https://github.com/kumar-shashank/online-grocery-shopping

INTRODUCTION

1. Introduction b. Requirement Analysis c Software Requirement Specification d. Analysis and Design, e. Implementation f. Testing

**CHAPTER 1**

**INTRODUCTION**

**Title of the project:**

**G-UNIVERSE**

**Description of the Project**

G-Universe is a web based project and it is about super market. Here we have admin who can add new items to the list and can update the details of the products and also he can view the orders placed by the users. The user get registered and then logs in and view the products and if he wishes to buy the product ,he can buy the product or add the product to the cart so he can buy it later the client orders, and the last one is for customers, in which he can order the items, look for his order details, and can choose option like home delivery or self pickup.

## from the images and also provides an option to compile extracted code. Our web application aims to help user understand the programming code that are present in their physical documents by taking a picture of the document and uploading it. Our service will be available to the users at all times.

## According to the research undertaken there are many web applications that provide the service for compiling the code. But there are very few or no online compiler that take images containing code as an input, generally users manually type the code and compile.

## In our application, the task of manually typing the code is removed. Our application will apply OCR technique on the image and also provides better user experience for compiling.

**CHAPTER 2**

**REQUIREMENT ANALYSIS**

**2.1 EXISTING AND PROPOSED SYSTEM**

**2.1.1 EXISTING SYSTEM**

The project focuses on buying grocery items online. In previous days some of the sites where introduced for online grocery shopping which did not provide 24\*7 services and also had limited number of items in their collection. So the online grocery shopping was not effective.

**2.1.2 PROPOSED SYSTEM**

In this project, the user will view the items from the collection and if he wants to buy, he will buy it by adding it to the cart and then makes the payment in two ways either Online Payment or Cash on Delivery. If he wish to buy the product later he will add it to the cart.

**Advantages:**

* Saves a lot of time to the customer by providing the required items at one place and also by eliminating the travel time.
* User can easily purchase their grocery.
* Provides home delivery facility.

**2.2 TOOLS AND TECHNOLOGIES USED**

* **VISUAL STUDIO CODE:**

VS Code is Microsoft visual studio is an IDE used to develop console. It is a good editor which provides languages like HTML, CSS, Bootstrap and many of languages to write the code which is useful to the developer. In our project we have used this platform for developing our project.

* **WEB BROWSERS:**

Web browsers are used to view the websites and we can retrieve required information from it. We have used browsers like Google Chrome, Mozilla Firefox etc.

* **APACHE SERVER:**

Apache will accept requests from the users and will provide the sufficient information which they have requested in the form of webpages. It is a open-source web server through which we will host our website.

* **MYSQL:**

MYSQL is used to create databases. We have used MYSQL by which we can easily create and handle databases.

* **XAMPP:**

It is free and open-source platform web server using this we can host our website.

**Software/Hardware Requirements:**

**Software Requirements:**

Operating System: Windows7 or above

Front End : Html, CSS, JS

Back End : MySQL

Tools :VS Code, APACHE, MYSQL

**Hardware Requirements:**

RAM :2GB or Above

Input Devices : Keyboard, Mouse

Output Devices: Monitor

Processor :i3 and above

Hard Disk :40GB

**Objectives:**

* Single site for all of them.
* Correct Analysis of the daily market.
* Daily Analysis of the Profit or Loss.
* To help the customers to order their needs easily.

**CHAPTER 3**

**SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 INTRODUCTION**

A software requirements specification (SRS) is a document that provides complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

**3.2 FUNCTIONAL REQUIREMENTS**

* **Admin:**

Admin is the person who manage the entire application. Admin has the authority to add new items, remove items and also can edit the details

of the items whenever required.

* **User:**

The users are the customers who register into the application and logs in to the application and views the grocery, orders the grocery and make payment.

**3.3 NON-FUNCTIONAL REQUIREMENTS**

The applicability of a structure or one of its components is described by a practical needs record. It also depends on the type of programming, target audience, and platform on which the product is used. While useful client requirements may be high-level declarations of what the framework should accomplish, useful framework requirements should also clearly depict what the structure should do.

* **Portability:**

This System uses simple programming languages and can run on any system

* Usability:

This system is easy to use and it has user friendly environment. It has Easily understandable interfaces

* Reliability:

Reliability is related to the performing ability of the system. We will full fill the task in given environment by evaluating the situation.

* Scalability:

We are going to develop this software which is scalable.

* Security:

System will use secure database and users can just edit/ read information but they cannot edit or modify anything except their personal information.

* Performance:

Any number of users can access to the system at any time and server will be working 24X7 times.

* Maintainability:

Maintainability is the effortlessness with which a thing can be monitored everything.

* Natural Interface:

If the new update is done, it will not effect to the user. They can easily adjust to the new system.

**CHAPTER 4**

**ANALYSIS AND DESIGN**

**4.1 CONTEXT FLOW DIAGRAM(CFD)**

Context flow diagram is the Top-level data flow diagram (0 Level). It only contains the one process node that generalizes the functions of the entire system in relationship to external entities. So with the help of below the CFD diagram shows how the process of our project contains.

**4.2 DATA FLOW DIAGRAM**

Data flow diagram (DFD) maps out the flow of the information for any process or system. With the help of this we can get to know what is the flow of our project and how it will be work. It shows the Input, output and each working flow of the project. And data flow diagram is shown below:

**4.2.1 DFD LEVEL 1**

**4.2.2 DFD LEVEL 2 ADMIN**

**4.2.3 DFD LEVEL 2 USER**

**4.3 DATABASE DESIGN**

Database design is the organization of data according to database model. In this project, database determines what data is stored and how they relate. Using this database, we performed database model. Also contains the logical and physical storage and design and also the information of the project data stored here.

**4.4 ER-DIAGRAM**

**4.5 SCHEMA INFORMATION**

A schema is pictorial representation of the relationship between the tables in the database created.

**4.5.1 TABLE DEFINATION**

**CHAPTER 5**

**IMPLEMENTATION**

**CHAPTER 6**

**TESTING**

**6.1 Test Cases**

* **Home Page**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO** | **ACTION** | **INPUTS** | **EXCEPTED OUTPUT** | **OBSERVED OUTPUT** |
| 1 | If on load of page | Loaded | Displays home page and available fields | Successful |
| 2 | On clicking available fields | Selected | Entered to the field | Successful |

* **User Registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO** | **ACTION** | **INPUTS** | **EXCEPTED OUTPUT** | **OBSERVED OUTPUT** |
| 1 | If user clicks on  Register | Inputs are not given | Displays registration form | Successful |
| 2 | User enter the correct details of the registration form | Inputs are  are completely given | Successfully registered | Successful |
| 3 | If any fields are blank | Inputs are not given correctly | Please enter required field | Successful |

* **User Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO** | **ACTION** | **INPUTS** | **EXCEPTED OUTPUT** | **OBSERVED OUTPUT** |
| 1 | If user clicks on  Login button | Inputs are not given | Displays login form | Successful |
| 2 | when the user clicks on login with valid username and password | Inputs are correct | Login Successful | Successful |
| 3 | If username is blank but password is entered | Username is not given | Please enter username | Successful |
| 4 | If username is entered but password is not entered | Password is not given | Please enter password | Successful |
| 5 | If the both username and password incorrect | Incorrect username and password given | Invalid username or password | Successful |

* **Admin Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL.NO** | **ACTION** | **INPUTS** | **EXCEPTED OUTPUT** | **OBSERVED OUTPUT** |
| 1 | If admin clicks on Login button | Inputs are not given | Displays login form | Successful |
| 2 | when the admin clicks on login with valid username and password | Inputs are correct | Login Successful | Successful |
| 3 | If admin username is blank but password is entered | Username is not given | Please enter username | Successful |
| 4 | If username is entered but password is not entered | Password is not given | Please enter password | Successful |
| 5 | If the both username and password incorrect | Incorrect username and password given | Invalid username or password | Successful |

**CONCLUSION**