework developed by Microsoft, which implements the model-view-controller (MVC) pattern. It is open-source software, apart from the ASP.NET Web Forms component which is proprietary.

In the later versions of ASP.NET, ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages) will merge into a unified MVC 6. The project was initially called 'ASP.NET next' and was later renamed to ASP.NET Core.

Based on ASP.NET, ASP.NET MVC allows software developers to build a web application as a composition of three roles: Model, View and Controller. The MVC model defines web applications with 3 logic layers:

Model (business layer)

View (display layer)

Controller (input control)

A model represents the state of a particular aspect of the application. A controller handles interactions and updates the model to reflect a change in state of the application, and then passes information to the view. A view accepts necessary information from the controller and renders a user interface to display that information.

"ASP.NET MVC framework is a lightweight, highly testable presentation framework that is integrated with existing ASP.NET features. Some of these integrated features are master pages and membership-based authentication. The MVC framework is defined in the System.Web.Mvc assembly."

The ASP.NET MVC framework couples the models, views, and controllers using interface-based contracts, thereby allowing each component to be tested independently.

2.6 SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

2.7 MySQL Features:

1. Always On Availability Groups -- This feature takes database mirroring to a whole new level. With Always On, users will be able to fail over multiple databases in groups instead of individually. Also, secondary copies will be readable, and can be used for database backups. The big win is that your DR environment no longer needs to sit idle.

2. Column store Indexes -- This a cool new feature that is completely unique to SQL Server. They are special type of read-only index designed to be use with Data Warehouse queries. Basically, data is grouped and stored in a flat, compressed column index, greatly reducing I/O and memory utilization on large queries.
3. User-Defined Server Roles -- DBAs have always had the ability to create custom database role, but never server wide. For example, if the DBA wanted to give a development team read/write access to every database on a shared server, traditionally the only ways to do it were either manually, or using undocumented procedures. Neither of which were good solutions. Now, the DBA can create a role, which has read/write access on every DB on the server, or any other custom server wide role.
4. Enhanced Auditing Features -- Audit is now available in all editions of SQL Server. Additionally, users can define custom audit specifications to write custom events into the audit log. New filtering features give greater flexibility in choosing which events to write to the log.
5. BI Semantic Model -- This is replacing the Analysis Services Unified Dimensional Model (or cubes most people referred to them). It's a hybrid model that allows one data model will support all BI experiences in SQL Server. Additionally, this will allow for some really neat text infographics
6. Sequence Objects -- For those folks who have worked with Oracle, this has been a long requested feature. A sequence is just an object that is a counter -- a good example of its use would be to increment values in a table, based a trigger. SQL has always had similar functionality with identity columns, but now this is a discrete object.
7. Distributed Replay -- Once again this is answer to a feature that Oracle released (Real Application Testing). However, and in my opinion where the real value proposition of SQL Server is, in Oracle it is a (very expensive) cost option to Enterprise Edition. With SQL, when you buy your licenses for Enterprise Edition, you get everything. Distributed replay allows you to capture a workload on a production server, and

replay it on another machine. This way changes in underlying schemas, support packs, or hardware changes can be tested under production conditions.

8. Power View -- You may have heard of this under the name "Project Crescent" it is a fairly powerful self-service BI toolkit that allows users to create mash ups of BI reports from all over the Enterprise.
9. SQL Azure Enhancements -- These don't really go directly with the release of SQL 2012, but Microsoft is making some key enhancements to SQL Azure. Reporting Services for Azure will be available, along with backup to the Windows Azure data store, which is a huge enhancement. The maximum size of an Azure database is now up to 150G. Also Azure data sync allows a better hybrid model of cloud and on premise solutions
10. Big Data Support -- I saved the biggest for last, introduced at the PASS (Professional Association for SQL Server) conference last year, Microsoft announced a partnership with Hadoop provider Cloudera. One part of this involves MS releasing a ODBC driver for SQL Server that will run on a Linux platform. Additionally, Microsoft is building connectors for Hadoop, which is an extremely popular NoSQL platform. With this announcement, Microsoft has made a clear move into this very rapidly growing space

Chapter 3: SYSTEM DEVELOPMENT

This Chapter includes the development of the application systematically. WordMan is a simple application which helps the user to find and send beautiful messages and greetings to their loved and special ones without any delay. The users can look for any kind of occasion and express their feelings without any hesitation. The system development helps in overall development of the application for smooth working of the project. In the beginning of the project, firstly the database is created and implemented according to the requirements of the application. Different

tables are created accordingly. When the database is implemented and the scripts are ready then further implementation is done which includes the Data Access Layer. This layer includes all the backend queries which are used to view categories, view messages, add categories, add message, edit categories, edit messages, delete category, delete messages, etc.

WordMan is very helpful when the user is unable to find the best messages and cannot write their thoughts on a piece of paper. This application is the one which helps to speak out the feelings of the user directly and make their special days more beautiful and special. In the data access layer, repository class is made which contains all the required methods and operations performed at the backend by the admin. After the implementation of Data Access Layer, the next is MVC implementation of the application. MVC includes all the frontend of the application. This helps the implementation of the user interface which helps the application to look beautiful and implements all the functionalities made by the admin.

Firstly, the guest user can view the application without any login details. The guest user has access to homepage, about us page, contact us page. The guest user can also give feedback by entering its emailed, username and the feedback in their respective text fields. Also, the user can view categories as per the occasion and further can view messages according to the particular category. When the guest tries to download the message it is redirected to the register page and will be asked for subscription. After the payment is done successfully then guest user will login as a registered user and then it has access to the whole web application without any hassle.

The registered user will directly login and can access anything according to the need. The registered user can edit its profile and change password. The registered user can download any number of messages. Also can rate the messages as well. Rest all the functionalities are accessible like view categories, view messages, download message as image, as pdf, as word document.

The Admin first logs in by entering the admin details. Admin performs all the backend functionalities Such as add category, update category, delete category and same is for the messages. Admin can view all the feedbacks given by the users both guest and registered.

3.1 Hardware Interface

The hardware interfaces define the hardware devices needed for an application to run as type of processor and the amount of memory required.

Hard Disk: 50GB

RAM: 2 GB

Processor: Intel Core i3 processor

Browser: Chrome, etc.

(Supports JavaScript, css3)

3.2 Software Interface

Operating System: Windows, Linux, Mac OS

Front End: HTML, CSS, JavaScript,

Back End: Asp.net MVC, C#

Database: SQL Server

3.2.1 Functional Requirements

Functional requirements define the internal workings of the software that is, the calculations, technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied. Functional requirements specify particular behaviors of a system. A typical functional requirement will contain a unique name and number, a brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, and to track the requirement through the development of the system. The core of the requirement is the description of the required behavior, which must be a clear

and readable description of the required behavior. This behavior may come from organizational or business rules, or it may be discovered through elicitation sessions with users, stakeholders, and other experts within the organization. Many requirements will be uncovered during the use case development. When this happens, the requirements analyst should create a placeholder requirement with a name and summary, and research the details later, to be filled in when they are better known. Software requirements must be clear, correct, unambiguous, specific, and verifiable

The basic functional requirements of the system are as follows:

Search Algorithm: The local search engine in this system lists all the keywords searched by the

User Registration: have to register themselves so that they can download messages

Admin Registration: Admin will generate account for Users.

Login System: Users must have to login to book coupons whereas users also have to login to get the greetings.

3.2.2 Non Functional Requirements

The basic non-functional requirements of the system are:

Reliability- Flexible services to users in future.

24*7 Availability

Maintainability-Optimizing the components for better performance at peak time.

3.2.3 Performance Requirements

Food Online Web Portal manages facilities required by the users quickly and easily. It offers to take orders faster with ease to both user and vendor online.

3.2.4 Other Requirements

Database: The records of all operation are stored in database

3.3 Project Development

3.3.1 Implementaion Of Database

While making MVC application, we need to decide the type of development approaches. There are basically types of development approaches that are:

- Database-first Approach
- Code-first Approach
- Code-first using an existing database Approach

In WordMan application, we have used the first approach that is Database-First Approach to further implement data access layer for the application.

Initially the database is created with the name ‘WordManDB’ database. Further, tables are created according to the requirement in the application. In total seven tables are created and values are inserted accordingly. Different constraints are applied in the columns to avoid any wrong inputs.

Each table consists of different attributes with their respective datatypes and size of the attribute.

Below is the class diagram of the database which is implemented. This diagram shows the overall layout of the database and the attributes of each table which their datatypes and size accordingly. The class diagram shows the association and dependency of each table with other tables

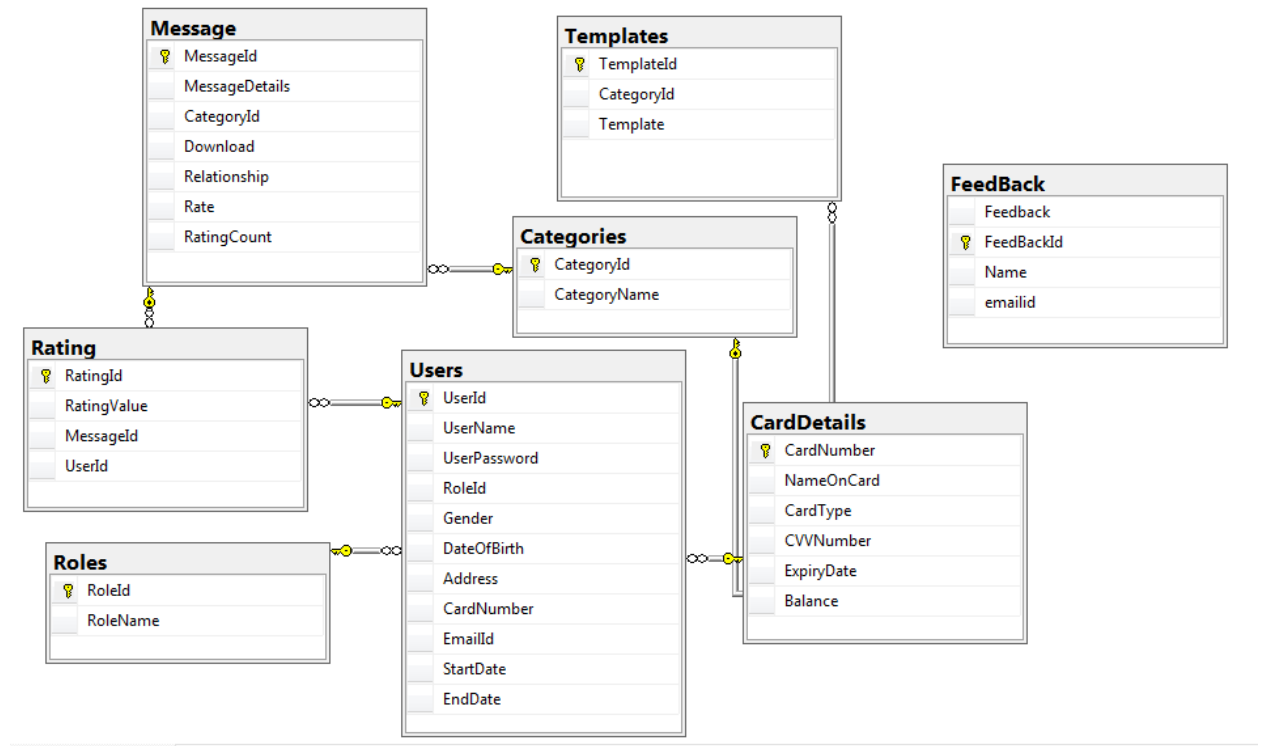


Figure 5: Class Diagram

Description of each table is listed below.

1.Users

The Users table is created by using create table query in database management system.

This table consists of all the details a user acquires such as UserId, UserName, UserPassword, RoleId, Card Number, Gender, Date of Birth, Address, EmailId, StartDate and EndDate. UserId being the primary key where as RoleId and CardNumber being the foreign keys for Roles table

S.No.	Field Name	Size	Type	Keys
1.	UserId	(1,1)	Int	Primary Key
2.	UserName	30	Varchar	Not Null
3.	UserPassword	15	Varchar	Not Null
4.	RoleId	1	Tinyint	Foreign Key
5.	Gender	1	Char	Not Null
6.	DateOfBirth	3	Date	Not Null
7.	Address	200	Varchar	Not Null
8.	Card Number	(16,0)	Numeric	Foreign Key
9.	Email Id	30	Varchar	Not Null
10.	Start Date	3	Date	Null
11.	End Date	3	Date	Null

and CardDetails table respectively. StartDate and EndDate are the starting and ending dates of subscription provided to the user by the admin.

Table 1: Users Table

2. Roles

This table consists of two roles that is Admin or Customer. RoleId being the primary key of the table.

Accordingly the role id's are given. RoleId 1 is given to Admin and RoleId 2 is given to Customer.

Admin is allowed to perform any operation in the application while the Customer can just view the operations performed and can make changes on its profile only.

S.No.	Field Name	Size	Type	Keys
1.	Role Id	(1,1)	Tinyint	Primary Key
2.	Role Name	20	Varchar	Unique

Table 2: Roles Table

3.Card Details

Card Details table consists of all the card details of the customer. The details include the card number, name on the card, expiry date, cvv number, card type and balance. Card Number being the primary key of the table. This table is used to make successful payment while taking subscription of the website.

S.No.	Field Name	Size	Type	Keys
1.	Card Number	(16,0)	Numeric	Primary Key
2.	Name On Card	40	Varchar	Not Null
3.	CardType	6	Char	Not Null
4.	CVV Number	(3,0)	Numeric	Not Null
5.	Expiry Date	3	Date	Not Null
6.	Balance	(10,2)	Decimal	Null

Table 3: CardDetails Table

4.Message

This table consists of all the messages according to the type of occasion. In this MessageId being the primary key and CategoryId being the foreign key for categories table.

MessageId is inserted as char datatype pf size 4 starting with M and preceding with the starting value 100. CategoryId being an identity in categories table and categories are listed according to the occasion or an event. Message field consists of the message which is displayed on the website.

Download field consists of the number of times a message is downloaded by the user. Relationship field consists of the type of relationship is searched by the user to find that particular message.

S.No.	Field Name	Size	Type	Keys
1.	Message Id	4	Char	Primary Key
2.	Message Details	500	Varchar	Not Null
3.	Category Id	1	Tinyint	Foreign Key
4.	Download	4	Int	Null
5.	Relationship	15	Varchar	Null

Table 4: Message Table

5.Feedback

This table consists of the feedback given by the user .The feedback can be given by the guest user or subscribed user. The admin can view the feedbacks of all the users .

The table contains FeedbackId as identity, Feedback as a textfield to input text by the user, Name of the user and EmailId accordingly. Feedback is not mandatory for the user.

S.No.	Field Name	Size	Type	Keys
1.	FeedbackId	(1,1)	Int	Primary Key
2.	Feedback	500	Varchar	Not Null
3.	Name	15	Varchar	Not Null

4.	Email	50	Varchar	Not Null
----	-------	----	---------	----------

Table 5: Feedback Table

6.Categories

Categories table consist of all the occasions which a user searches for in the application. Categories are Diwali, Holi, New Year, Birthday, Farewell, Joining Anniversary, etc.

This table consists of CategoryId and Category Name fields. CategoryId being an IDENTITY and PRIMARY KEY.

S.No.	Field Name	Size	Type	Keys
1.	Category Id	(1,1)	Tinyint	Primary Key
2.	Category Name	20	Varchar	Not Null

Table 6: Categories table

7. Rating

Rating table consists of all the ratings given to the message by the user. This table is helps the other users to select the highly rated messages and send the best message to their loved ones. RatingId being an IDENTITY and PRIMARY KEY of the table. Rating Value is the value

selected by the user out of 5. MessageId being the foreign key for message table and emailId contains the email of the user who is rating the message.

S.No.	Field Name	Size	Type	Keys
1.	RatingId	(1,1)	Int	Primary Key
2.	RatingValue	4	Int	Null
3.	MessageId	4	Char	Foreign Key
4.	UserId	4	Int	Foreign Key

Table 7: Rating Table

8. Template

This table includes the template Id, template and categoryId fields. The TemplateId being the primary key and CategoryId being the foreign key.

Template field includes some templates which the user will select according to its choice and can download the message as an image with the selected template in the background.

<u>S.No.</u>	<u>Field Name</u>	<u>Size</u>	<u>Type</u>	<u>Keys</u>

1.	TemplateId	4	Char	Primary Key
2.	CategoryId	1	Tinyint	Foreign Key
3.	Template	Max	Varbinary	Not Null

TABLE 8: Template Table

The Data Access Layer of an MVC application consists of all the backend functionalities which are to be performed and showed on user interface. These functionalities include numerous methods which includes different algorithms using hard coding and otherwise. In our we application WordMan we have made the data access layer by the name “Infosys.WordMan.DataAccessLayer”. This is an .NET core project which includes models, dependencies, appsettings.json file, a repository class named as ‘WordManRepository.cs’.

The models contains the classes which are automatically generated with the help of scaffolding. The approach which we have selected that is the database first approach is also known as Reverse engineering. On the basis of database which is already implemented the run-time libraries which is the EF Core API and Entity framework core commands, these two are required to create the Entity Data Model. Entity is the C# classes which are represented in the form of database tables. The data model represents the relationship between the classes. Now, the scaffolding which is done by executing the scaffold-dbcontext command in the Package Manager Console which is further used to create entity data model. Scaffolding command includes the connection string which is used to establish a secure connection with the database which is already implemented. The database name and the server name is mentioned in the connection string of the command. The provider string provides the database provider name which contains all the runtime libraries required to communicate with the database. There is one to many mapping with the entities.

The models includes classes of all the tables such as:

- CardDetails.cs
- Categories.cs
- Feedback.cs
- Message.cs
- Rating.cs
- Roles.cs
- Templates.cs
- Users.cs

Also a context class is added named as 'WordManDBContext.cs'. The context class represents a secure and successful connection with the implemented database which is used to perform various database operations that are the CRUD operations that is create, update, delete and read using a complete collection of the objects of entity. This class is very helpful in managing database connections, configuration of entities and the relationship between them, perform the basic create, update, delete and read operations.

The appsettings.json file must be present in the root folder of the application. It contains the connection string which has a key-value pair in which key is the WordManDBConnectionString and value is the data source, initial catalog, and integrated security should be true.

The methods that are written to perform the functionalities are mentioned below:

- GetCategories

This method returns the list of categories by writing the appropriate linq query.

- GetCategoryNames

This method returns the list of type string which just gives the names of the categories.

- **GetRelationships**

This method returns the list of datatype string which gives the list of relationships extracted from the relationship table by executing the query.

- **IsEmailAvailable**

This method returns the bool value which tells whether the email id is available or not from the users table by executing the appropriate query.

- **GetAllMessages**

This method returns the list of type messages by executing the ling query.

- **GetMessageOnCategoryId**

This method returns the list of type messages on particular category id from categories table.

- **AddCategory**

This method takes category name as input and returns a bool value whenever a new category is added.

- **AddMessage**

This method takes object of message as input and returns a bool value whenever a new message is added.

- **UpdateCategory**

This method takes object of categories as input and returns a bool value whenever the category is updated.

- **UpdateMessage**
This method takes object of messages as input and returns a bool value whenever the message is updated.
- **DeleteCategory**
This method takes object of categories as input and returns a bool value whenever the category is deleted.
- **DeleteMessage**
This method takes object of messages as input and returns a bool value whenever a message is deleted.
- **AddFeedBack**
This method takes object of feedback as input and returns a bool value whenever a feedback is added.
- **GetFeedBack**
This method returns a list of datatype feedback which gives the list of all feedbacks on successful execution of query.
- **ValidateCredentials**
This method takes two inputs : username of type string and password of type string. It returns the byte value and checks whether the inputs entered are valid or not.
- **RegisterUser**

This method takes object of users as input and returns a bool value whenever a new register is added.

- EditUserProfile

This method takes object of users as input and returns bool value and edits the details of the user correspondingly.

- ValidateCardDetails

This method takes object of carddetails as input and returns an integer value and checks the card details of the user and matches with card details inserted in the table.

- GetNextMessageId

This method returns a string value and gives next message id to the user without writing it manually.

- SearchMessages

This method is the most important method of the project as it returns the string value and searches for any type of input entered and searches the input in the corresponding categories and messages table.

- ChangePassword

This method takes two inputs: newPassword of type string and userId of type int. It changes the password to the new password entered of the corresponding userId.

- GiveRating

This method takes object of rating as input and returns the integer value by executing the query to get rating.

- AddRating

This method takes object of rating as input and returns the integer value by executing the query to add rating which is added by the user.

- TopRatedMessages

This method returns The list of type messages from the message table which are top rated by executing the query to calculate rating to get top rated messages.

- CalculateRating

This method returns a bool value and calculates overall rating of a particular message.

- UpdateMessageRating

This method takes object of message as input and returns bool value whenever the rating of the particular message is updated. It swaps the old rating with new one if the same user rates the message.

- IncDownload

This method takes object of message as input and returns bool value whenever a download is incremented. It performs the operation on number of times a message is downloaded.

- getUserDetail

This method takes userid as input of type integer and returns an object of type users. The object contains all the user details of the corresponding userid which is given as input.

3.3.2 MVC Implementaion

Next Step after implementation of data access layer is creating a new project of ASP.NET MVC application named as 'Wordman'. In this project there are different controllers and views which runs on browser and functionalities are performed on user interface. The application also contains the models which are already present in data access layer.

3.3.2.1 Controllers

The controllers are the classes which are derived from the base classes. These classes contain the methods which are public named as Action methods.

Controllers are helpful in interaction of user with that of mvc application. It helps in determining the request given by the user and controls how to respond back to that request.

In this class, a parametrized constructor is made and in that two parameters are passed: Object of WordManRepository class and second is the object of IMapper class. IMapper class helps in mapping the entities of data access layer and mvc models with each other.

There are in total 4 controllers used in the project which are listed below:

- AdminController
- GuestController
- HomeController
- UserController

Admin Controller:

This controller controls the basic functionalities of the project. All the addition, updation and deletion operations are performed by the admin. Admin controller handles the backend system of the project.

The functionalities or methods implemented by the admin are:

- AdminHome
- AddCategory
- SaveAddedCategory
- ViewMessages
- ViewCategory
- DeleteCateGory
- SearchMessages
- EditMessage
- EditCategory
- SaveEditCategory
- SaveEditMessages
- DeleteMessages
- AddMessages
- SaveAddMessages
- ViewFeedBack

All the methods perform functionalities by calling the corresponding methods written in the repository class using mapper object passed in constructor.

Guest Controller:

This controller contains the functionalities which a guest can view until it is redirected to the register page. The views which are viewed by the guest are the home page, about us page, contact us page, feedback page, categories page, messages page.

Following are the functions implemented in the controller:

- GuestHome
- ViewCategories
- ViewMessages
- SearchMessages
- GetTopRatedMessage
- Getrating
- SearchSaveFeedBack
- AboutUs
- ContactUs

Home Controller

This controller views the functionalities of a home page. Whatever it consists of, it shows all of them that are home page, about us, contact us, feedbacks, etc.

Below mentioned are the functionalities of a home controller:

- Index
- Checkrole
- Logout
- About
- Contact
- FeedBack
- FeedBackNew

- RegisterUser
- SaveRegisterUser
- CheckEmailId
- IsAlreadySigned
- ValidateDate

User Controller:

This controller is used to view all the basic functionalities a user can view which includes everything. The user is the registered user and can access each and every functionality of the website.

This controller performs the following methods which are listed below:

- UserHome
- ViewMessages
- ViewCategory
- DeleteCateGory
- SearchMessages
- GetTopRatedMessage
- EditMessage
- Getrating
- IncrementDownload
- EditUser
- SaveChangedPassword
- SaveEditedUser
- SearchSaveFeedBack
- SortybyratingViewMsg
- SortybyDownloadsViewMsg

- SortbyRatingSearch
- SortbyDownloadSearch
- CheckAccount

3.3.2.2 Views

Views is a form of user interface which handles all the front end functionalities and performs basic tasks. Each controller has various views and a shared folder is also created which contains all the shared views which are common to all the controllers. These views are the cshtml files which contains both html and css documents.

Shared views are listed below:

- _AdminLayout.cshtml
- _CookieConsentPartial.cshtml
- _FinalLayout.cshtml
- _FinalLayout2.cshtml
- _FinalLayout3.cshtml
- _HomeLayout.cshtml
- _HomeLayout1.cshtml
- _Layout.cshtml(default)
- _ValidationScriptsPartial.cshtml
- EditLayout.cshtml
- Error.cshtml
- ErrorLogin.cshtml

- ErrorRegister.cshtml

Home Views

1. Home pages(Slide Show):

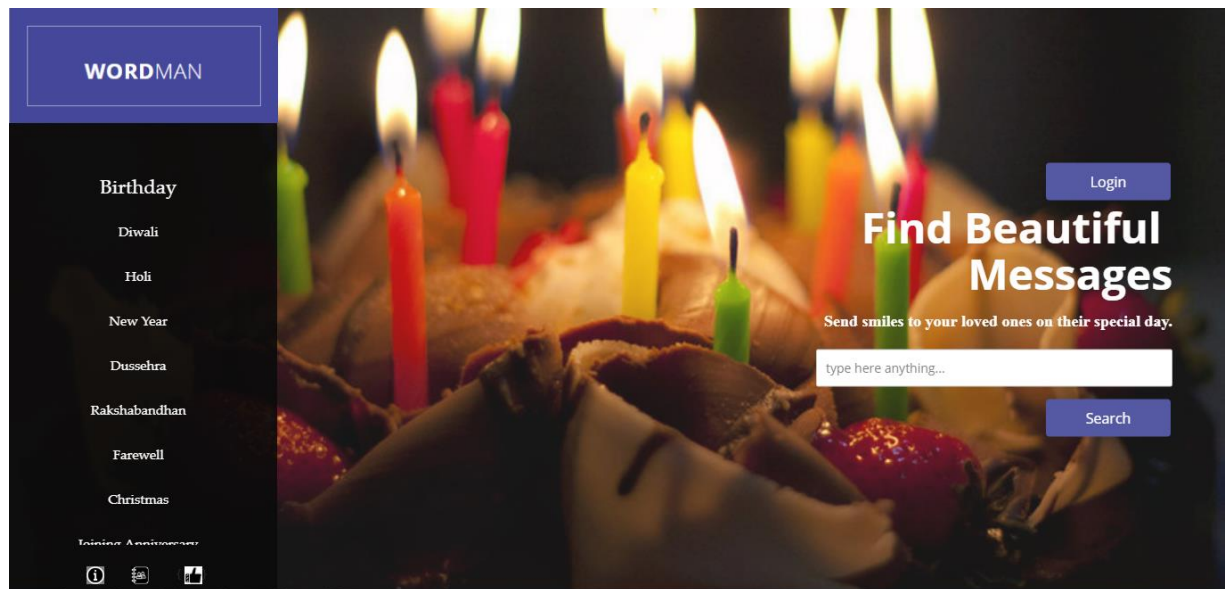


Figure 6: Home Page 1

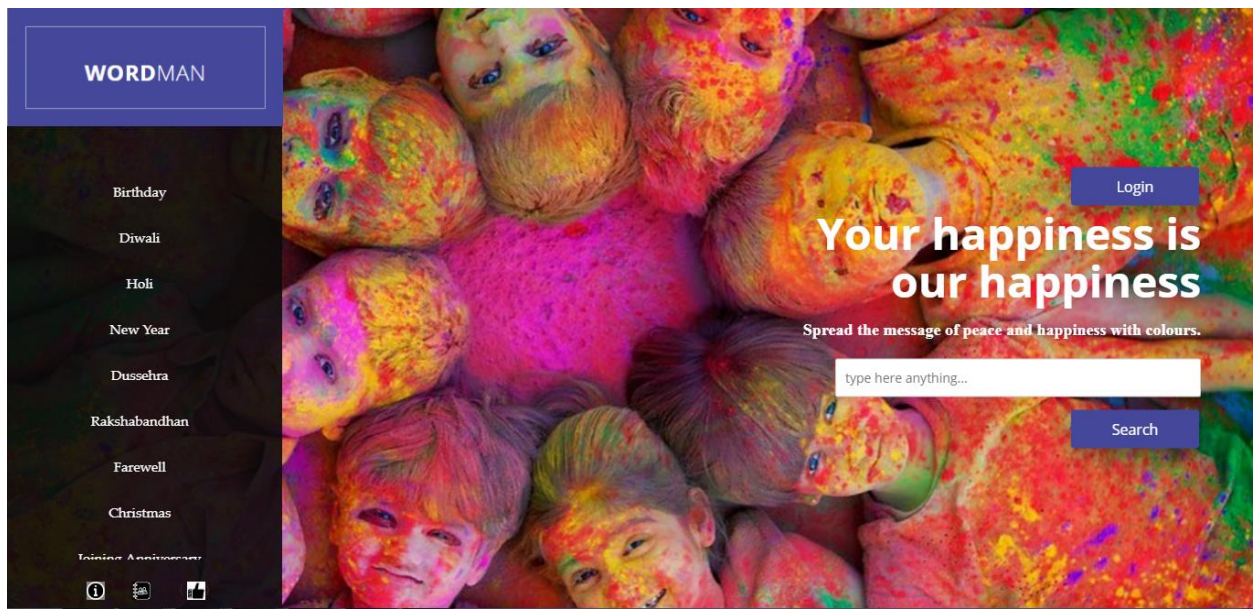


Figure 7: Home Page 2



Figure 8: Home Page 3

2.About Us

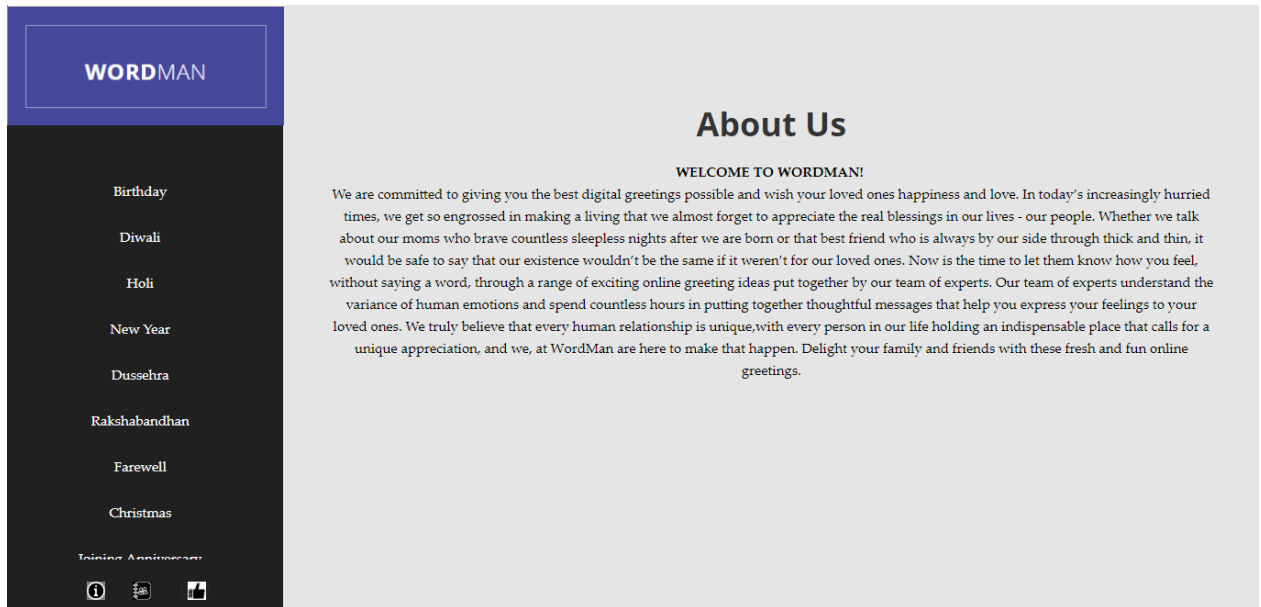


Figure 9: About Us

3.ContactUs(SlideShow)

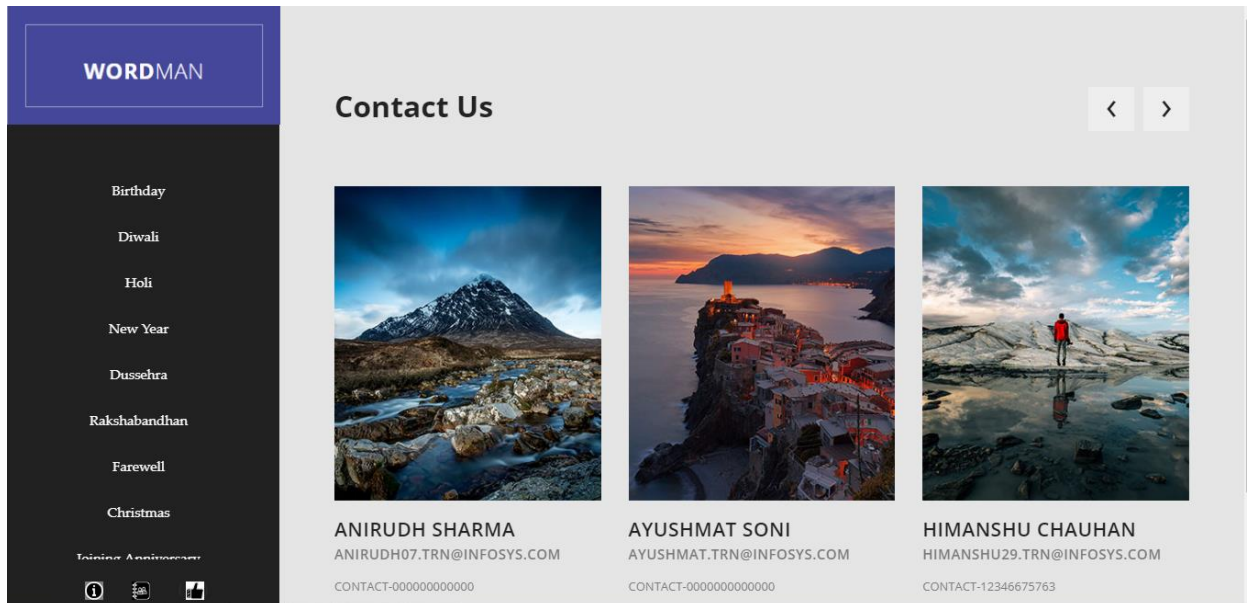


Figure 10: Contact Us 1

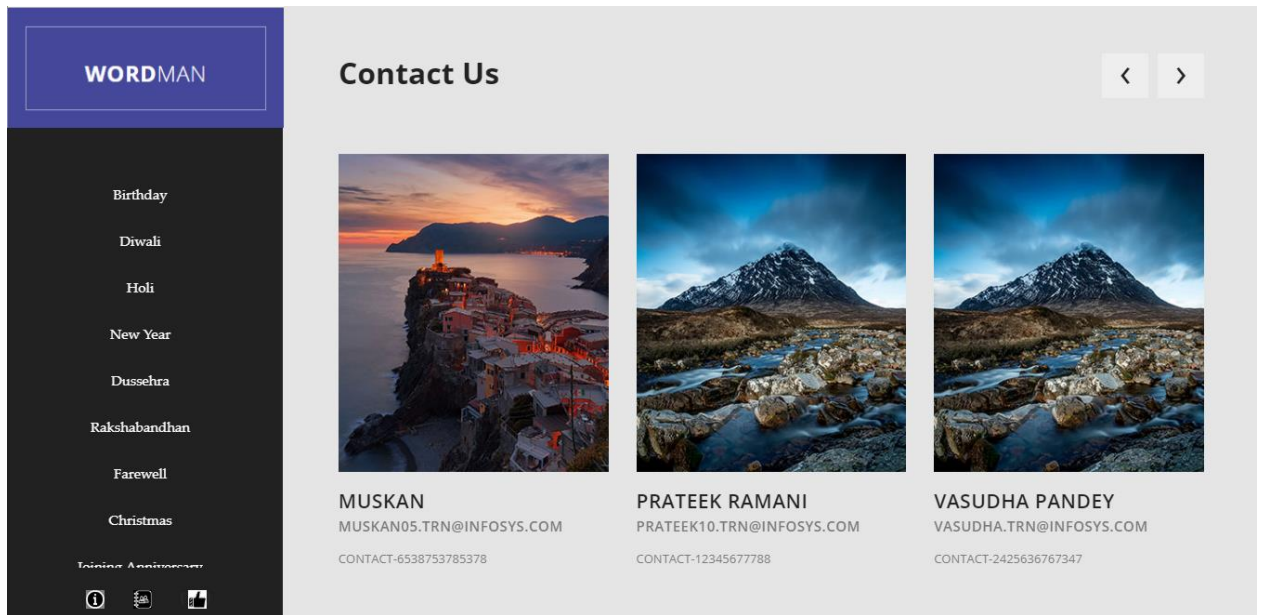


Figure 11: Contact Us 2

4.Feedback

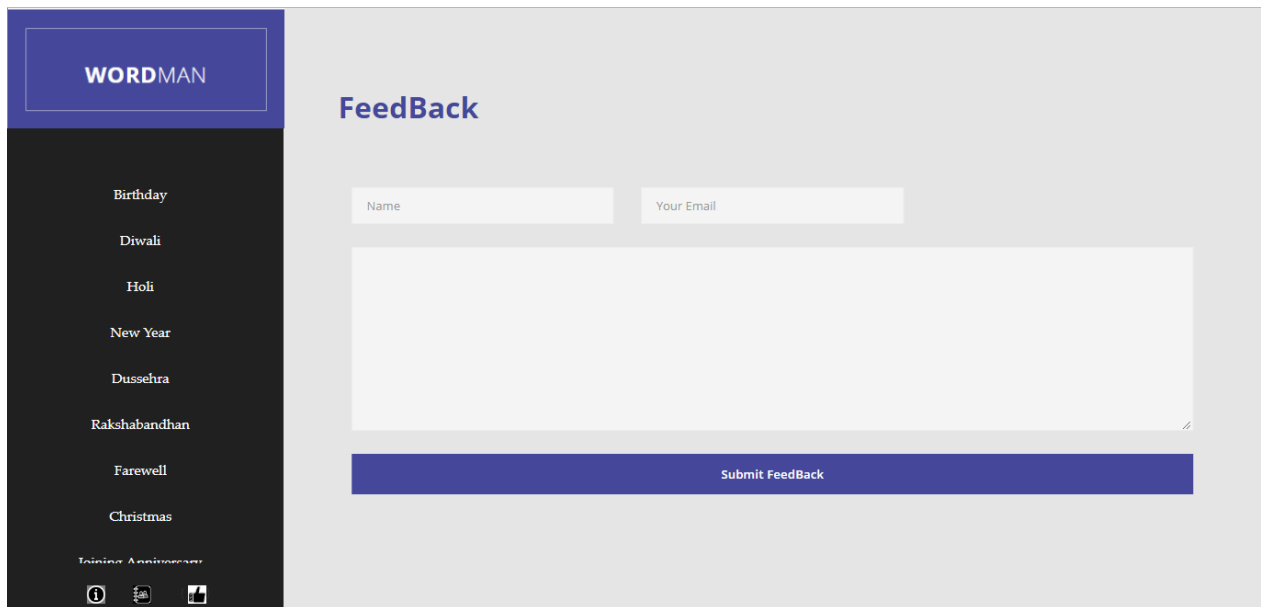


Figure 12: Feedback

User

1.User Login Page

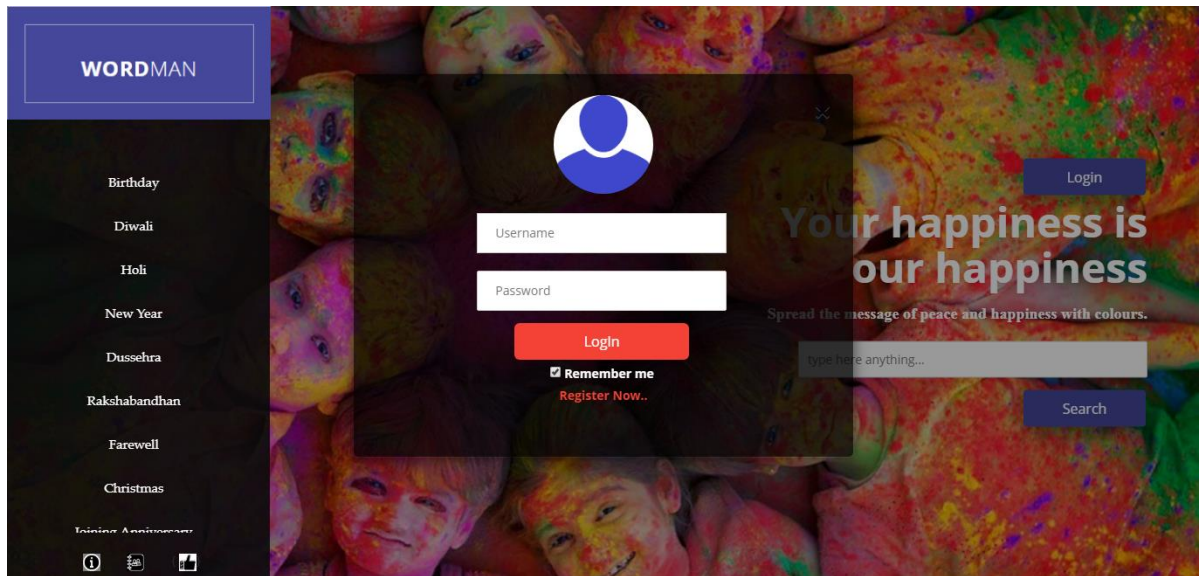


Figure 13: Login Page Of User

2.User Home Page

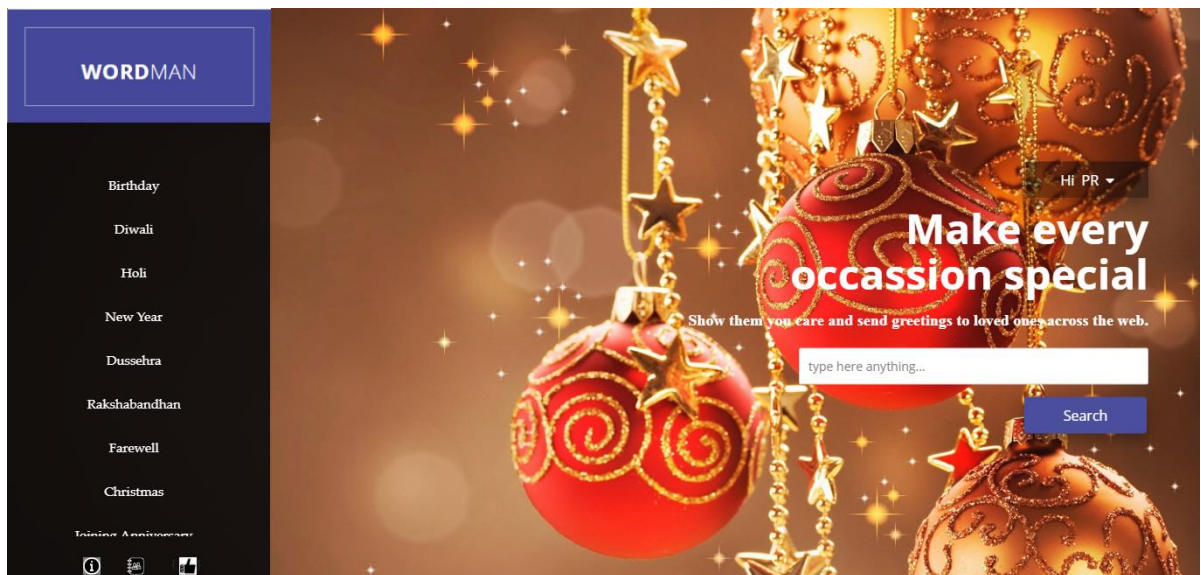


Figure 14: Home Page Of User

3.View Messages by category

Category birthday:

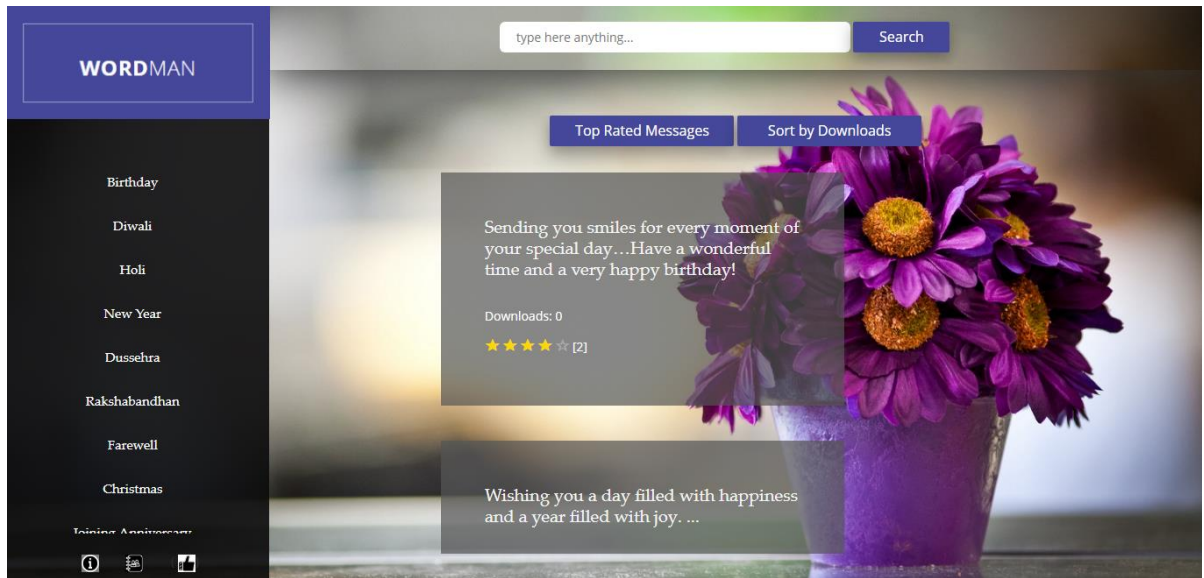


Figure 15: Messages of Category Birthday

Category: Farewell

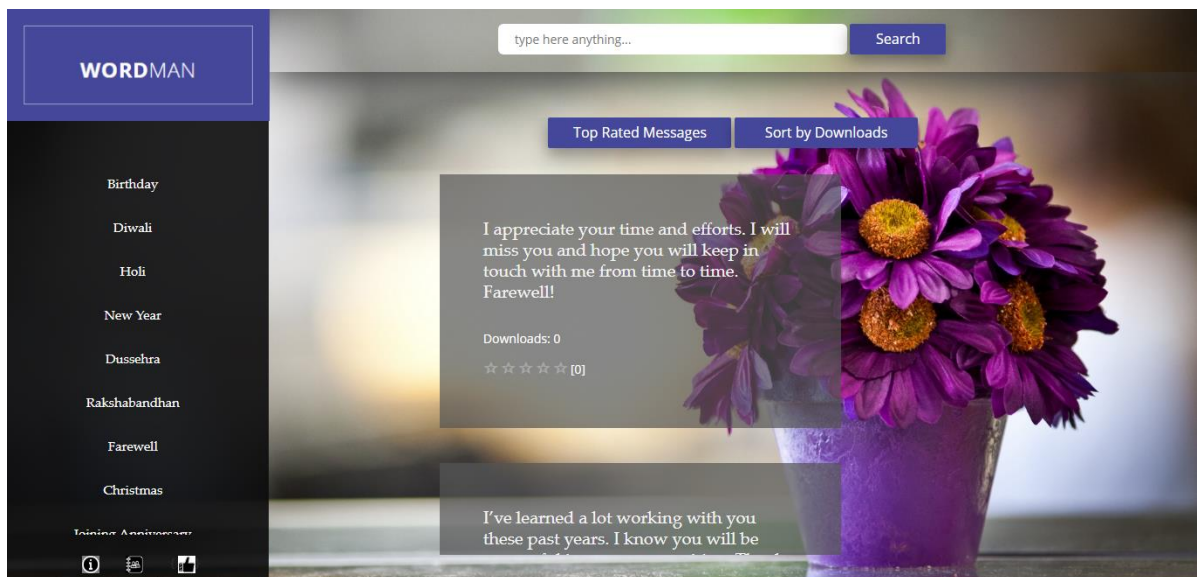


Figure 16: Messages of Category Farewell

Category Joining Anniversary

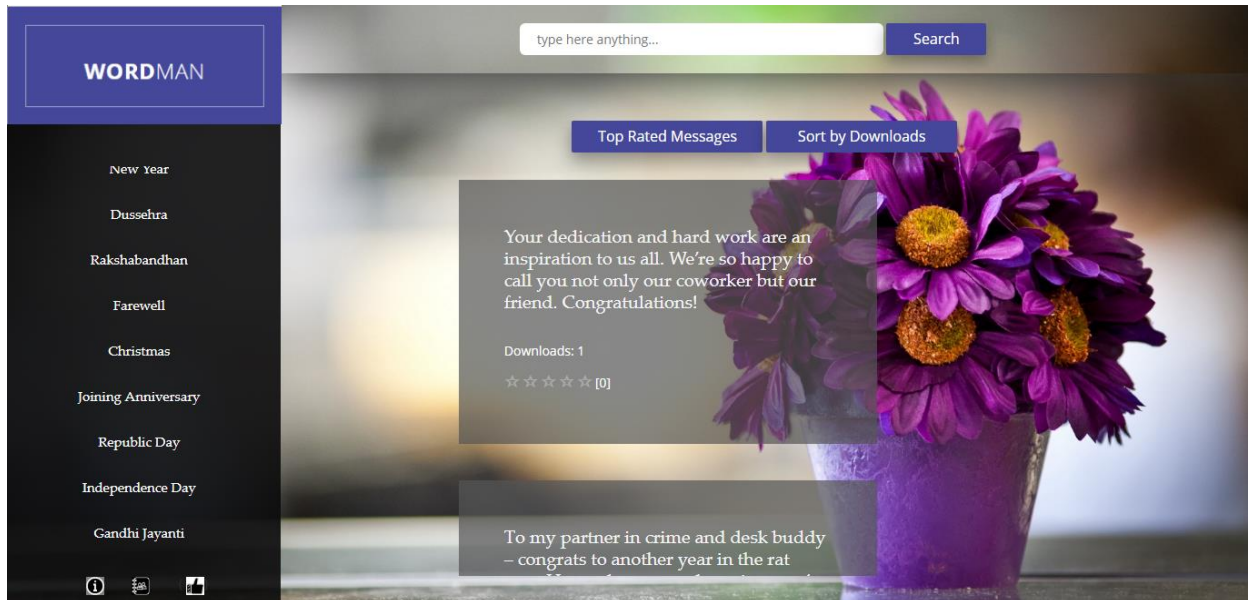


Figure 17: Messages of Category Joining Anniversary

4.Edit User Details

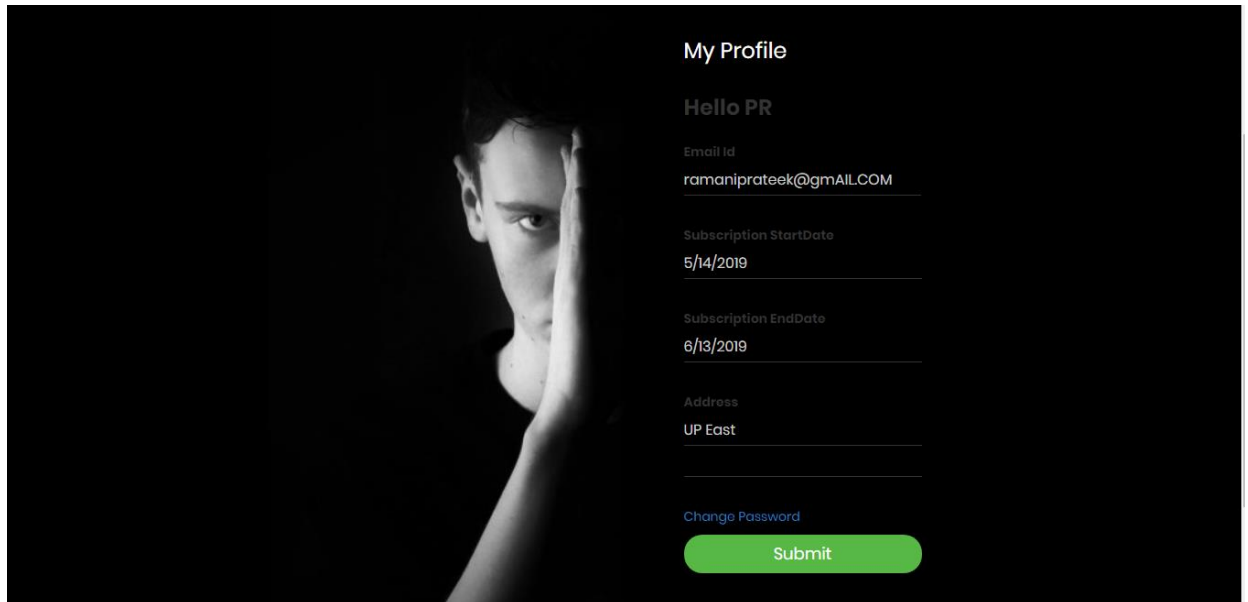
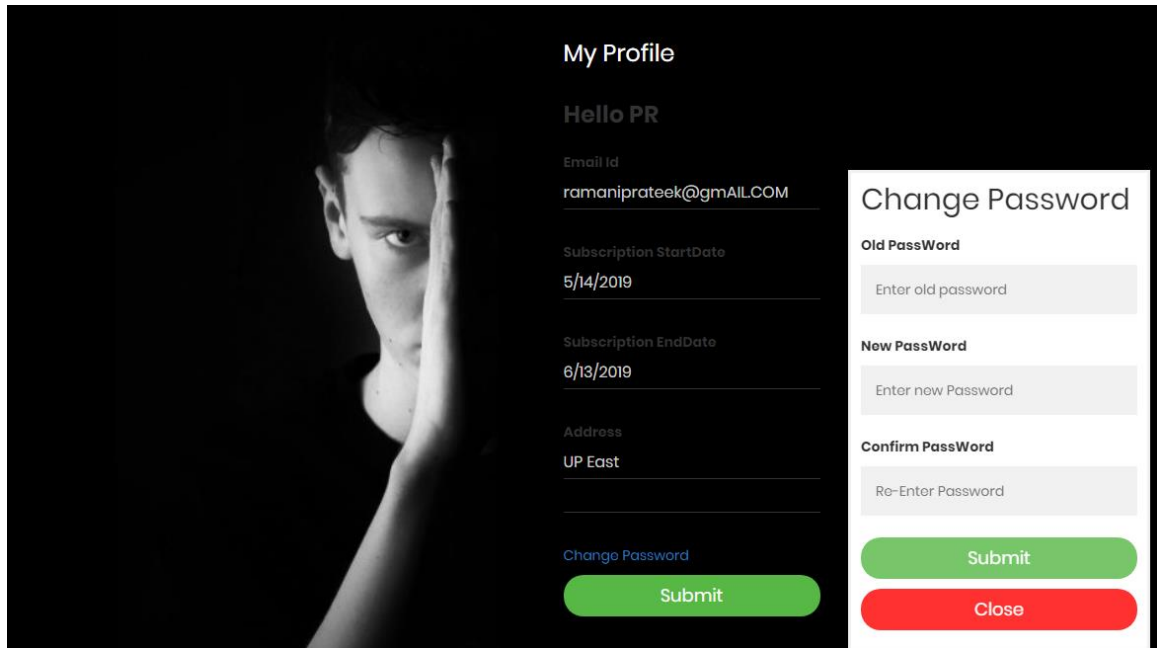


Figure 18: Edit Details

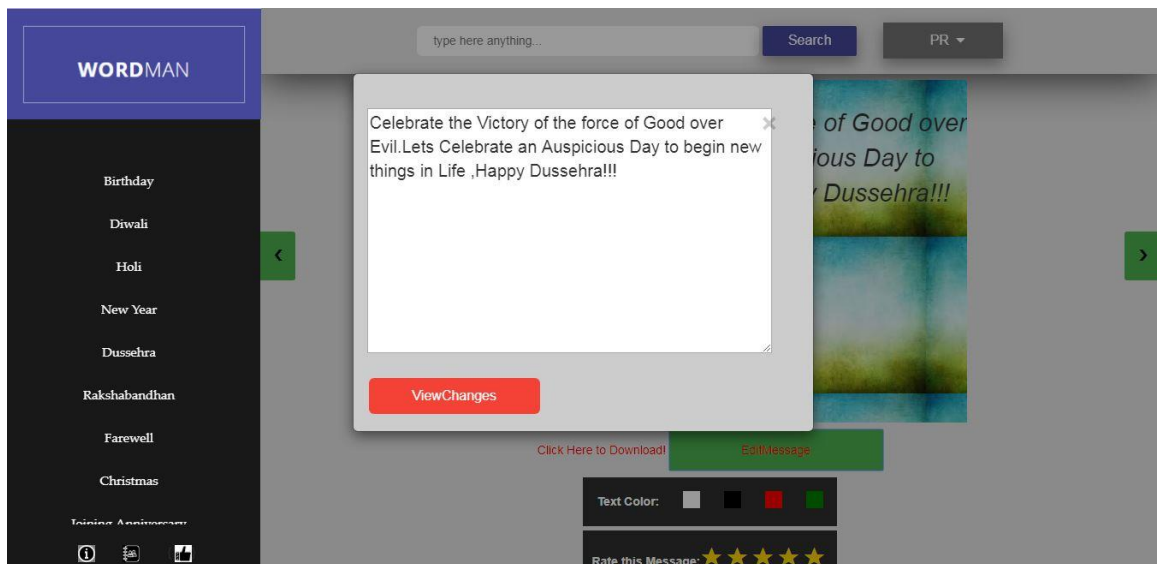
5.Change Password of Registered User(Pop Up)



The screenshot shows a user profile page titled "My Profile" with a background image of a person. The profile details include: "Hello PR", Email Id: ramaniprateek@gmail.COM, Subscription StartDate: 5/14/2019, Subscription EndDate: 6/13/2019, and Address: UP East. A "Change Password" button is visible below the address. A "Change Password" pop-up form is overlaid on the right side of the page. The form contains three input fields: "Old PassWord" (labeled "Enter old password"), "New PassWord" (labeled "Enter new Password"), and "Confirm PassWord" (labeled "Re-Enter Password"). At the bottom of the form are two buttons: a green "Submit" button and a red "Close" button.

Figure 19: Change password Pop Up

6.Edit Message



The screenshot shows a web application interface for editing a message. On the left is a sidebar with the "WORDMAN" logo and a list of categories: Birthday, Diwali, Holi, New Year, Dussehra, Rakshabandhan, Farewell, and Christmas. The main content area displays a message box with the text: "Celebrate the Victory of the force of Good over Evil.Lets Celebrate an Auspicious Day to begin new things in Life ,Happy Dussehra!!!". Below the message box is a red "ViewChanges" button. To the right of the message box is a green "EditMessage" button. At the bottom of the message box, there is a "Text Color:" selector with four color options (white, black, red, green) and a "Rate this Message:" section with five yellow stars. The background of the main content area shows a blurred image of a person.

Figure 20: Edit Message

7. Top Rated Messages

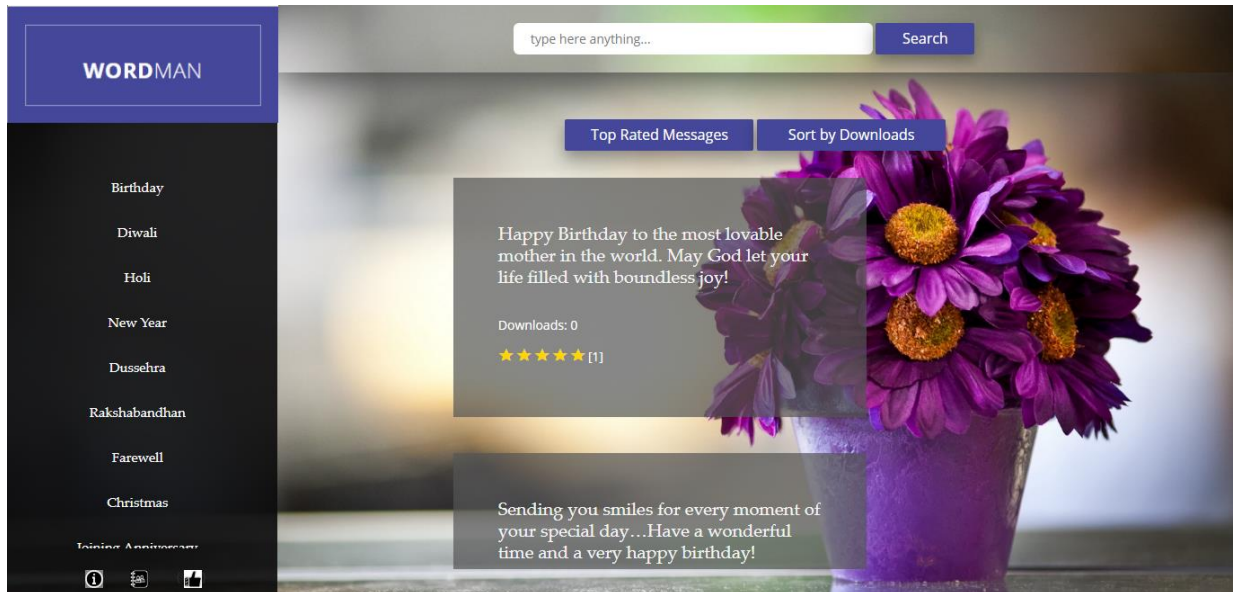


Figure 21: Top Rated Messages

8. Sort By downloads

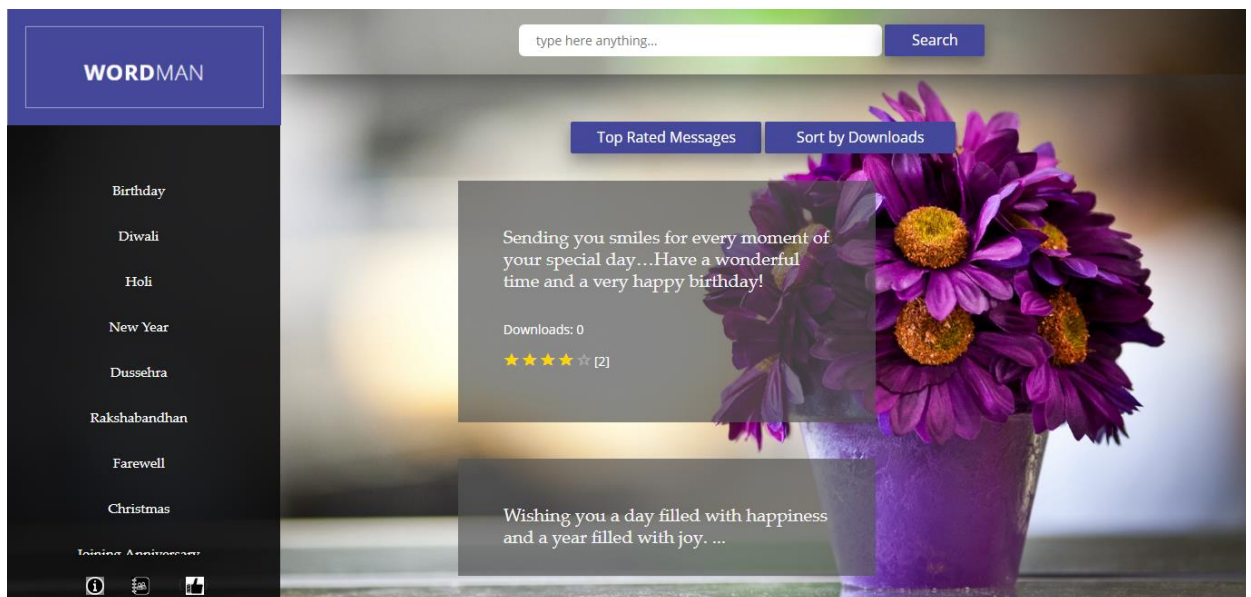


Figure 22: Sort By downloads

9.View Message By Templates

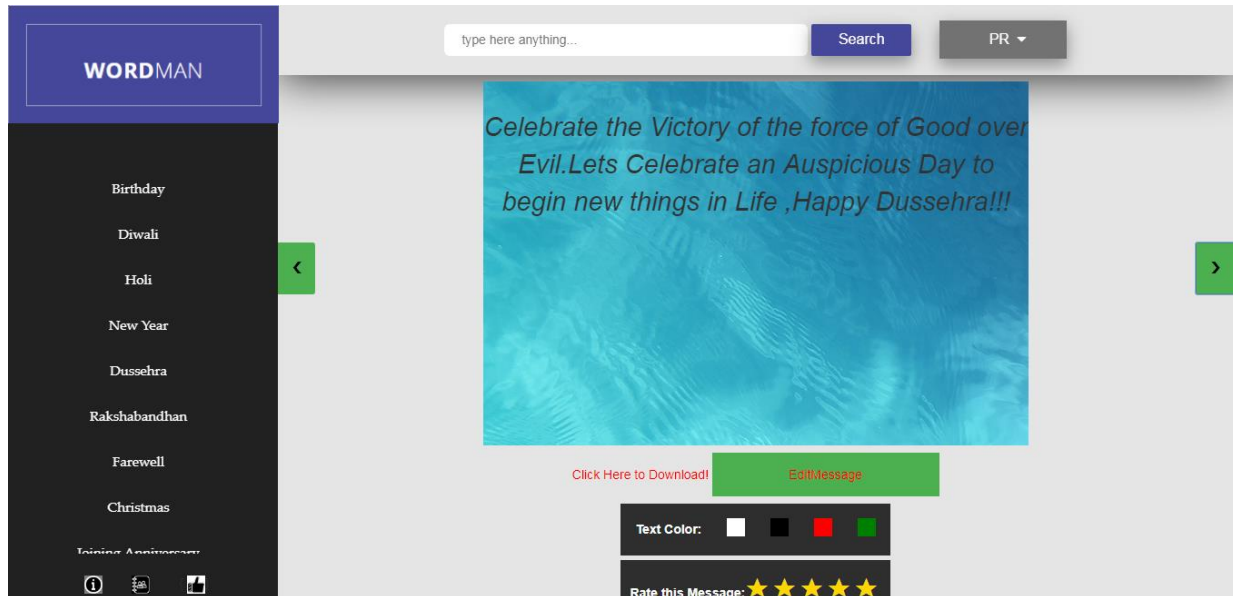


Figure 23: View Message By Template 1

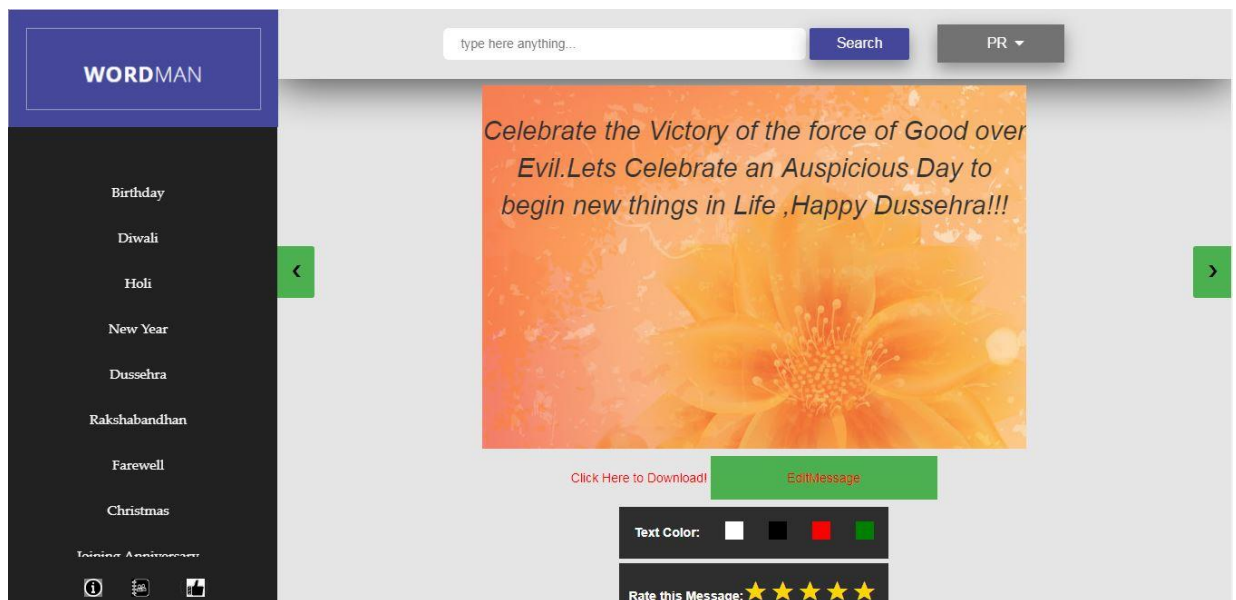


Figure 24: View Message By Template 2

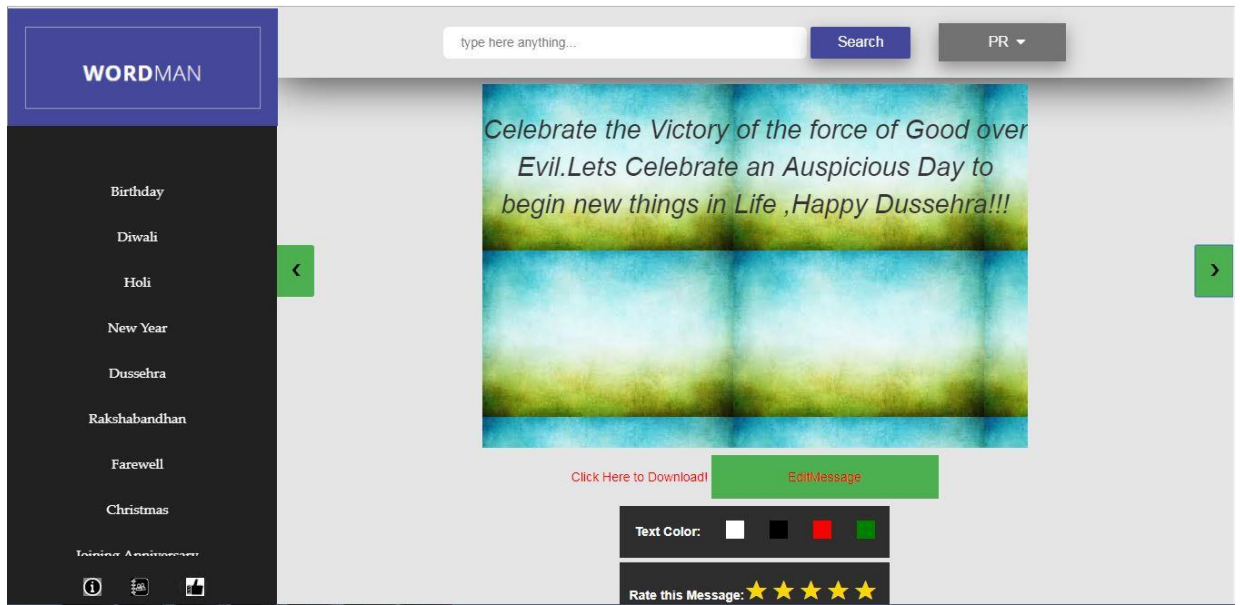


Figure 25: View Message By Template 3

Admin

1.Admin Home Page

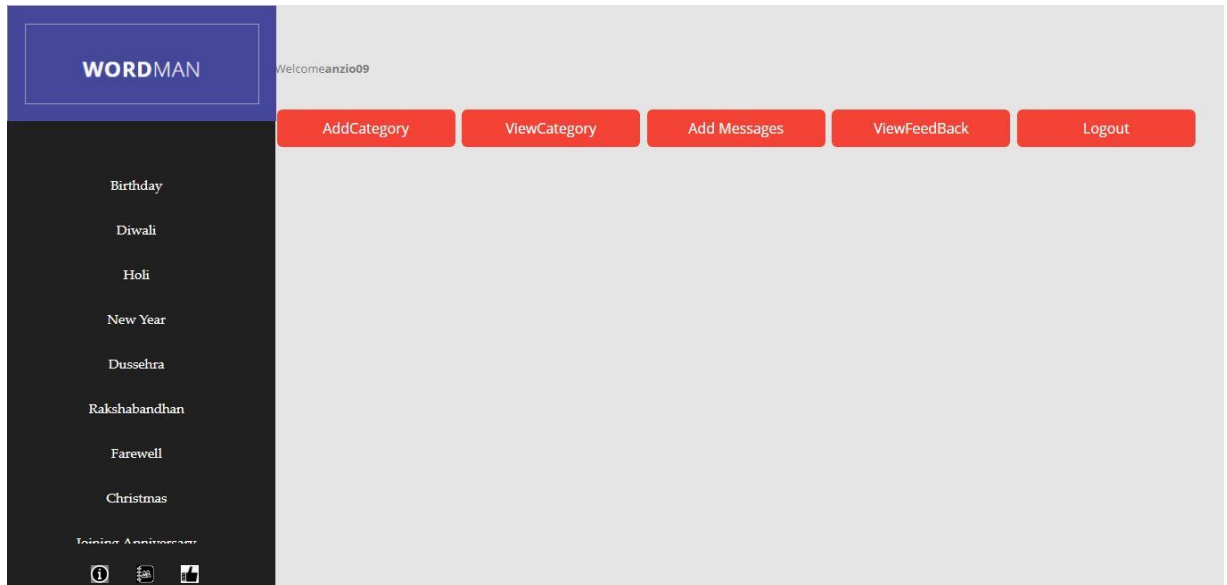


Figure 26: Home Page Of Admin

2.Add Category

The screenshot shows the 'AddCategory' form in the WORDMAN application. On the left is a dark sidebar with the 'WORDMAN' logo at the top and a list of categories: Birthday, Diwali, Holi, New Year, Dussehra, Rakshabandhan, Farewell, and Christmas. At the bottom of the sidebar are icons for 'Training Assistant', a document, and a social media icon. The main content area has a light gray background. At the top, it says 'AddCategory' and 'Categories'. Below this is a form with two input fields: 'CategoryId' and 'CategoryName'. A red 'Add category' button is positioned below the 'CategoryName' field. At the bottom of the form is a blue link that says 'Back to List'.

Figure 27: Add Category

3.Add Message

The screenshot shows the 'AddMessages' form in the WORDMAN application. The sidebar is identical to the one in Figure 27. The main content area has a light gray background. At the top, it says 'AddMessages' and 'Message'. Below this is a form with three input fields: 'MessageDetails', 'Relationship', and 'CategoryId'. An 'AddMessage' button is positioned below the 'CategoryId' field. At the bottom of the form is a blue link that says 'Back to List'.

Figure 28: Add Message


3.View categories

Categories		
CategoryId	CategoryName	
1	Birthday	Edit Name ViewMessage Delete
2	Diwali	Edit Name ViewMessage Delete
3	Holi	Edit Name ViewMessage Delete
4	New Year	Edit Name ViewMessage Delete
5	Dussehra	Edit Name ViewMessage Delete
6	Rakshabandhan	Edit Name ViewMessage Delete
7	Farewell	Edit Name ViewMessage Delete
8	Christmas	Edit Name ViewMessage Delete
9	Joining Anniversary	Edit Name ViewMessage Delete
10	Republic Day	Edit Name ViewMessage Delete
11	Independence Day	Edit Name ViewMessage Delete
12	Gandhi Jayanti	Edit Name ViewMessage Delete

Figure 29: View Categories

Guest

!. Register User



Registration Info

UserName

Password

Email

Date of birth

mm/dd/yyyy

Gender

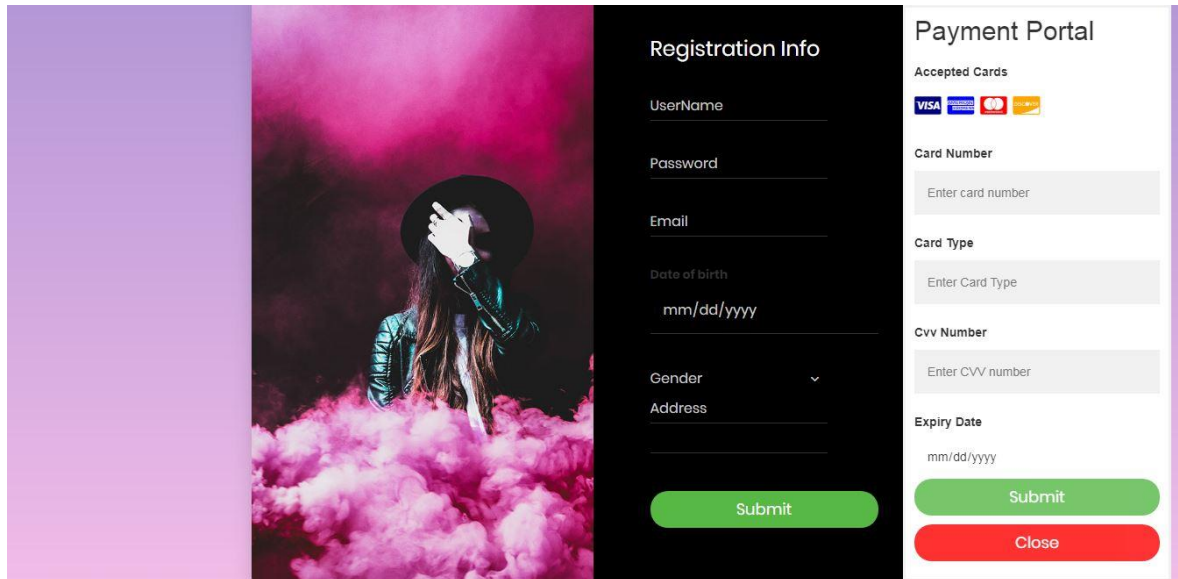
▼

Address

Submit

Figure 30: Register New User

2.Payment Portal(Pop Up)



The image shows a payment portal pop-up with a dark theme. On the left, there is a vertical purple bar and a central image of a person with long hair and a black hat, surrounded by pink smoke. The main form area is black and contains the following fields:

- Registration Info**
 - UserName:
 - Password:
 - Email:
 - Date of birth: mm/dd/yyyy
 - Gender:
 - Address:

At the bottom of the registration info is a green **Submit** button.

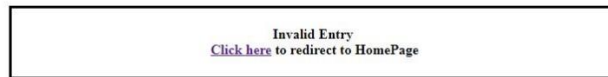
On the right, there is a white **Payment Portal** section with the following fields:

- Accepted Cards**: VISA, MasterCard, American Express, Discover
- Card Number**: Enter card number
- Card Type**: Enter Card Type
- Cvv Number**: Enter CVV number
- Expiry Date**: mm/dd/yyyy

At the bottom of the payment portal are two buttons: a green **Submit** button and a red **Close** button.

Figure 31: Payment Portal Pop Up

Error Pages



The image shows an error page with a white background and a black border. The text reads:

Invalid Entry
[Click here](#) to redirect to HomePage

Figure 32: Error Page 1

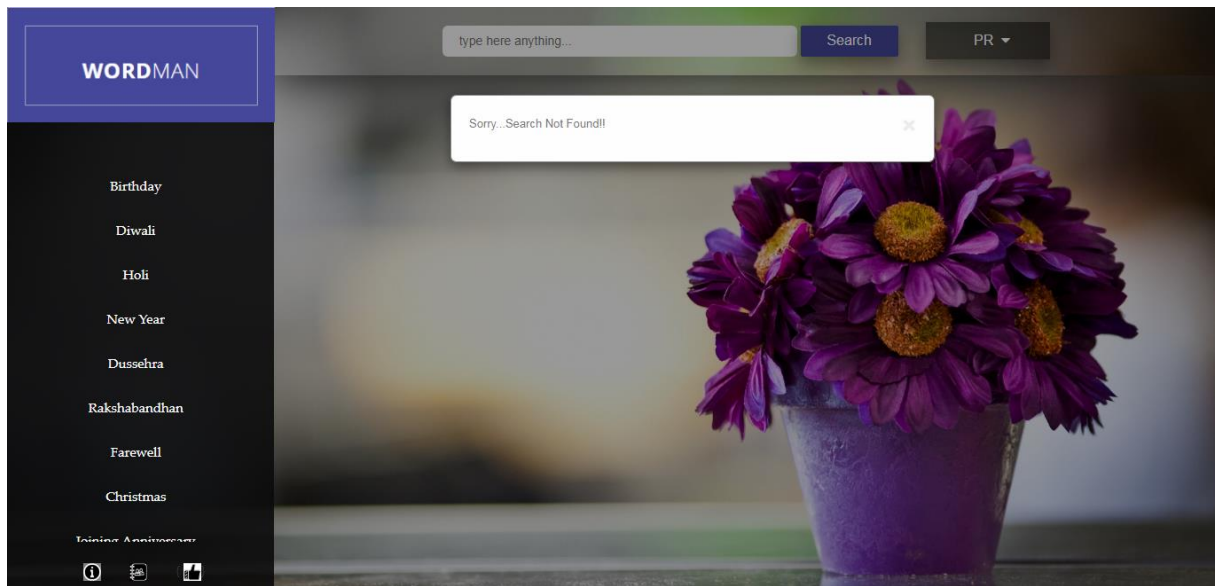


Figure 33: Error Page 2

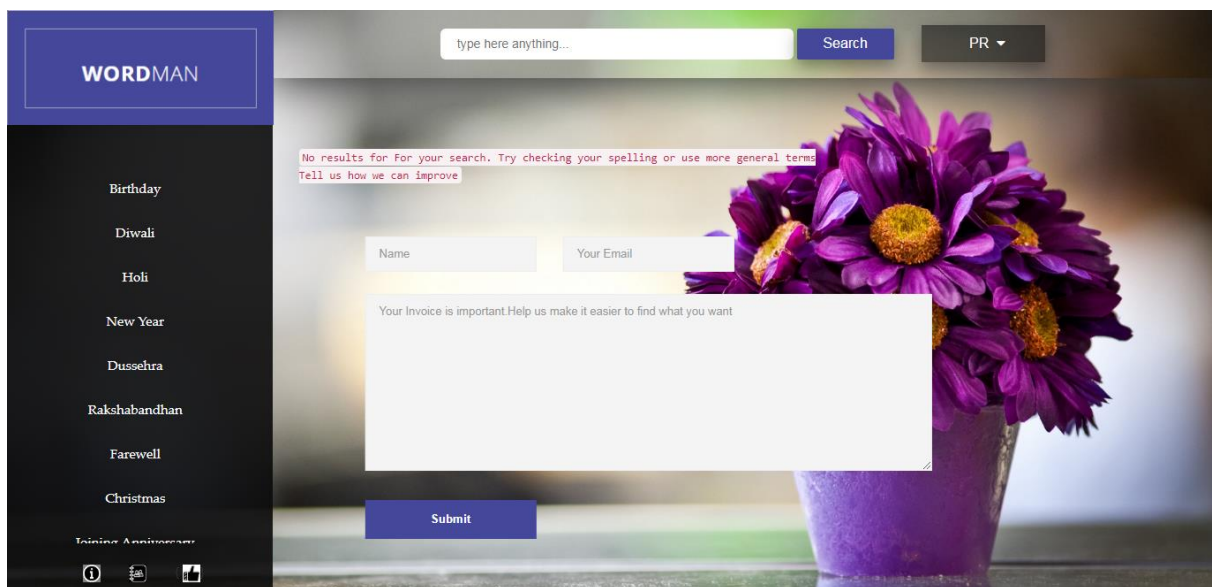


Figure 34: Error Page 3

Chapter 4: PERFORMANCE ANALYSIS

4.1 TEST ENVIRONMENT

Hardware : Windows 7 Desktop

**Software : Microsoft Visual Studio, 32-bit Operating System,
: Microsoft SQL Server 2012.**

Special test needs : NA

4.2 UNIT TESTS

UNIT TEST 1

- | | |
|--|---|
| 1. Author | : Prateek Ramani |
| 2. Functionality Tested | : Annotations in register field. |
| 3. Test Execution Procedure | : Manual |
| 4. Test Results Checking Method | : Manual |

Sl: No:	Condition to be tested	Expected Result	Passed/Fail
1	User name is entered as blank spaces.	Message Displayed: Enter valid user name	P
2	DOB is Incorrect	Message Displayed: Enter valid DOB	P
3	Password is entered as blank spaces.	Message Displayed: Please enter any other password	P

4	Any Field left empty	Message Displayed: Field cannot be left empty	P
---	----------------------	--	---

Table 9: Test Data 1

UNIT TEST 2

1. Author : Himanshu Chauhan
2. Functionality Tested : Check annotations in login page.
3. Test Execution Procedure : Manual
4. Test Results Checking Method : Manual

Sl: No:	Condition to be tested	Expected Result	Passed/Fail
1	User name is incorrect.	Message Displayed: No user found.	P
2	User name is Empty.	Message Displayed: Please fill out this field	P
3	Password is incorrect	Message Displayed: Enter correct password	P
4	Password is empty.	Message Displayed: Please fill out this field.	P

Table 10: Test Data 2

Unit Test 3

1. Author : Anirudh Sharma
2. Functionality Tested : Check admin functionalities
3. Test Execution Procedure : Manual
4. Test Results Checking Method : Manual

Sl: No:	Condition to be tested	Expected Result	Pass/Fail
1	Empty message stored	Message Displayed: Field cannot be empty	P
2	Category Name is Empty.	Message Displayed: Stored Field cannot be empty.	P
3	Edit name and put blank spaces	Message Displayed: Field cannot be empty.	P
4	Delete category which is already existing	Message Displayed: Stored Cannot be deleted, foreign key reference.	P

Table 11: Test Data 3

UNIT TEST 4

1. Author : Muskan
2. Functionality Tested : Check Feedback annotations
3. Test Execution Procedure : Manual
4. Test Results Checking Method : Manual

Sl: No:	Condition to be tested	Expected Result	Pass/Fail
1	Feedback empty.	Message Displayed: Message cannot be empty	P
2	Email id empty	Message Displayed: Field cannot be empty	P
3	Email format is incorrect	Message Displayed: Invalid email id entered	P
4	Name empty.	Message Displayed: Name cannot be empty	P

Table 12: Test Data 4

UNIT TEST 5

- | | |
|---------------------------------|--------------------------------|
| 1. Author | : Ayushmat Soni |
| 2. Functionality Tested | : Payment portal verifications |
| 3. Test Execution Procedure | : Manual |
| 4. Test Execution Procedure | : Manual |
| 5. Test Results Checking Method | : Manual |

Sl: No:	Condition to be tested	Expected Result	Pass/Fail
1	Card number is incorrect	Message Displayed : Invalid card number.	P

2	Card number empty.	Message Displayed: Please enter the card number	P
3	Expiry is incorrect	Message Displayed: Please enter valid expiry date.	P
4	CVV is incorrect	Message Displayed: Card not found.	P
5	CVV left empty.	Message Displayed: Field cannot be empty	P
6	Insufficient balance in a valid card.	Message Displayed: Insufficient balance.	P

Table 13: Test Data 5

Chapter 5: CONCLUSION

5.1 CONCLUSION:

The Web Application was according to the need of our Client/Mentor. All the client requirements were fulfilled and were working properly in the intranet environment. The functionalities were meticulously tested in different erroneous condition. All the unanticipated situations, wrong inputs, application failure were handled in the debugging portion. User can access all the facilities provided in the Web Application if he's a verified user and has paid for the subscription. We followed the same model which is used by Companies like- NETFLIX, PRIME and iMUSIC. This is a prepaid service in which you pay first and then get services. Our web application can be made more attractive by adding additional functionalities with message handling. The main advantage with this application is that it works better with a bigger database as it can give more choices to user according to his query.

5.2 FUTURE SCOPE:

This web application was made according to the requirement provided by our Client/Mentor. However, there is a lot of future work that can be done on it. The use of growing technology like machine learning and artificial intelligence can be used to generate new messages every time user searches something.

Instead of storing messages in the database and updating the database everytime for adding new message, artificial intelligence along with machine learning algorithms can generate logically and grammatically correct messages according to the user's query. The most common advantage of Artificial Intelligence is speech recognition which is popularly seen in Siri of iOS.

Other than that there is a plethora of other advantages. Same kind of technology can be used to generate messages.

We can also include a function that automatically sends email, text message, WhatsApp message etc. to a receiver on a predefined date. We can include the functionality to send the selected message directly as tweet, Facebook post or Instagram story.

Further work that can be done to improve the web application include:

- 1) Using better algorithms for searching the message.
- 2) Artificial intelligence for generating new message instead of giving suggestions from previously stored messages.
- 3) Use of machine learning to predict what the user expect from the application.
- 4) Better user interface can be included to make the application more interactive.
- 5) Better templates and better customization options for user.

REFERENCES

Books

- [1] Professional ASP.NET MVC 5
- [2] Book by Brad Wilson, David Matson, Jon Galloway, and K. Scott Allen

Website

- [3] <https://docs.microsoft.com/en-us/dotnet/index>
- [4] <https://docs.microsoft.com/en-us/aspnet/mvc/mvc5>
- [5] <https://www.tutorialspoint.com/csharp/>
- [6] <https://www.microsoft.com/en-us/sql-server/sql-server-2016>
- [7] <http://stackoverflow.com/>
- [8] <https://lex.infosysapps.com/login>