**Software Requirements Specification Template**

**EMart**

Software Requirements Specification

Version 1.1

06/30/2019

Jyoshna Boppidi

Sai Prakash Reddy Mamidi

Ramesh Nutulapathi

Havya Ravipati

Sai Sri Lakshmi Vancha

Aakash Valluru

Venkata Sai Krishna Dasari

**Submitted in partial fulfilment**

**Of the requirements of**

**CSIS 44-691 Graduate Directed Project 1**

# 

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 6/11/2019 | Version 1.0 | Havya Ravipati  Jyoshna Boppidi  Aakash Valluru  Ramesh Nutulapathi  Sai Prakash Reddy Mamidi  SaiSriLakshmiVancha Venkata Sai Krishna Dasari | First Revision |
| 6/30/2019 | Version 1.1 | Havya Ravipati | Second Revision |
|  |  |  |  |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents Page Number**

1.Introduction

1.1. Purpose

1.2. Scope

1.3. Definitions, Acronyms, and Abbreviations

1.4. References

1.5. Overview

2.General Description

2.1. Product Perspective

2.2. Product Functions

2.3. User Characteristics

2.4. General Constraints

2.5. Assumptions and Dependencies

3.Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

3.1.2. Hardware Interfaces

3.1.3. Software Interfaces

3.1.4. Communications Interface

3.2. Functional Requirements

3.3. Use Cases

3.4. Class/Objects

3.5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

3.5.3. Availability

3.5.4. Security

3.5.5. Portability

3.6. Inverse Requirements

3.7. Design Constraints

3.8. Logical Database Requirements

3.9. Other Requirements

3.10. Prototypes (for complete project)

3.11. Use Case Diagrams

4. Design

4.1. ER diagram

4.2. GUI

5.Analysis Models

4.1. Data Flow Diagram

4.2. Sequence Diagram

**1.Introduction**

**1.1 Purpose:**

EMart is an ecommerce website to shop for various products. We are using HTML5, JavaScript, CSS, and Oracle to develop the project. This website reduces the stress on the customers by providing a good user interface. The payment is done securely using a real time system using PayPal or google pay.

**1.2 Scope:**

|  |  |
| --- | --- |
|  | **Scope Statement** |
| **Project Title** | EMart |
| **Date** | 6/11/2019 |
| **Project Description** | This is an interactive ecommerce website which makes the effort of shopping simple to the users. By using this web application, users should be able to buy products from the online store. The customers should be able to track their products and they should be able to Add/Remove the products from the cart without any trouble. |
| **Requirements and Functions** | * The User Interface should be simple to use. * The website is supposed to have a login/register page. * There should be an option for guest checkout. * There should be a search tab where the user can shortlist the desired product by typing in the key words. * There should be a Homepage where all the products are supposed to be listed. |
| **Deliverables** | Primary Deliverable: A prototype i.e. static to show how the web application looks like.  Final Deliverable:A website with Homepage and login/register page working as stated in the requirements. |
| **Future Scope** | At a later point, This website will be added with pages like My Cart, Orders etc. There is a plan to implement some more features where the user can filter the desired product by adjusting the price, color or gender etc. |

**1.3. Definitions, Acronyms, and Abbreviations:**

**1.4. References:**

**1.5. Overview:**

**2. General Description**

**2.1 Product Perspective:**

EMart is a one stop website application for purchasing products online. The main features of this application is:

* Login & Signup: User can register to the site and create an account at ease.
* Account Profile: User can check order history and track packages.
* Support Page: Can easily reach customer service for any assistance related to the product.
* Ratings: User can compare ratings and reviews of each product before purchase.

**2.2 Product Functions:**

Using this webpage, a user will be navigated to different departments where user gets an experienceto visit different products in each categorized department. They get an option to add to cart or buy now button.

**2.3 User Characteristics**

A user can review the products and have a glance at frequently asked questions and can purchase a product even without signing up.

**2.4 General Constraints**

Extraction of data from the database server might delay the response time depending on the internet service provider.

**2.5 Assumptions and Dependencies**

We assume that the applications will be suitable on all the devices with any type of browser installed in it.

**3.Specific Requirements**

**3.1. External Interface Requirements**

**3.1.1. User Interfaces:**

The website will be using the Menu driven Interfaces where the customer can choose a category from the menu bar, navigation bar and other options that will be provided on the web page.

**3.1.2. Hardware Interfaces:**

This website should be able to work on any device with an appropriate browser.

**3.1.3. Software Interfaces:**

* Visual Studio Code
* Oracle

**3.1.4. Communications Interface:**

We are using Skype, Emails for communication interface.

**3.2 Functional Requirements:**

**Login Page:** This page should display two text fields with Username and password. There should a feature Forgot Password which helps the users to recover the password.

**Admin Login Page:** The admins and suppliers can login here in order to add the products, review the products they had added and check the orders they had received. The suppliers should register with the admin directly, so there is no register option for suppliers.

**Register Page:** The users without a Username and password must register in this page. This page consists of username, email address, password and contact information.

**Home Page:** The products are supposed to be displayed in this page. There should be an image, price and product description for every item in this page.

**Admin Home Page:** This page is where the admin can see the added products, admin can add the new products from this page. They can review on the received orders and post the status of the orders.

**My Cart Page:** The users can see the products they want to order and add/remove the products from the cart.

**My Profile Page:** The user can view/edit his profile which consists of User Id, addresses, payment information and they can change the password and also the payment information.

**Support/Customer service Page:** The user can easily reach customer service for any assistance related to the product using this page.

**Payment page:** The user should enter the card number, expiration date, CVV etc in this page to complete the order (transaction).

**Menu (Navigation) Bar:** The list of categories are displayed in this bar.

**3.3. Use Cases:**

**3.4. Class/Objects**

**3.5. Non-Functional Requirements**

**3.5.1. Performance:**

**Web performance** is the speed that web pages are loaded onto a client’s web browser. In Emart website customer can login, signup and view products in less than seconds. The website will be maintaining high performance all the time.

**3.5.2. Reliability:**

EMart is a reliable website where all the ACID rules are applied in order to provide a reliable website for the users.

**3.5.3. Availability:**

Ensure that the website is available all the time for the user to view the products and place the order.

**3.5.4. Security:**

The website ensures that the payment information and the personal information of the customer is safe and secured.

**3.5.5. Portability:**

The website can be used on multiple platforms or instantly accessed from any device with proper internet and software.

**3.6. Inverse Requirements**

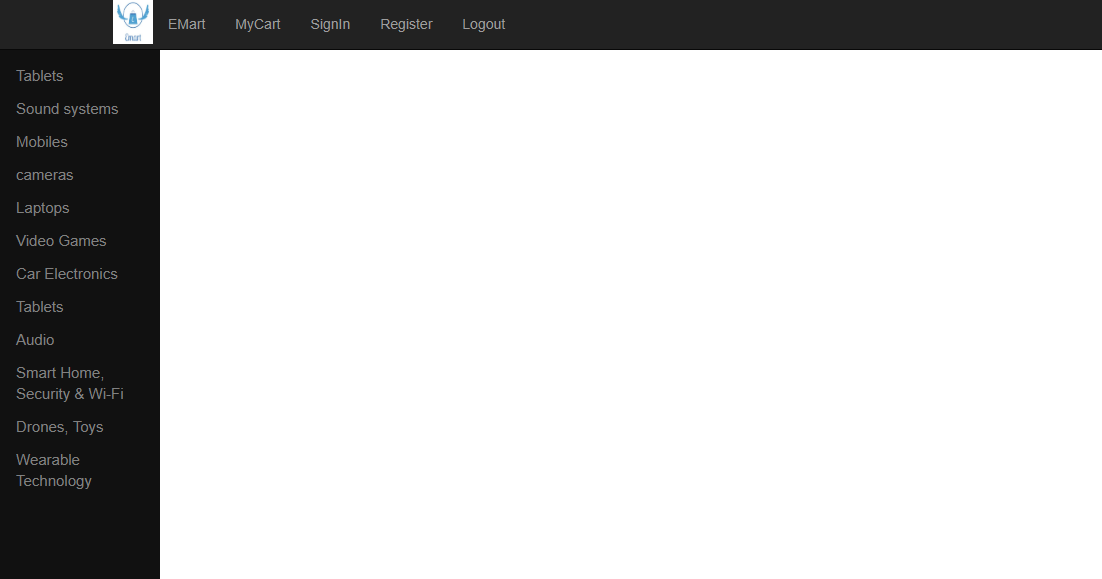
**3.7. Design Constraints**

**3.8. Logical Database Requirements**

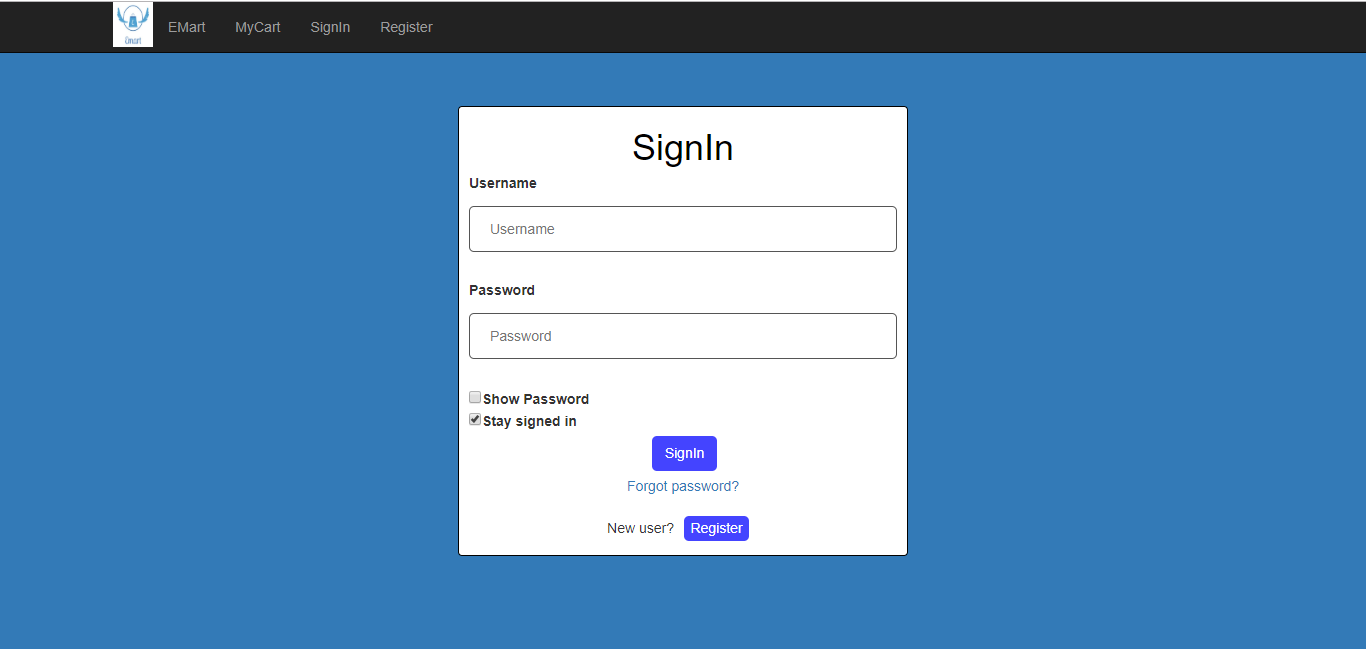
**3.9. Other Requirements**

**3.10. Prototypes (for complete project)**

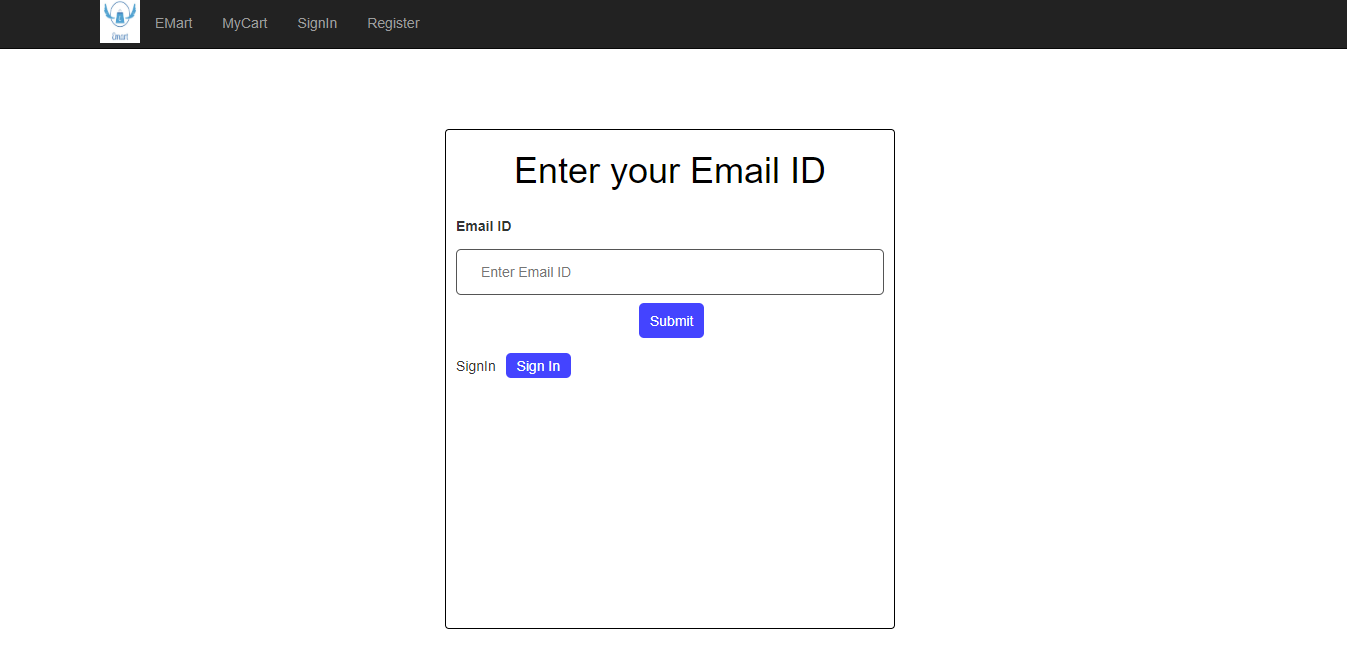
**Home page:  In this page** different categories of the products are displayed and there should be an image, price and product description for every item.

****

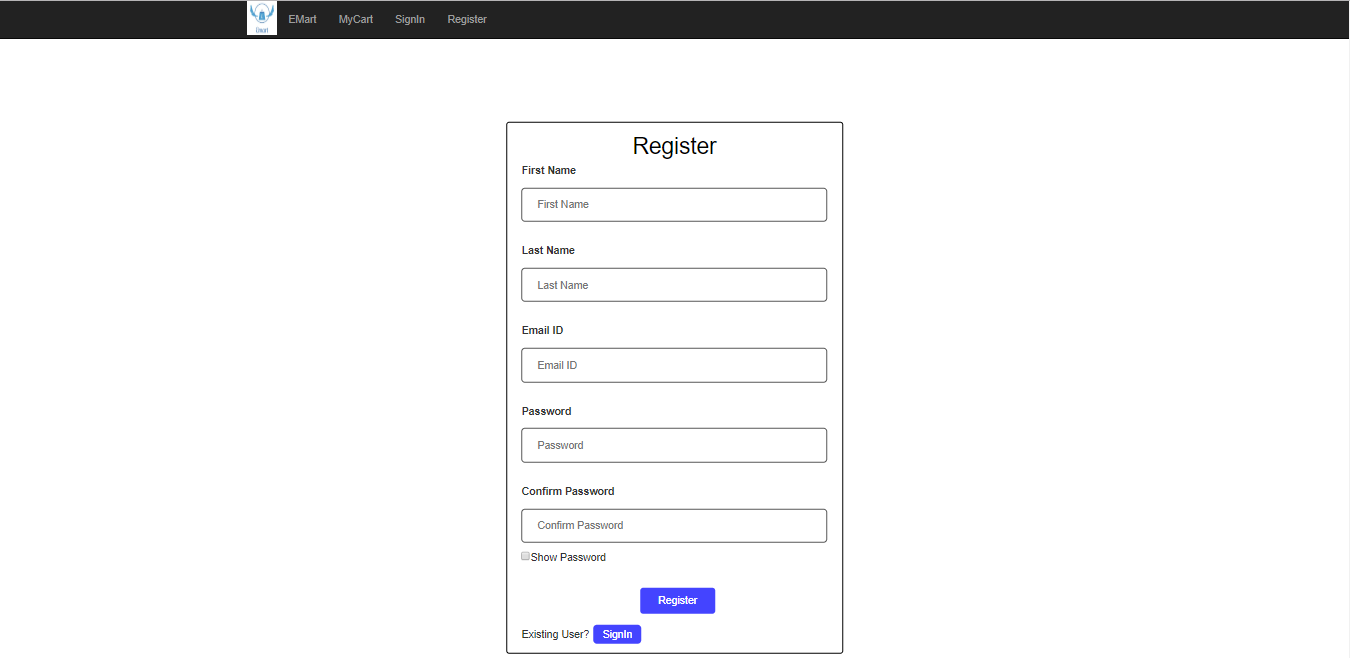
**Login/Register page:**  In this page the user can see the two text fields with Username and password.

****

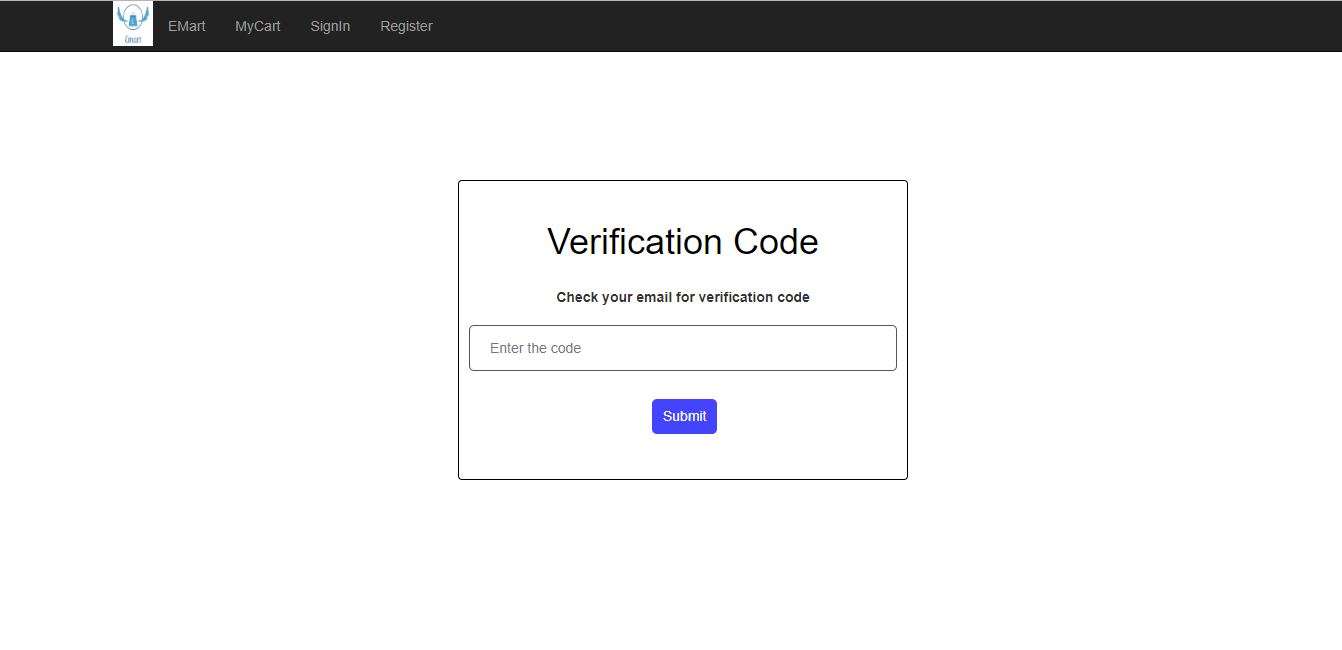
**Forgot password:** In this page the user can see the forgot password feature which helps the users to recover the password.



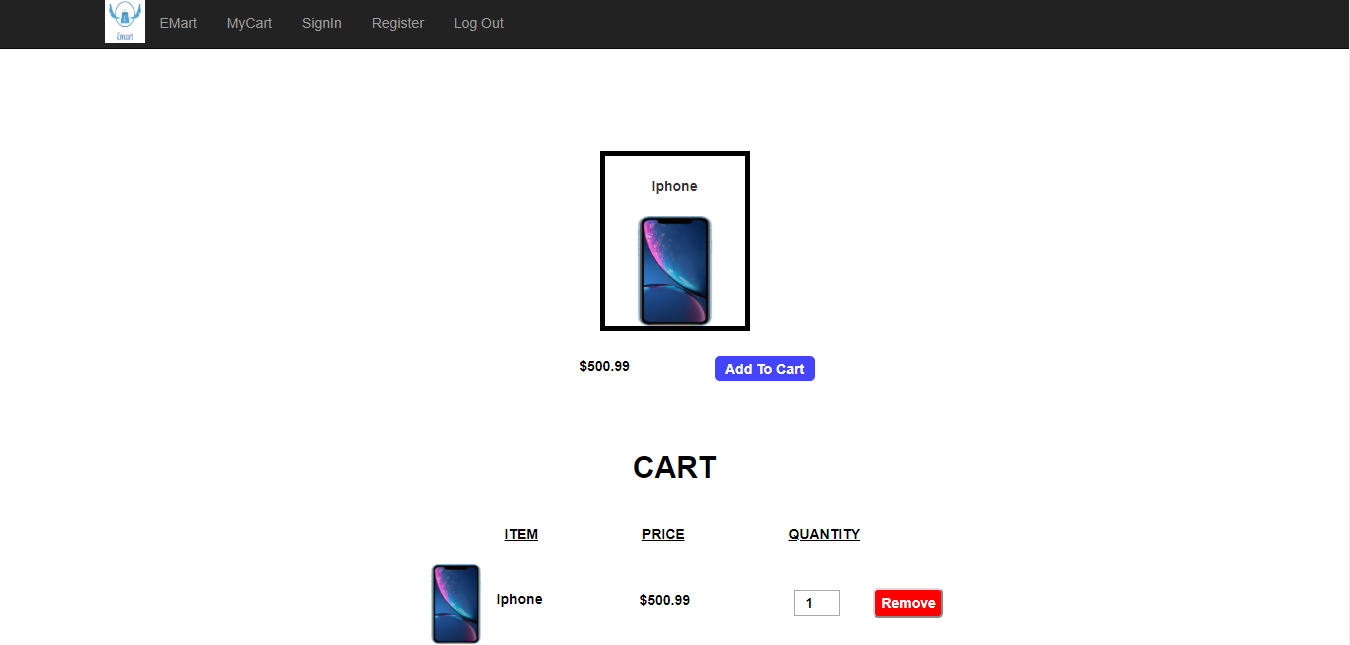
**Registration page:** In this pagethe users without a Username and password must register in this page. This page consists of username, email address, password and contact information.



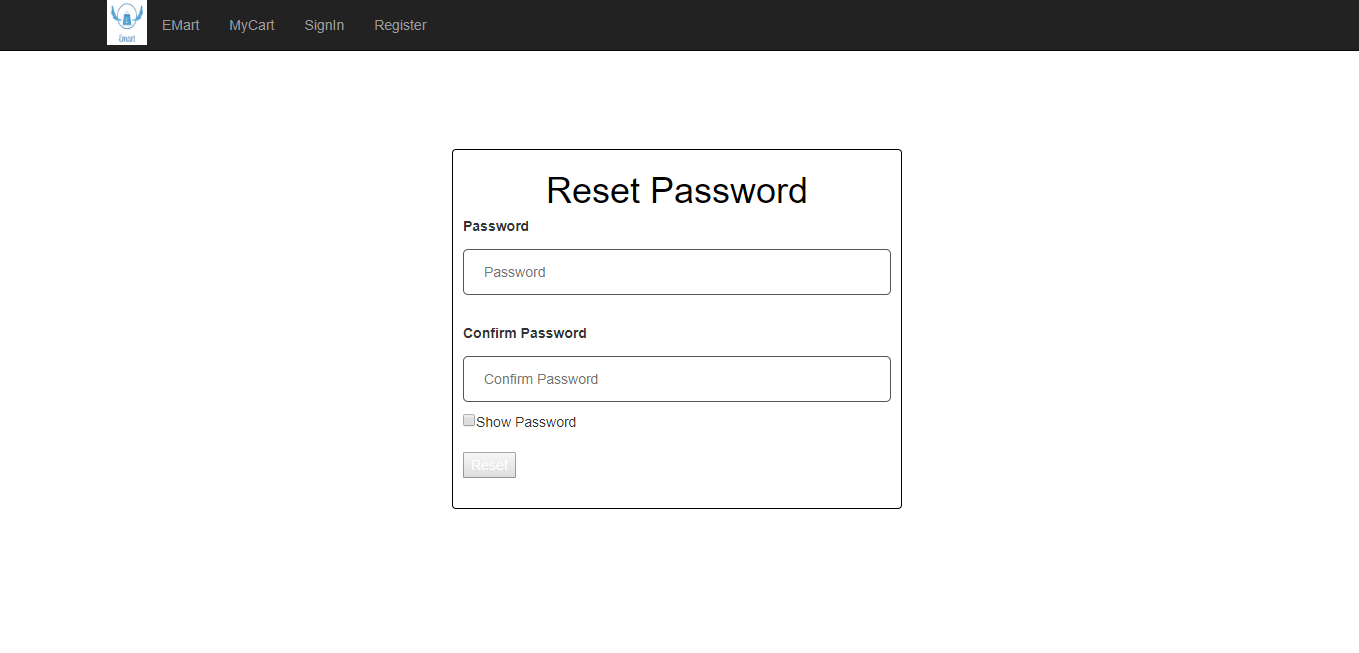
**Verification code:** In this pagethe users will get the verification code while doing the registration process to the website.

****

**My Cart Page:** In this page the users can see the products they want to order and add/remove the products from the cart.



**Reset Password Page:** In this page user can reset the password by entering the text fields password and confirm password.



**3.11. Use Case Diagrams**

**4. Initial Draft Of Design:**

**4.1. ER Diagram:**

Our ER Diagram has 9 entities with 2 associative entities.

In the Customer entity, the information about the customer such as name, address, phone number etc. are stored.

The Seller entity is a weak entity and used to store the seller ID, first and last name, company, email, phone number, address, city, state, zipcode and country.

In this ER diagram product seller is associative entity between the product and seller. It stores seller ID, product ID, manufacture date, quantity on hand. Here the seller ID, product ID acts as both primary key and foreign key.

The order entity is used to store the order ID, product ID, Customer ID, order date, delivery date and payment ID. Here the order ID plays as a primary key in order to make the orders unique.

The product entity stores the product ID as primary key, seller ID, category ID as foreign keys, name, price and description of the product.

Here, the customer payment details is a weak entity which depends on the customer entity because the customer payment details exist only when a customer exists. Similarly, payment is also a weak entity because the payment only exists when there is an order. The product seller and orderline are considered as weak as well as associative entities.

In the below diagram, there is a mandatory one to optional many relationship between the customer and the order because a customer can place any or none number of order but the order should be placed by exactly one customer. The relationship between the order and the product is considered many too many because a single order must consist of at least one product or many products whereas the single product can be in multiple orders or may not be in any order. A seller can add multiple products or no products at all whereas the product can be sold by one or many sellers. Therefore, the relationship between the seller and product is many to many. A product should belong to exactly one category whereas the category can no products or multiple products. So, the relationship was considered as mandatory one to optional many.

****

**Mockups:**

**Home Page:**

****

**My Cart page:**

****

**User Product Page:**



**Admin Product Page:**



**Technical Manual:**

**Product Name:** EMart (Ecommerce website)

**Product Description:** This website has all the features that are generally found in an ecommerce website. They include home page, my cart page, product page, payment page, login page, register page etc.

**Cloud Repository:** The website is being built using the GitHub repository. The repository consists of codes, documentation and readme.md. The codes folder consists of the actual codes that are required for the website. These codes have been developed using various to tools. The documentation folder consists of the various documents that help to check the progress of the project.

Link to the cloud repository: <https://github.com/prakashreddy-97/EMart>

**Planned objectives:**

Phase-1: This phase consists of implementation of the login and register pages.

|  |  |
| --- | --- |
| **Objectives** | **Status** |
| Decide the name of the product | Done |
| Gathering the requirements for the entire product | Done |
| Build the ER diagram for the entire website | Done |
| Develop prototypes | Done |
| Implementation of the frontend part of the website with login and register pages | Done |
| Implementation of the backend part for the login and register pages | Done |
| Connect the frontend and backend for the login/register pages | Done |
| Implementation of Email verification for security purposes | In progress |
| Implementation of password recovery | In progress |

**Tools Used:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools** | **Front-ware** | **Middle-ware** | **Back-ware** |
| HTML5 | ✓ |  |  |
| CSS4 | ✓ |  |  |
| JavaScript | ✓ |  |  |
| PHP |  | ✓ |  |
| MySQL |  |  | ✓ |

**Front-end Development:**

In the GitHub EMart Repository, there are several files including .html, .css & .js as their extensions. All these folders have been created and developed by the frontend developers.

This website was made responsive using bootstrap.

Login page:

In the above image, there are mainly two fields where the user can enter the username and password. By default, the username is the email address of the user. The fields’ username and passwords are given as inputs in the login.html. This page also consists of Sign-In and Register buttons where the sign-in button takes the user to the home page of the website (if the user name and password are correct). If there is a new user, the user can register by simply clicking on the register button. The user can’t leave any of the above mentioned text fields empty. If the user leaves these text fields empty, there would be an alert stating that he has to fill all the fields.

login.html: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/login.html>

login.css: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/login.css>



Register page:

In the below image, The user enter the first name, last name, Email Id, Password, Confirm Password. All these above fields are given as inputs in the register.html code. The fields first name, last name, Email id, password and confirm password are given as inputs in register.html. There are certain validations made in the register.html to make the user account secure.

If the user leaves these text fields empty, there would be an alert stating that he has to fill all the fields. The password requires special characters, numerical value, uppercase alphabet and at least one lowercase alphabet. If the user enter different values in password and confirm password, the user may not be able to register.

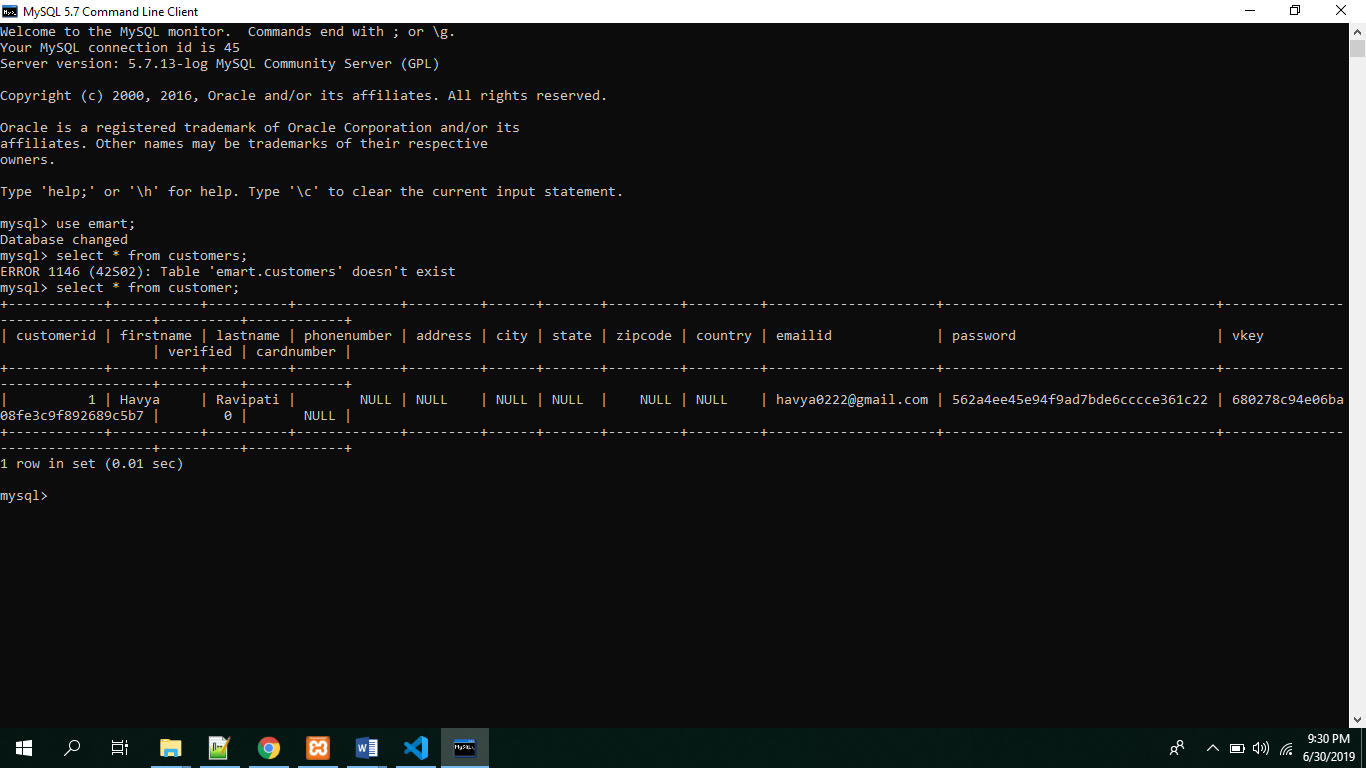
register.html: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/register.html>

register.css: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/register.html>



**Back-end Development:**

As per the ER Diagram, A database has been created with the name ‘Emart’. For now, this database consists of one table with the name customer. The details of the user are stored in this table. This table consists of the users first name, last name, phone number, address, city, state, zip, payment details, password (in encrypted format) and a verification key (in encrypted format). So, whenever a person clicks on the register button in the front end. The details provided are supposed to be stored in this table.



**Middleware:**

There are several folders in the repository with .php as extension. The website has PHP as a connection between the front end and the backend.

register.php: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/registration.php>

connect.php: <https://github.com/prakashreddy-97/EMart/blob/master/Codes/connect.php>

Register.php helps to establish a connection between the register.html page and the emart database. Similarly, connect.php helps to retrieve the stored data during the process of login.