Software Requirements Specification

for

**Electronics Mart**

Version 1.0 approved

**Prepared by Mohit, Namit, Manish, Deepanshu, Pearl**

**NCU**

**02-02-2022**

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

# Introduction

## Purpose

Electronic Mart is a Web-Application whose requirements are mentioned in this document, this is first version of application. It is a User-Friendly E-commerce website it will enable customers to browse the products efficiently. The administration module will enable system administrators to manage products, orders and users.

## Document Conventions

* Entire document should be justified
* Convention for main title.
* Font Face: Arial
* Font Style: Bold
* Font Size: 32
* Convention for sub title.
* Font Face: Times
* Font Style: Bold
* Font Size: 18
* Convention for body.
* Font Face: Arial
* Font Style: Normal
* Font Size: 11

## Intended Audience and Reading Suggestions

|  |  |
| --- | --- |
| * Developers | * Project Managers |
| * Marketing Staff | * Testers |
| * Documentation Writers | * Users |

This SRS contains description, requirements, features of project. Suggested reading sequence is from description to different types of mentioned requirements and then features.

## Product Scope

* Secure registration & profile management facilities for customers with an additional functionality of 2-factor authentication.
* Adequate searching mechanism for easy & quick access to particular products.
* Creating a Shopping cart so that customers can shop 'n' no. of items and checkout finally with the entire shopping carts. Customers can add or delete items in the cart.
* Regular updates to registered customers through newsletter about Sales, Amazing Deals and New Arrivals.
* Showing the products with customer rating in each category.
* Maintaining database of customers with their interested category of products.
* Administrators are responsible for internal affairs like processing orders, updating order's status and managing products.
* Customers can give ratings & review for the product which they have purchased & used (or delivered to them).
* Adequate payment mechanism and gateway for all popular credit cards, cheques and other relevant payment options.

## References

* Mongo DB – <https://docs.mongodb.com/manual/tutorial/>
* NPM - <https://www.npmjs.com/>
* Heroku - <https://devcenter.heroku.com/articles/deploying-nodejs>
* Node JS - <https://nodejs.org/en/docs/>
* Express JS - <https://expressjs.com/>
* HTML5 – <https://www.w3schools.com/TAGS/default.ASP>
* CSS3 – <https://www.w3schools.com/cssref/>
* JAVA SCRIPT - <https://www.w3schools.com/jsref/default.asp>

# Overall Description

## Product Perspective

Electronic Mart is a Web-Application, it is a User-Friendly E-commerce website it will enable customers to browse the products efficiently. This website is intended to be a new, self-contained product and should depend on the availability of mongo cloud, Heroku servers. It should be able to run on any device which supports browser.

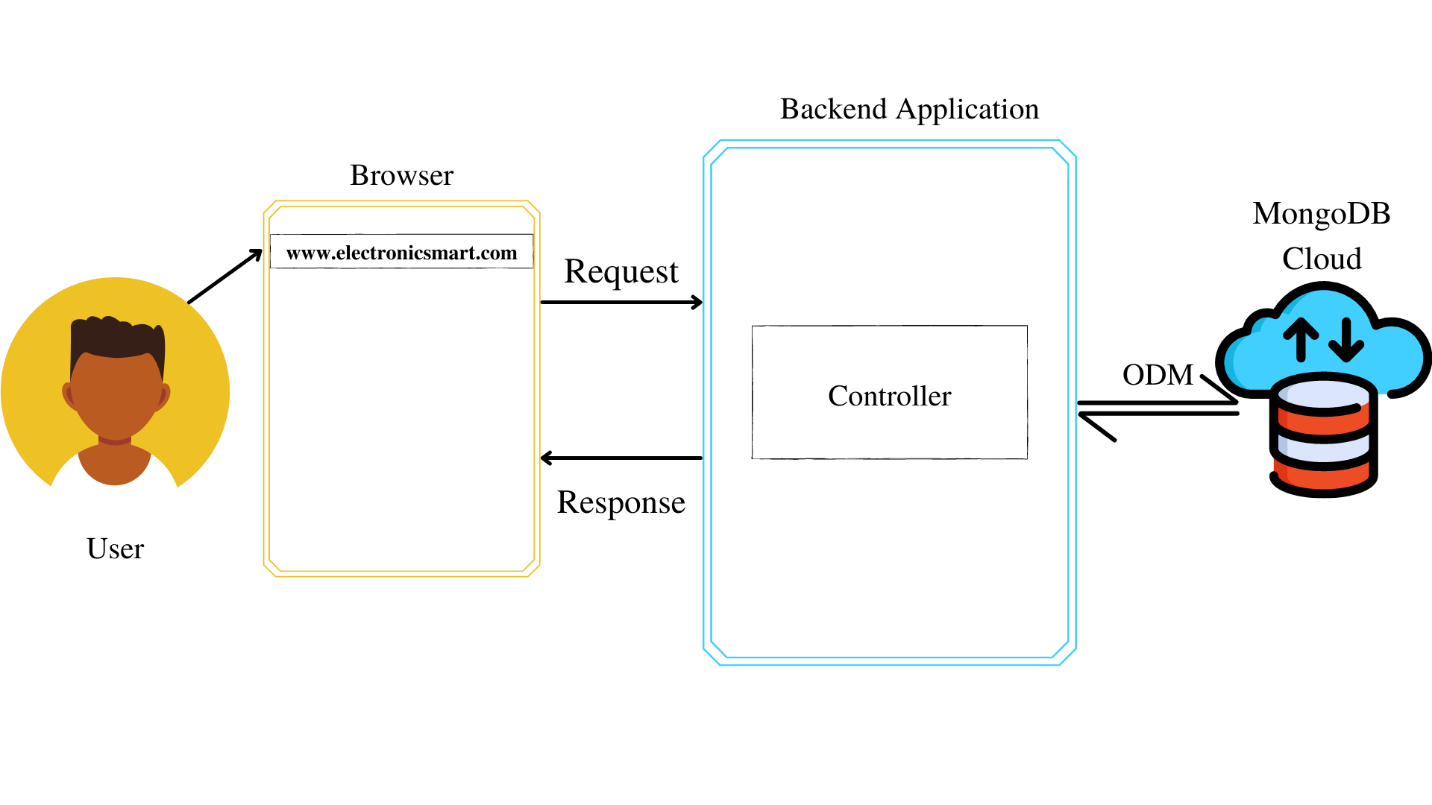


Fig. 2.1

\*ODM – Object Data Manipulation

## Product Functions

* The main purpose of this project is to reduce manual work.
* Function:
* Customers can search products by category, name, ratings and price range.
* After successful login Customers can add & edit items in cart.
* Customers can also view the status of any previous orders & cancel any order that has not been shipped yet.
* Customers can modify personal profile information (such as name, phone No., email & other details) & can enable of disable Two-Factor Authentication for additional security.
* Admins can manage products, orders and users.

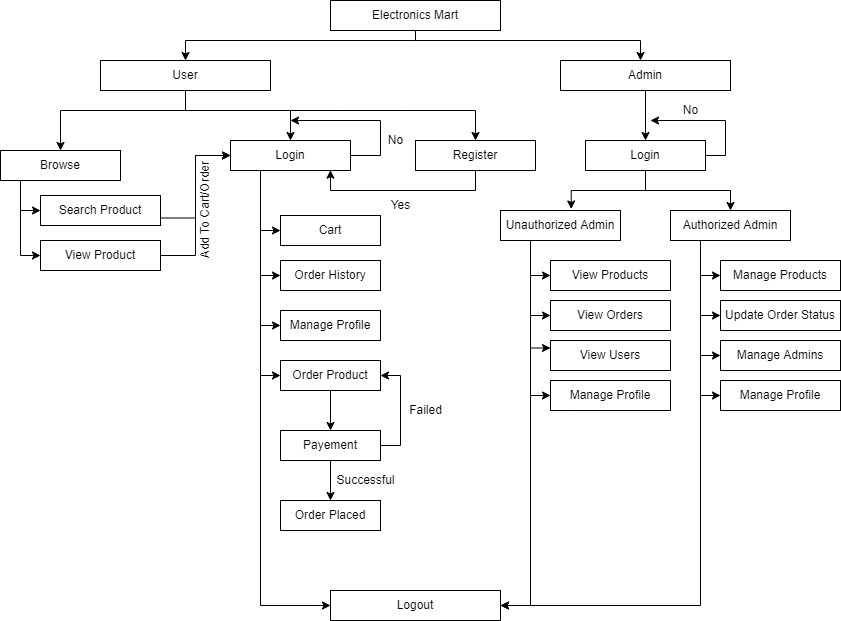


Fig. 2.2

## User Classes and Characteristics

* Customers should be able to search products, manage cart, order products, manage their profile.
* Admins have following management functionalities –
* Admin – view products, users, orders
* Authorized Admins – manage products, orders, users, admins.

## Operating Environment

The product will be operating in any device which supports browser functionality.

Electronics Mart is a website and shall operate in all famous browsers, for a model we are talking about Google Chrome, Microsoft Edge, Mozilla Firefox, etc.

Hardware configuration include necessary requirements to run a browser application in device.

To use website the device should be connected to internet.

## Design and Implementation Constraints

* Website application should be light weight & simple API.
* The database should be able to manage huge amount of data and should have flexible schema with sharding (basically horizontally scalable) so preferred database language is Mongo DB.
* Mongo atlas should be used for storing data on cloud having 10GB storage & 2GB RAM.
* Website should be built under Micro service Architecture.
* The programming language should support Distributed application with many nodes, asynchronous operations (basically I/O) and good in data centric processes.
* Programming language for backend – Node JS and framework – Express JS.
* Programming language for frontend – HTML 5, CSS3, Java Script.
* Project design should be modular with better readability.
* Communication protocols should be in XML documents encoded in UTF−8 and sent using HTTP or HTTPS.
* Security consideration are encryption of password in database, authentication for all users & admins for accessing pages which requires identification, authorization check for admins, two factor authentication check if enabled and secured account recovery options.
* Loading time should ne minimum and for that use AWS servers for deployment of project for best performance. With RAM 2GB, OS Windows 8 64-bit, Hard Disk size of 10GB.
* The application should be able to run on cross platform so CORS should be enabled.
* User & admins have different Front-End applications to perform actions

## User Documentation

The product will include user manual. The manual will include product overview, complete configuration of the software (such as Mongo DB server and other technologies used), technical details, backup procedure & contact information which will include email address. The product will be compatible in any browser and database will be created in Mongo DB.

## Assumptions and Dependencies

The assumptions are –

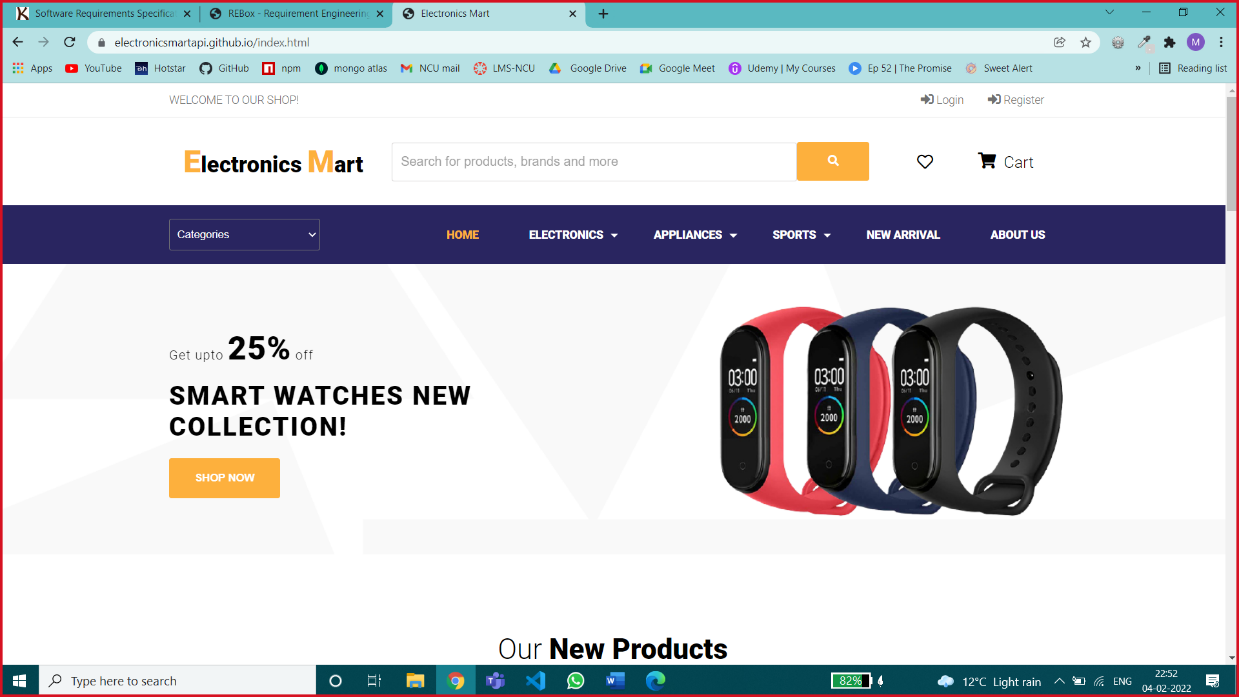
* The coding should be error free.
* Any Front-End request should not give Timeout error or should not have any Internal Server Error.
* The system should be user friendly so that it is easy to use for the users.
* The Electronics Mart is a 24X7 running website except the servers where application & database are deployed & stored goes down.
* User can access Electronics Mart from any device which supports web browser and an internet connection.
* User & admins have their correct usernames & passwords to enter into their online accounts & do actions.

The dependencies are –

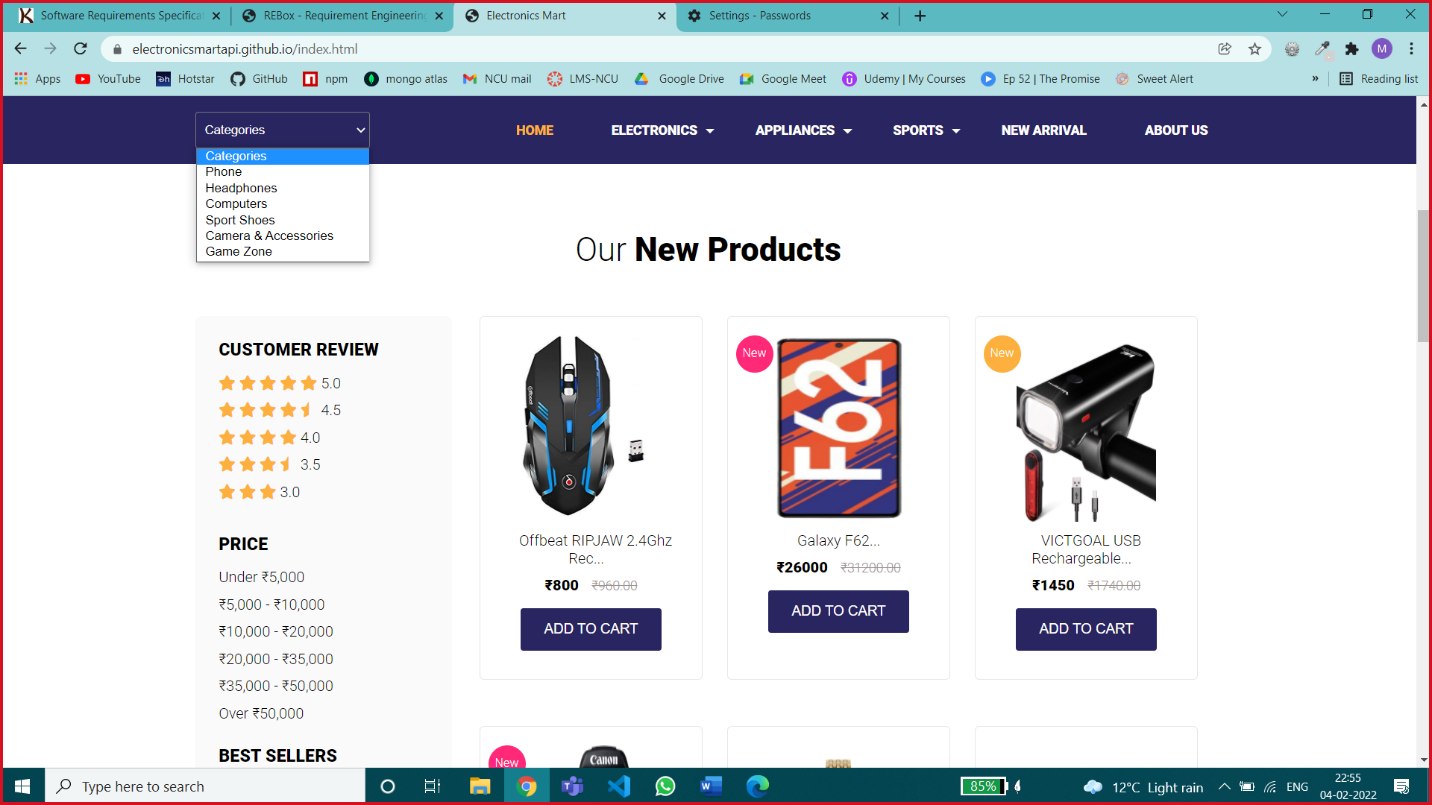
* The Servers due to which the web application will run.
* The Payment process is dependent on the payment plugin while purchasing the product.
* On the basis of listing requirements & specifications the project will be develop & run.
* The admins, Client IT team should have proper understanding to the project.
* The Information of all users must be stored in a database that is accessible by Electronics Mart system.

# External Interface Requirements

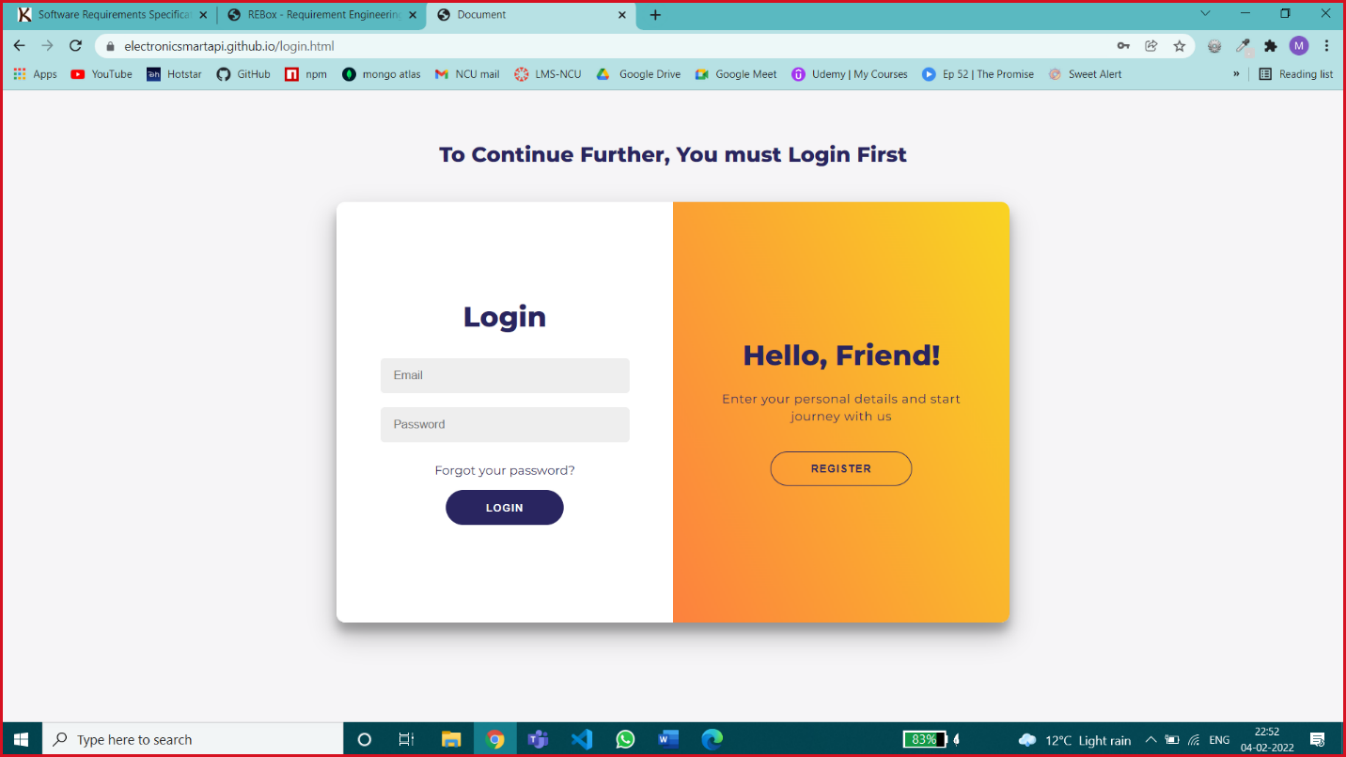
## User Interfaces



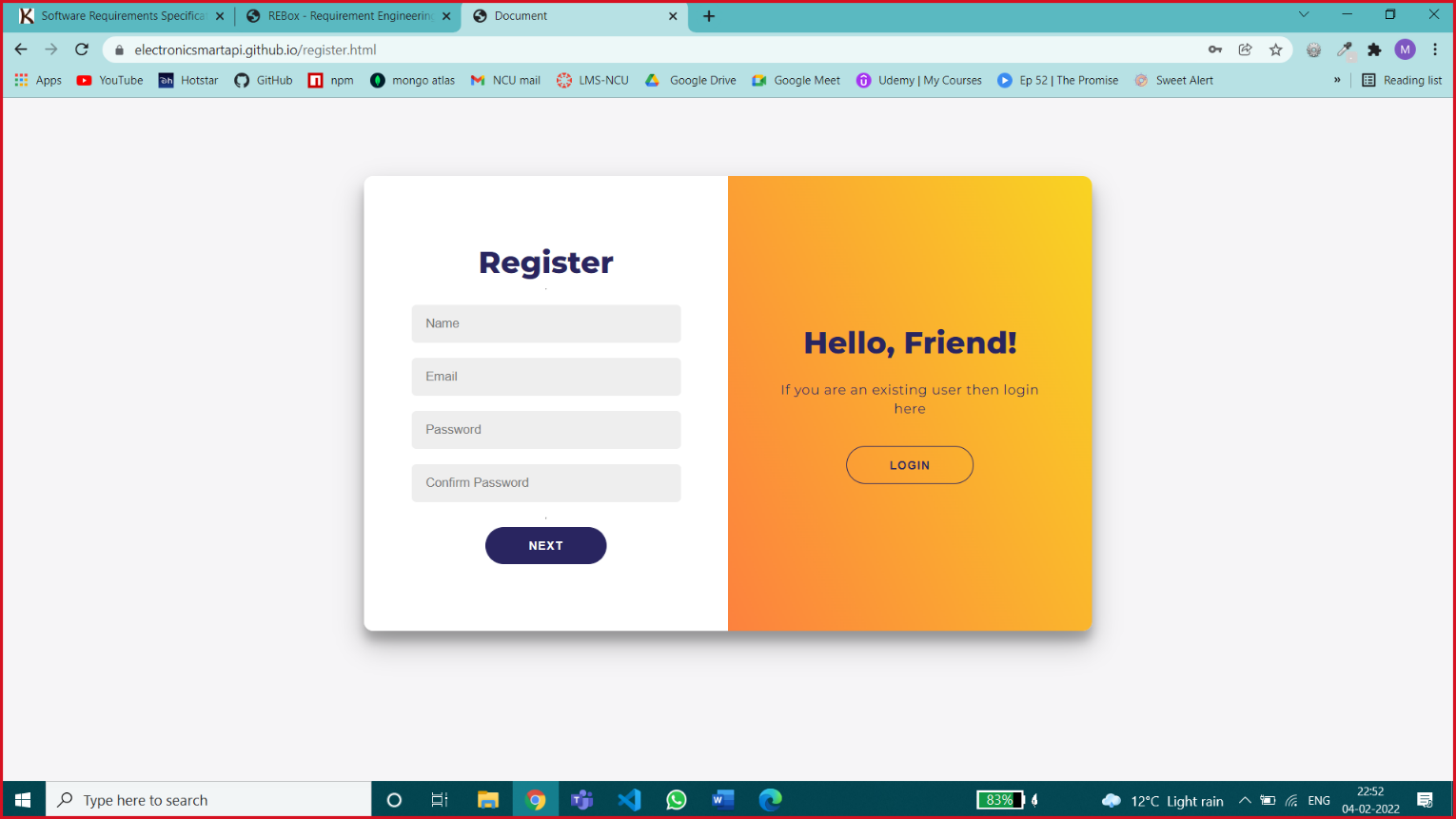
Home Page – Customers can view products on this page.



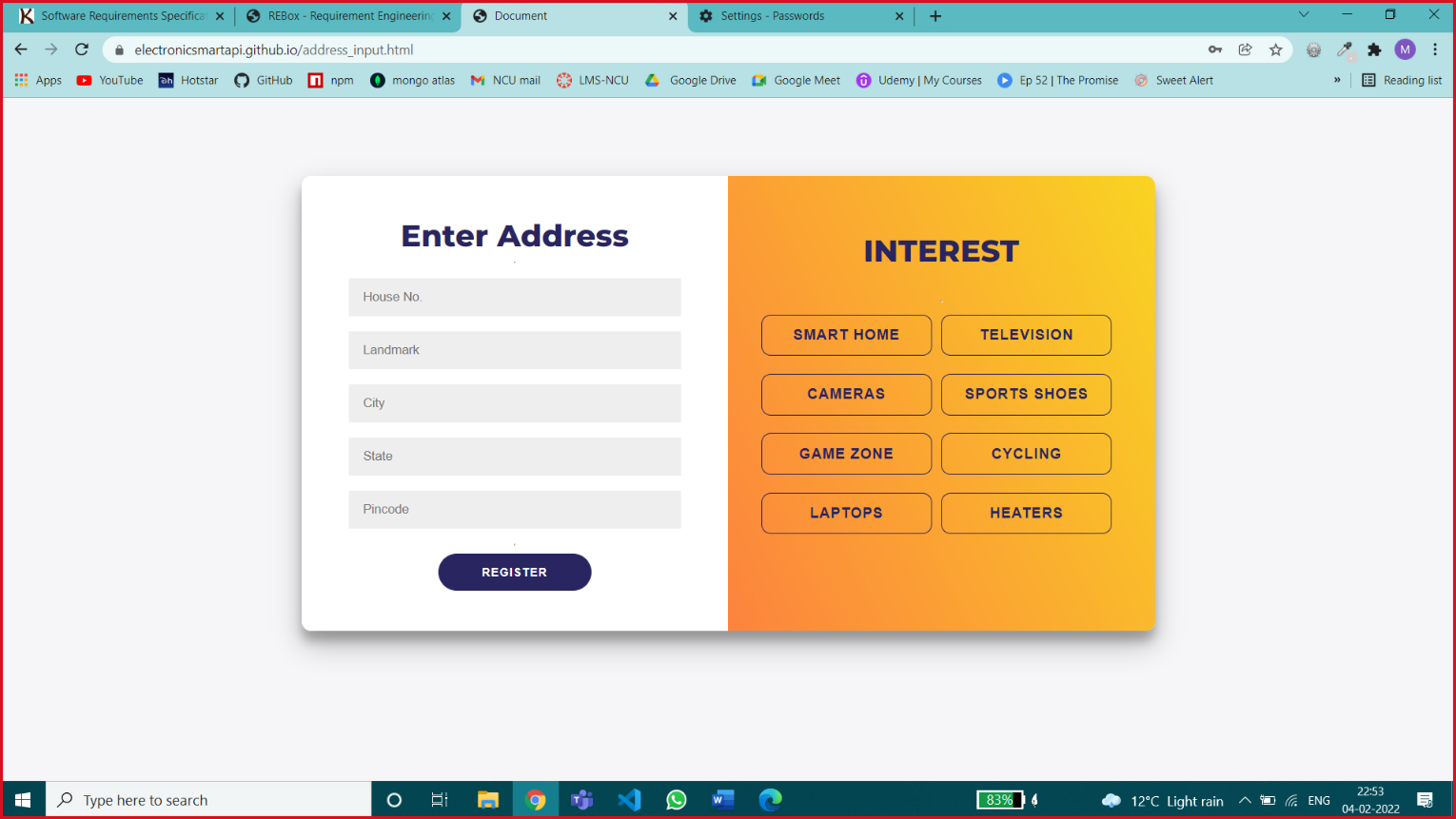
Different Options for Searching Products – Customer can search products by specific category, by Customer Review, by price range, by name.



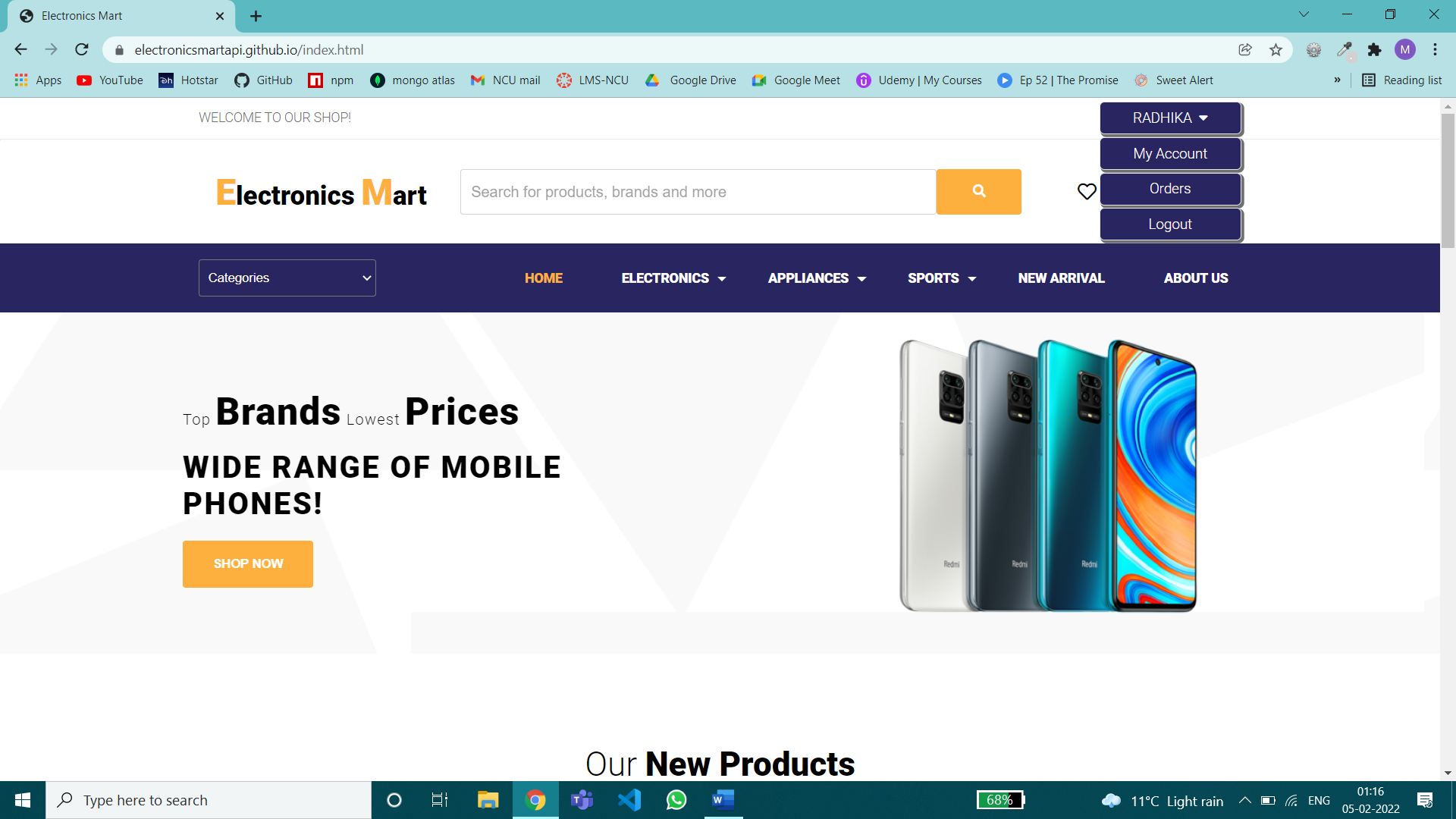
Login Page – Customer can login with correct E-mail Id & password or can create a new account by clicking on register button



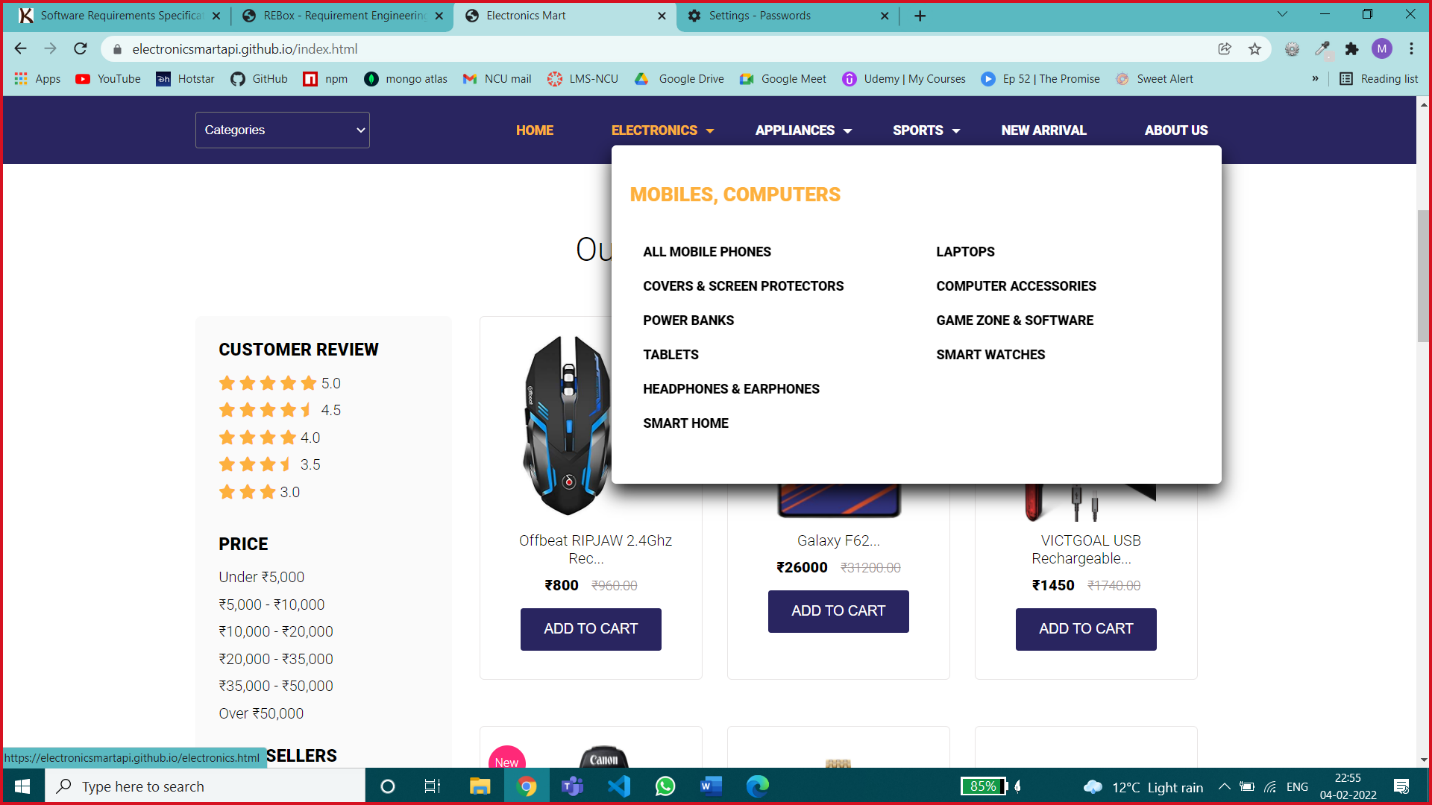
Registeration Page 1 – this page is accepting all primary & sensitive information like email id & password for creating new account, all fields are necessary.



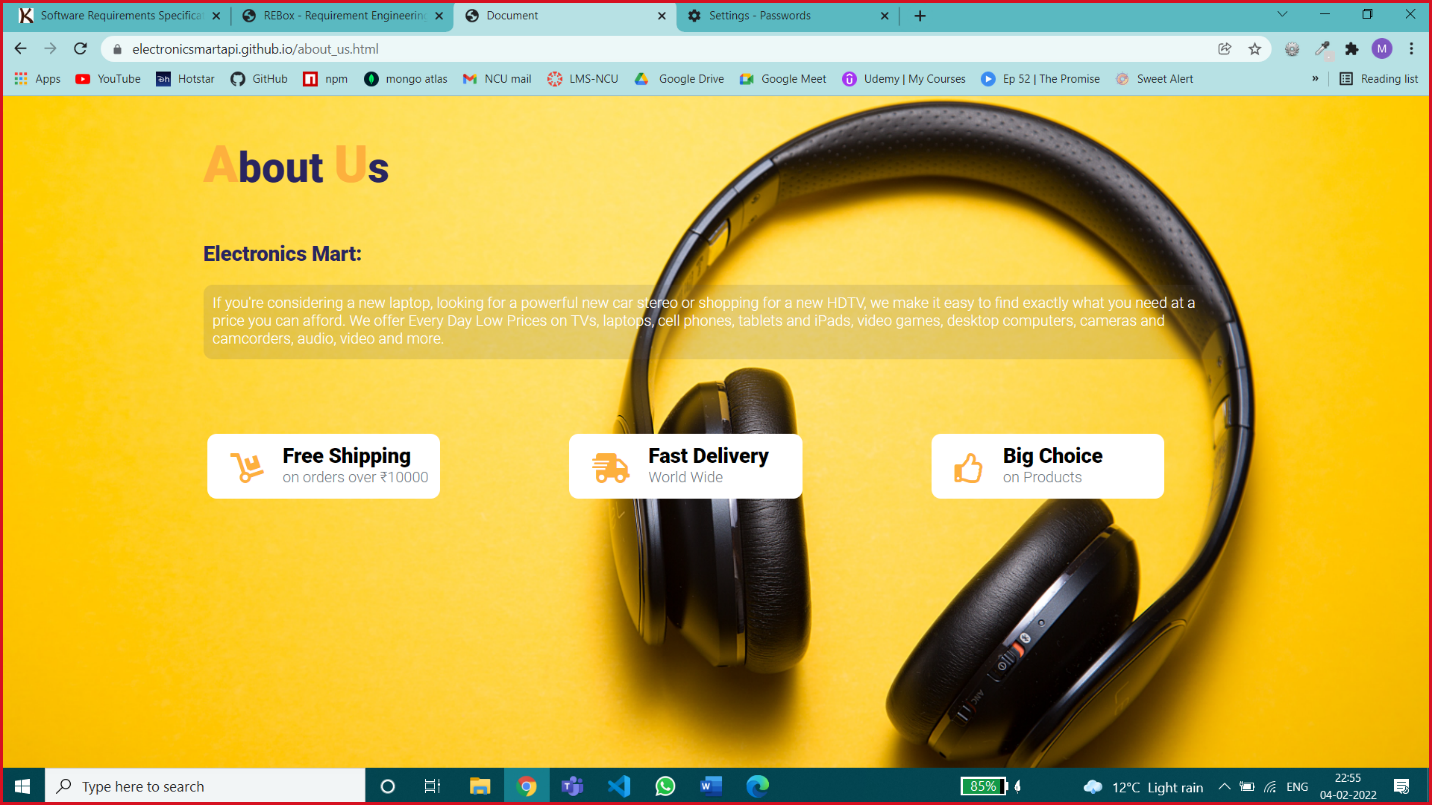
Registeration Page 2 – this page is accepting customer details like address & interested category of products by selecting popular categories & address is necessary, interest is an optional field.



After Successful Login User will be redirected to home page for browsing products, add products to cart, order products & will be provided my account, my orders, logout feature on the top left.



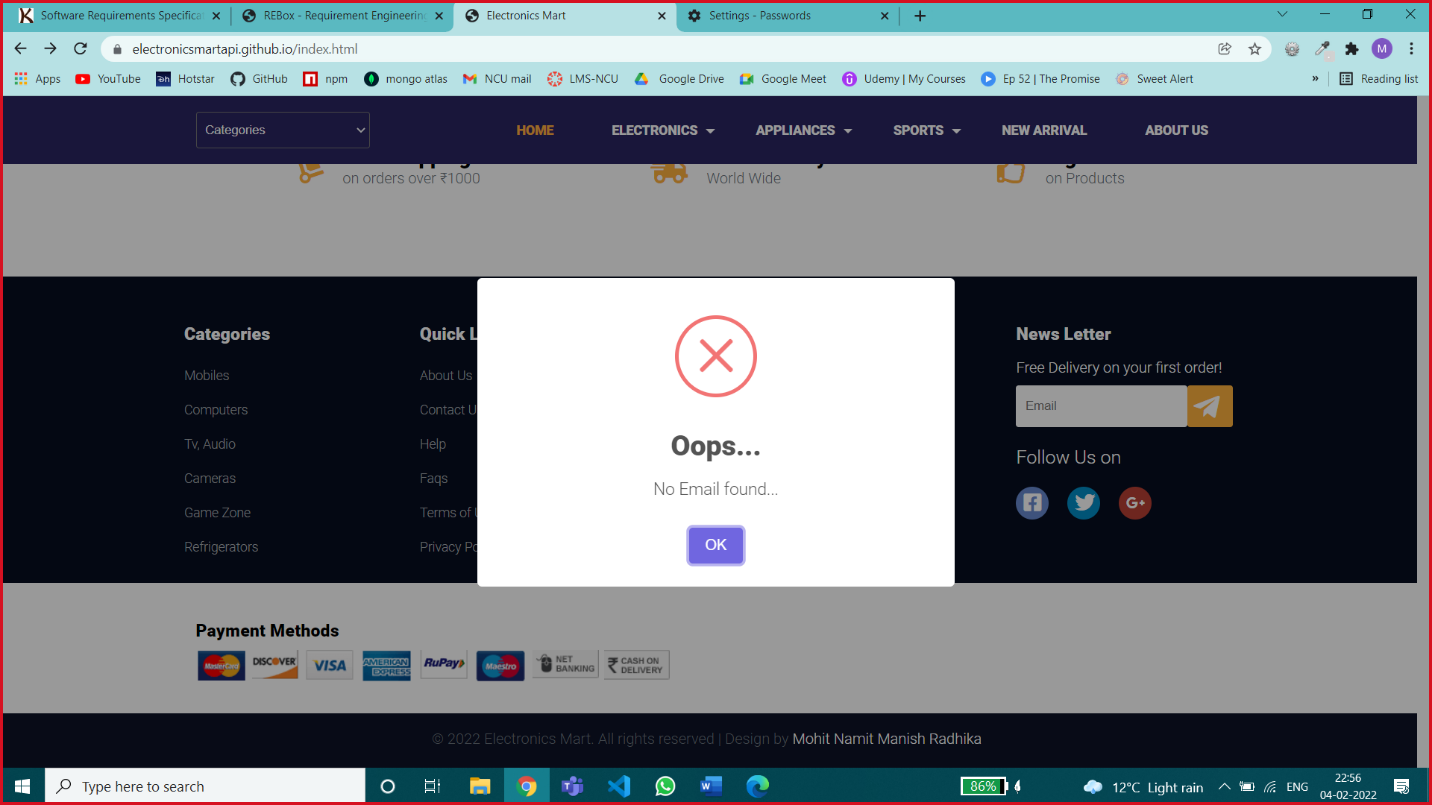
Sub-Category Dropdown Menu – In which major categories are divided into sub categories



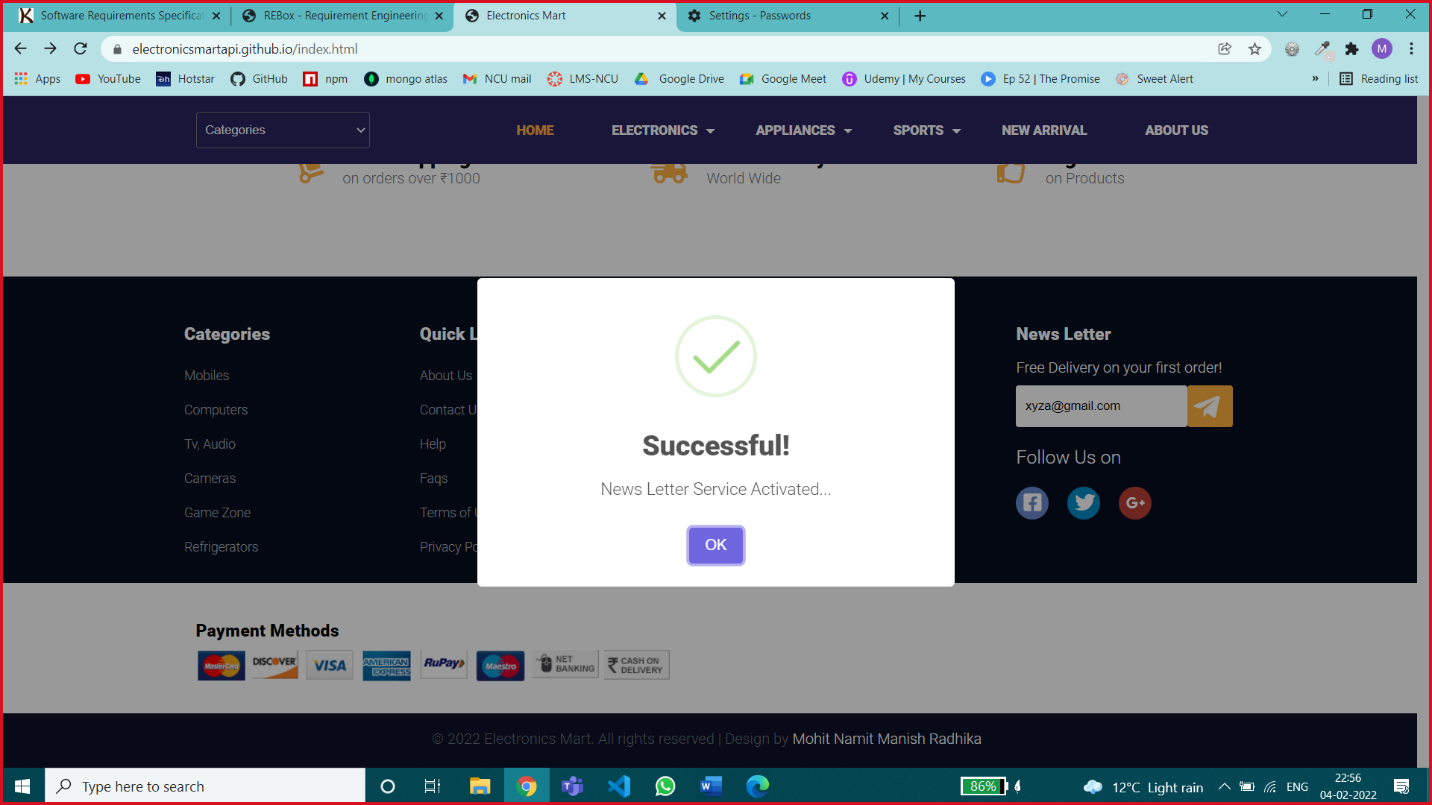
About Us Page – consist of basic information of website



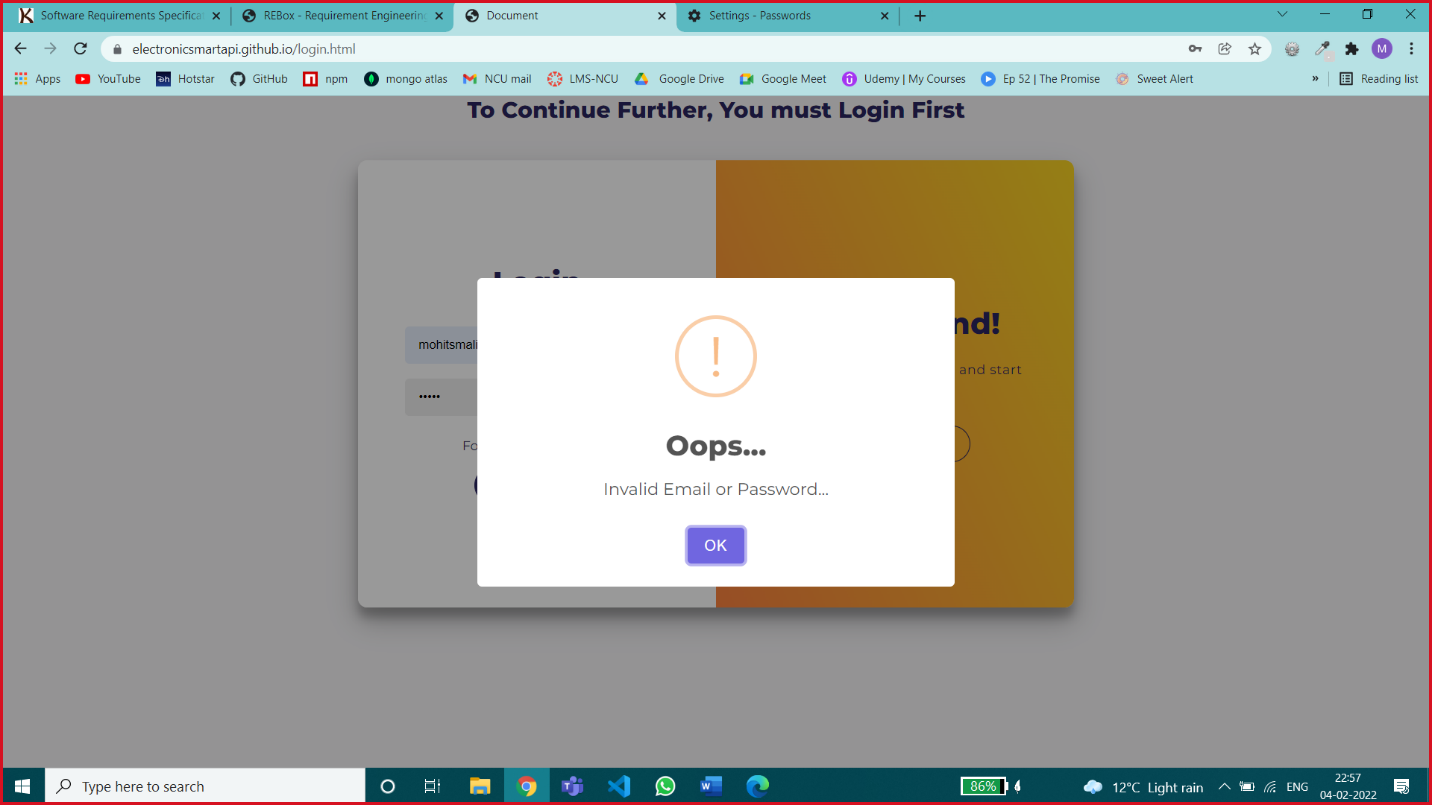
Footer of Website – News Letter service, Few other basic options.



Error Message for any Invalid Input or for any Blank Input where field input is necessary.



Success Message – for success Status



Error Message for Wrong Credentials

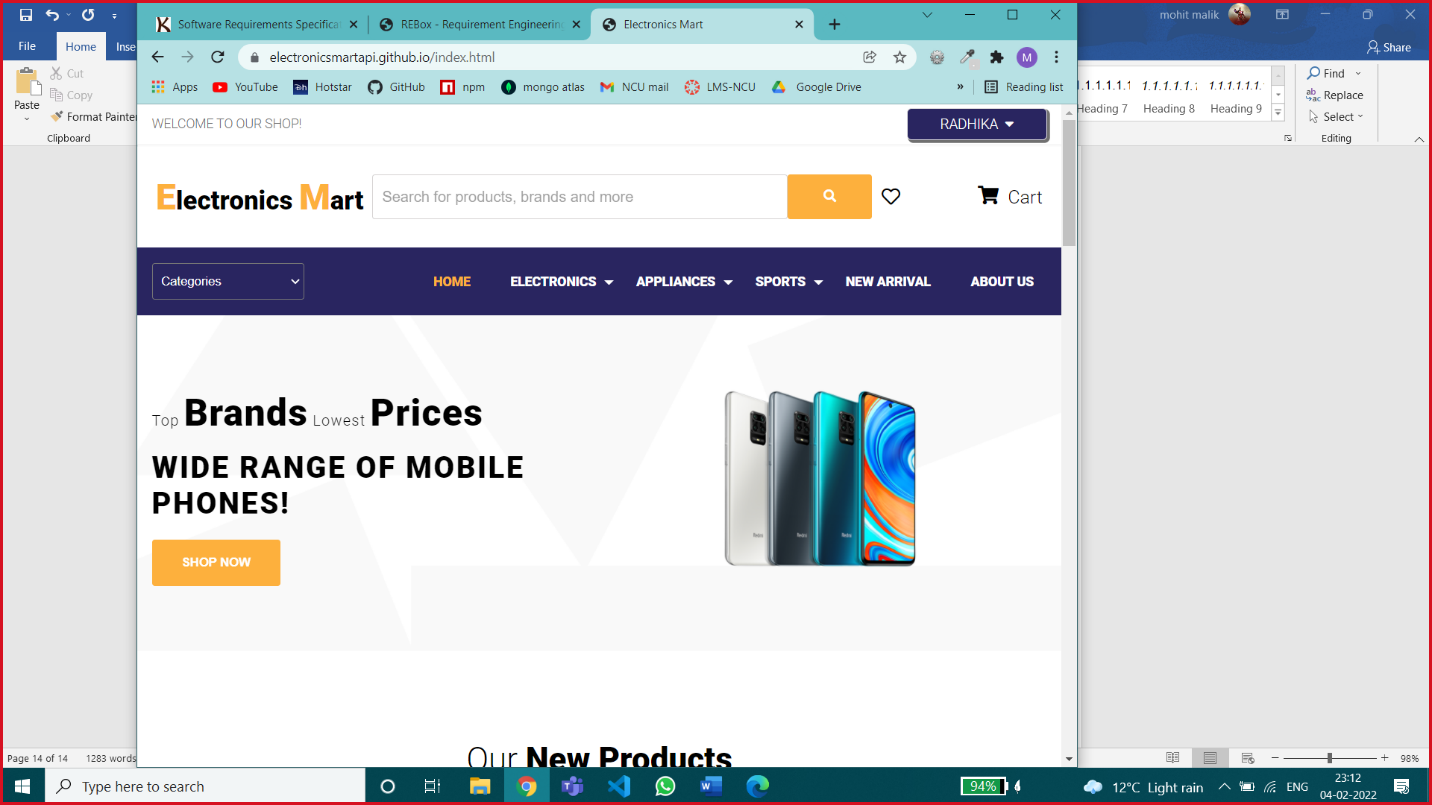


Loading Message

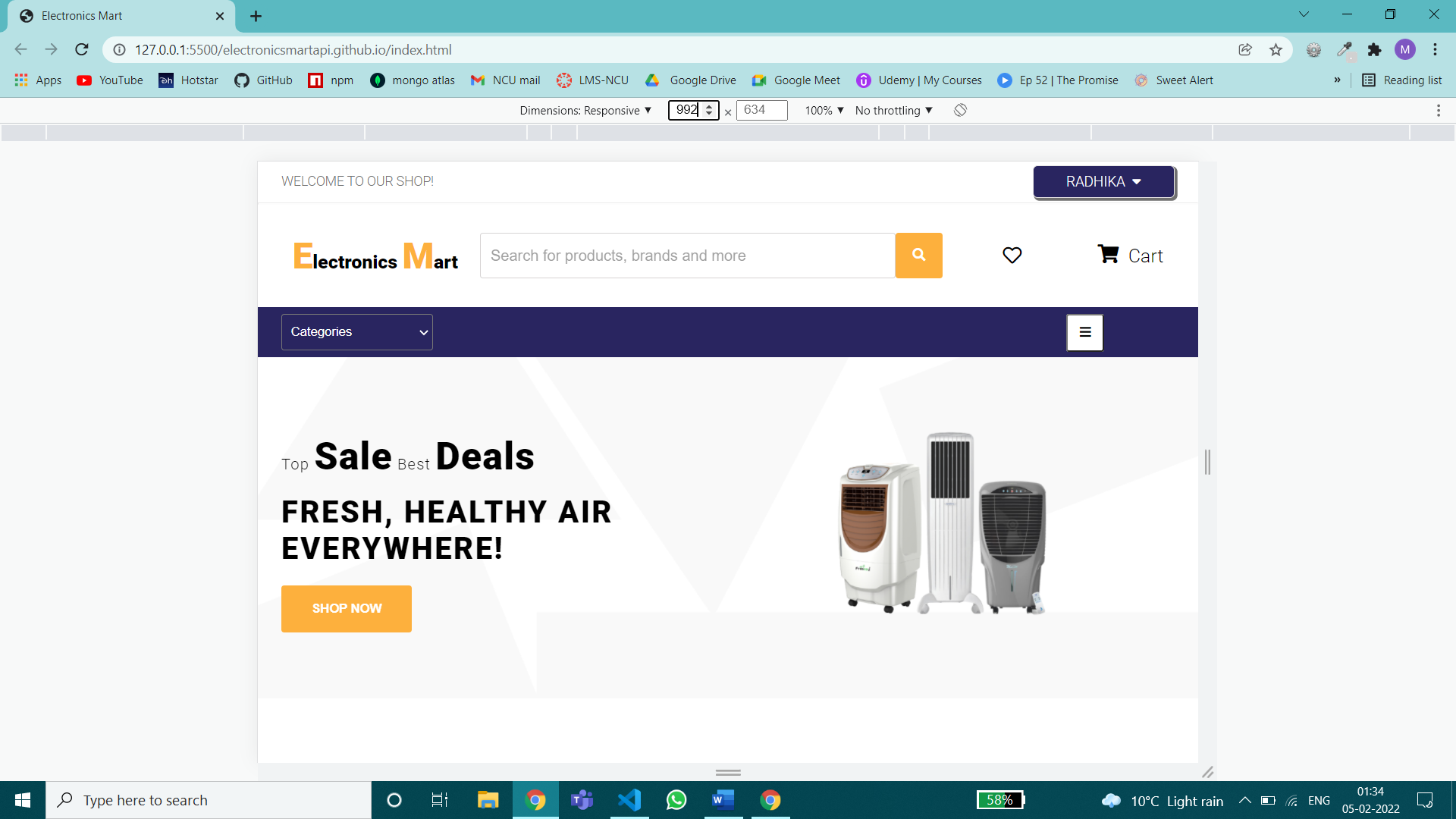
Screen Layouts Constraints –

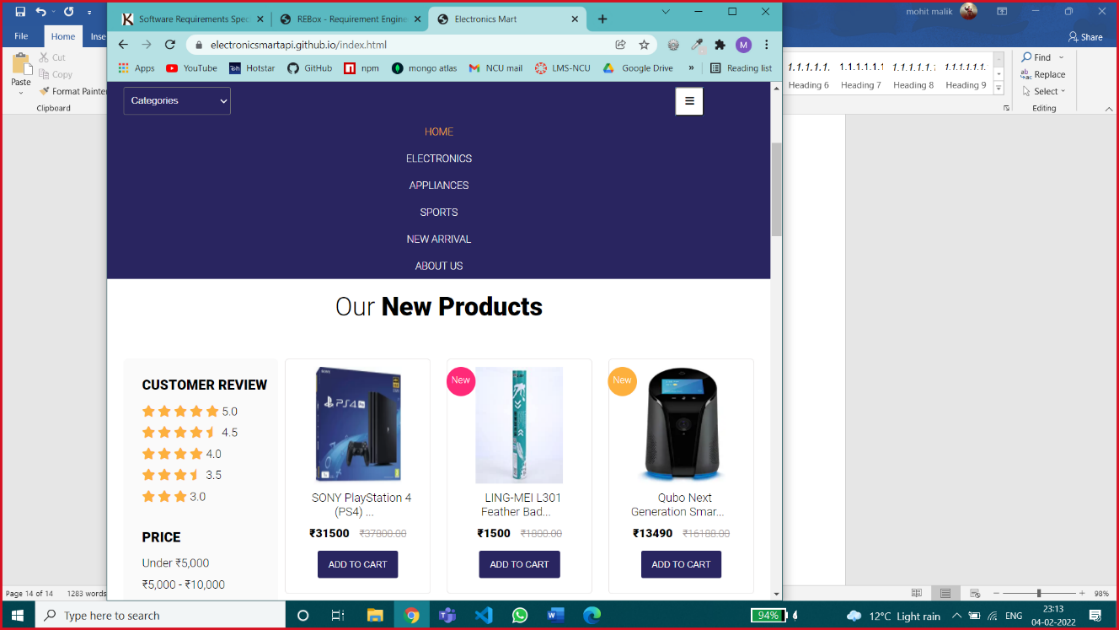
Device height is not a screen layout constraint, if the height varies the content will be vertically scrollable

For particular device width ranges different website screen layouts are shown below.

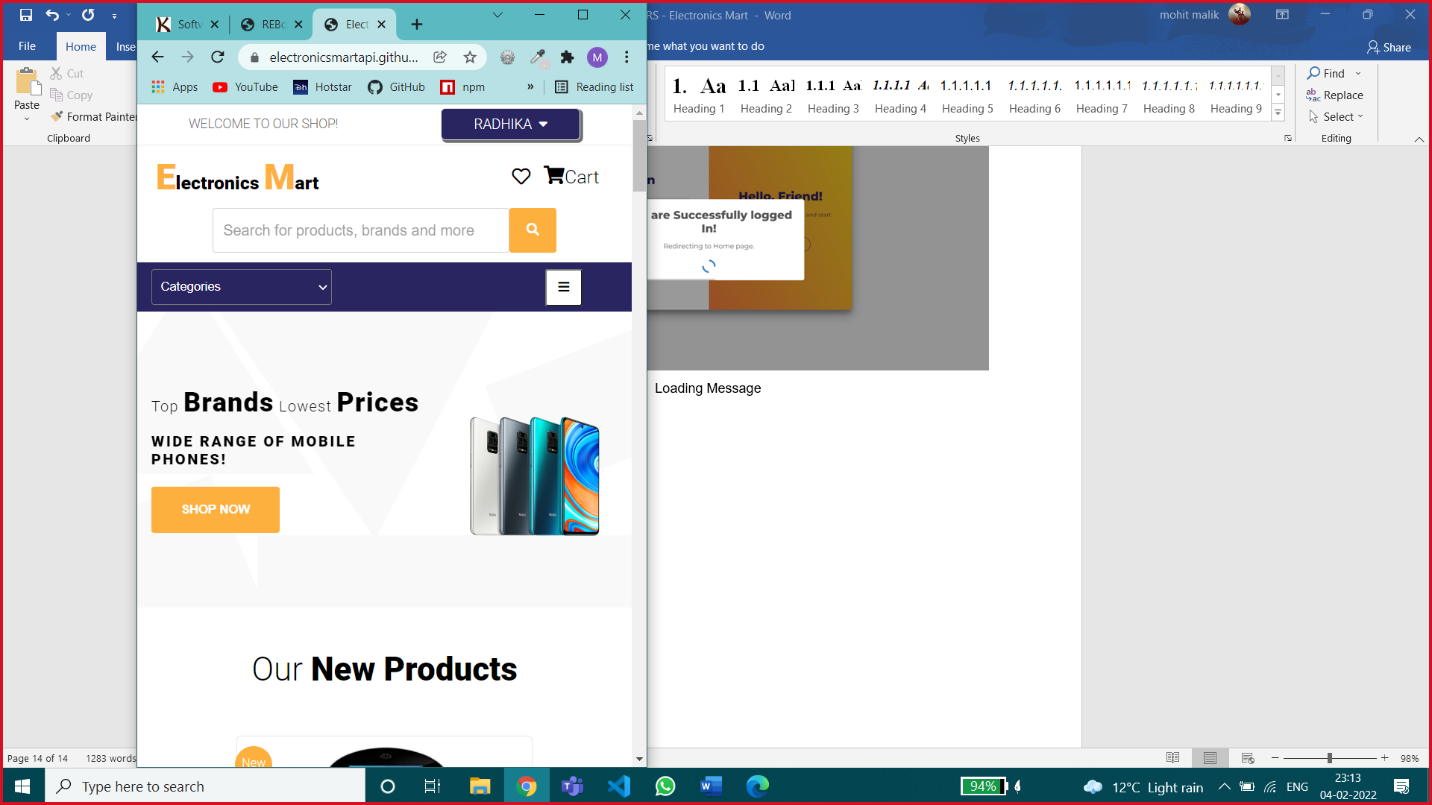
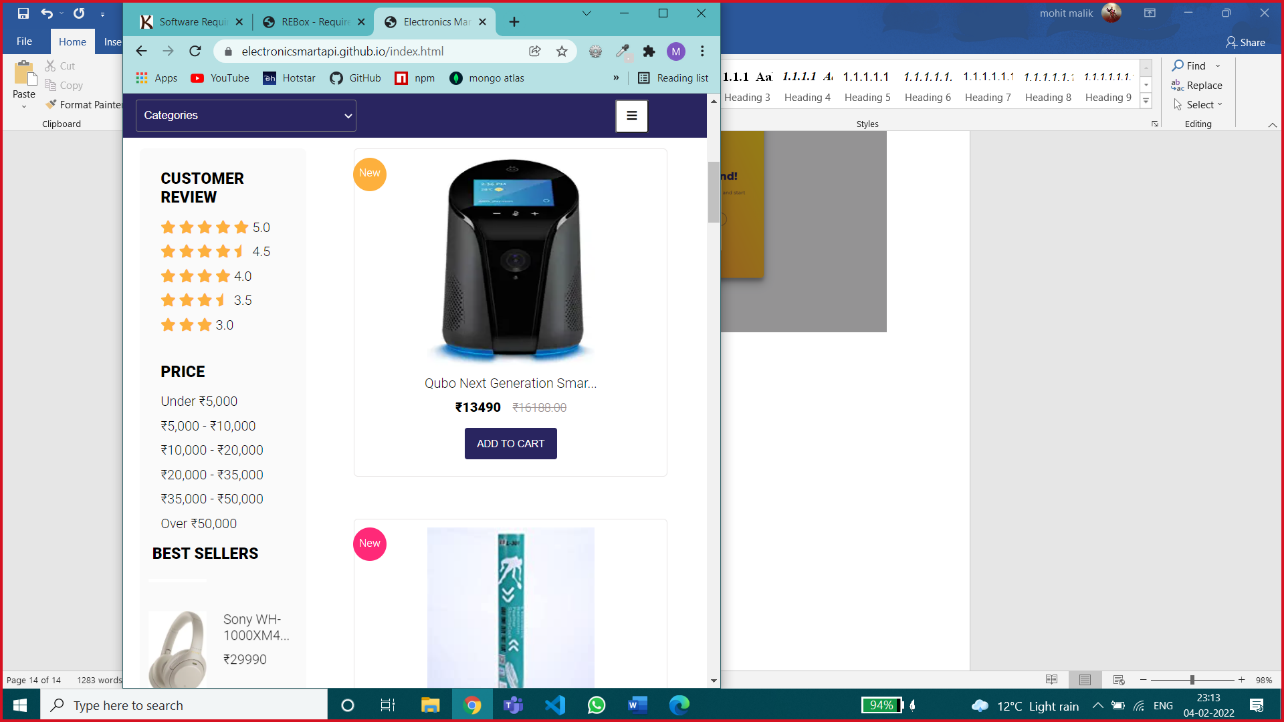


Screen width – greater than 992px

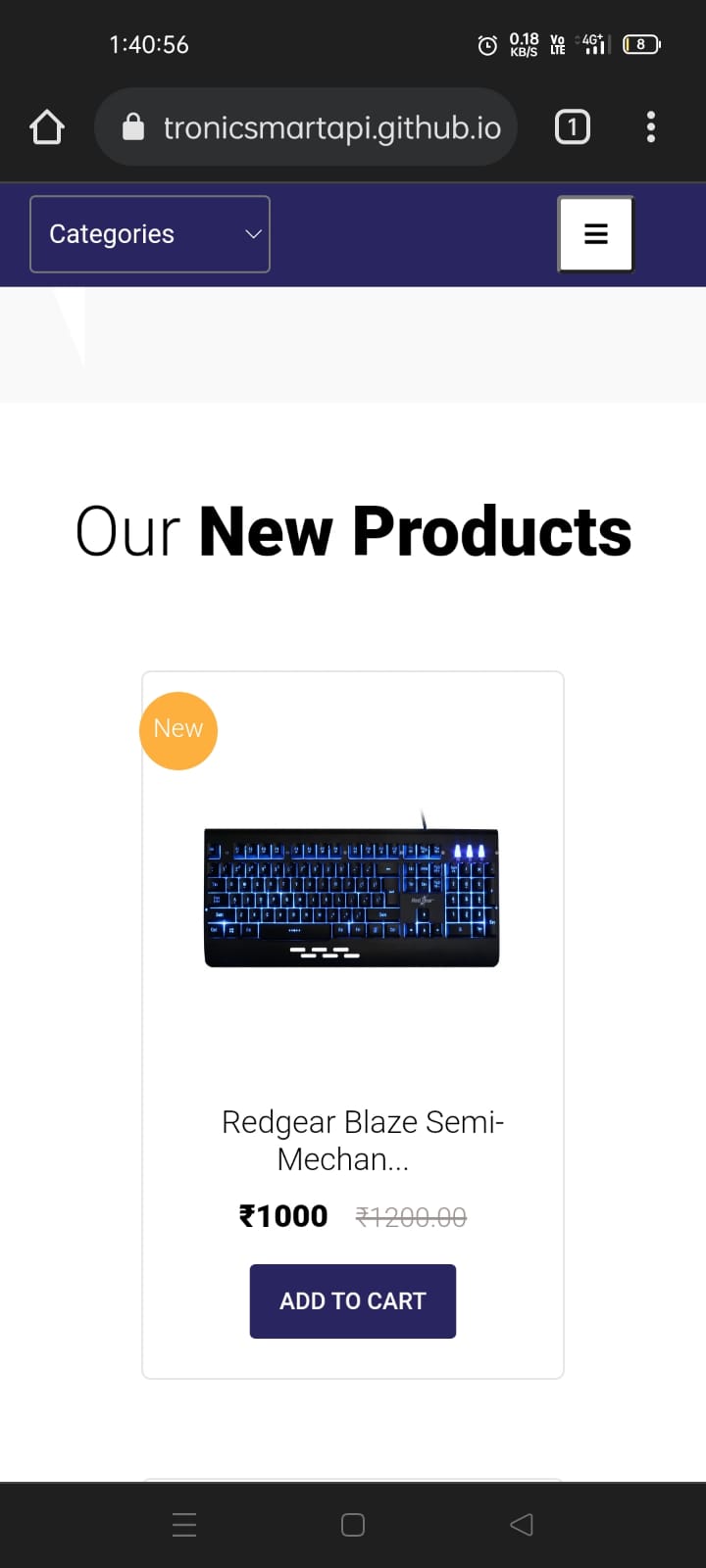
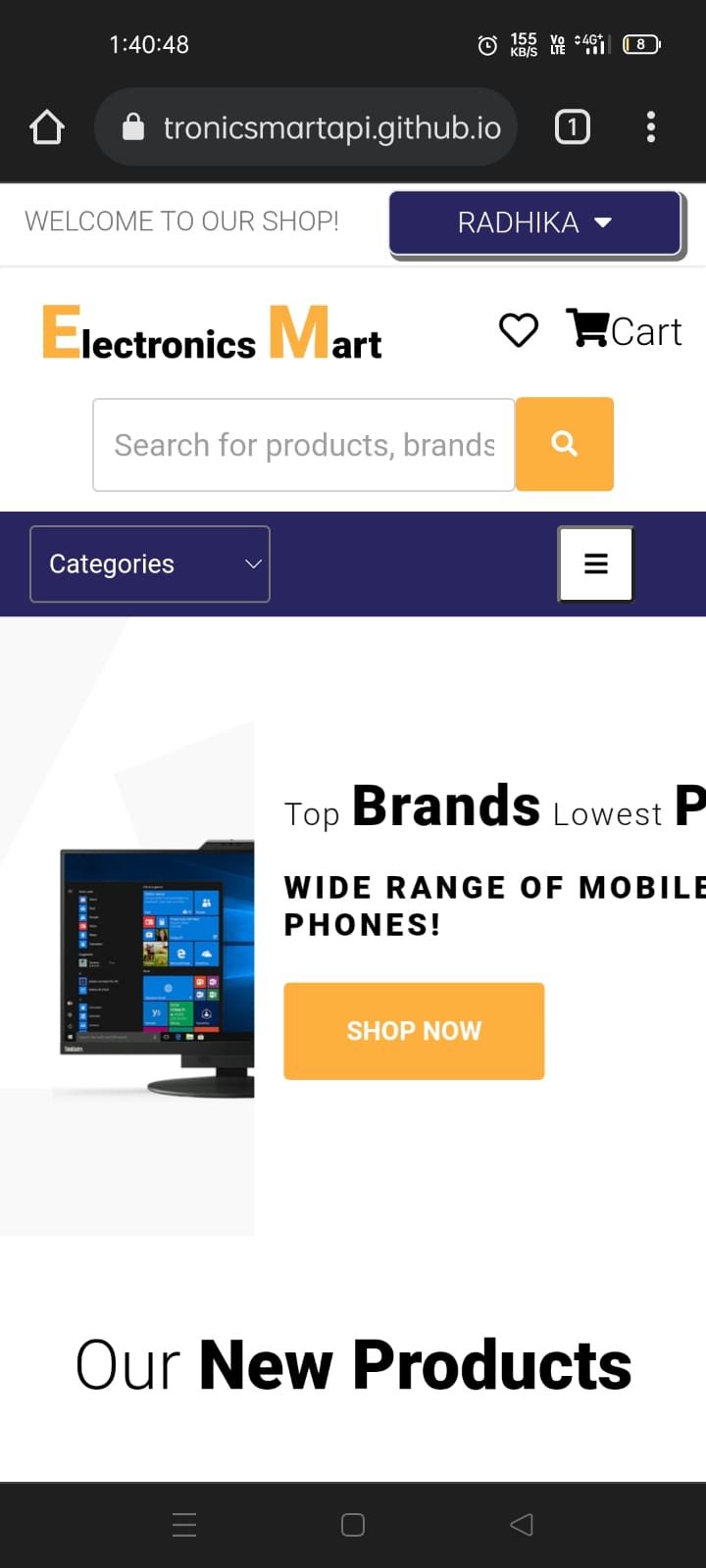




Screen Width between 992px & 768px

Screen width between 768px & 576px



Screen width below 576px

## Hardware Interfaces

Any Hardware Device that supports the following Browsers will be able to run Electronics Mart –

Good Data is tested against and supports the following desktop browsers:

Microsoft Windows 10,8,7

* Google Chrome (latest stable version)
* Firefox (latest stable version)
* Microsoft Edge (latest stable version; Chromium-based only)

MacOS 10.12 and newer

* Google Chrome (latest stable version)
* Safari (latest stable version)
* For best results, use the latest version of Google Chrome.

Mobile Web Browser

You can view dashboards through the following mobile browser:

Android 5 and newer

* Google Chrome (latest stable version)

iOS 10 and newer

* Google Chrome (latest stable version)
* Safari (latest stable version)

Browser Configuration

Any supported browser that you choose to use must meet the following requirements:

* JavaScript is enabled.
* Cookies are enabled.

## Software Interfaces

Databases – Mongo DB managed by Mongo Atlas on cloud.

Operating System – Windows /Mac or any other OS which are able to execute Node JS.

Tools – Postman for testing Backend Application, Github Desktop.

Libraries & Integrated Commercial Products – most of these are imported through NPM (Node Package Manager)

* Express JS – light-weight backend application framework.
* JWT – Json Web Token – for generating secure token for authentication.
* Uniquid – for generating unique Id.
* Bcrypt – For secure password encryption.
* Mongoose – for using Mongo DB queries in node js.
* Nodemailer – for sending mails to customers.
* Otp-generator – for generating otp for two factor authentication.

CORS –It is essential to enable CORS in the backend so that the frontend can request the API for sharing data. Express allows to configure and manage an HTTP server to access resources from this application.

## Communications Interfaces

Communication Standards &, network server communications protocols are HTTPS as a secure network.

Website E-mail – [electronicsmartapi@gmail.com](mailto:electronicsmartapi@gmail.com)

# System Features

## Authentication

4.1.1 Description and Priority

This feature will help company to identify user uniquely. So, this plays a crucial rule in system hence having high priority. This includes risk for a user in case of security breach like account hack or data leak, etc.

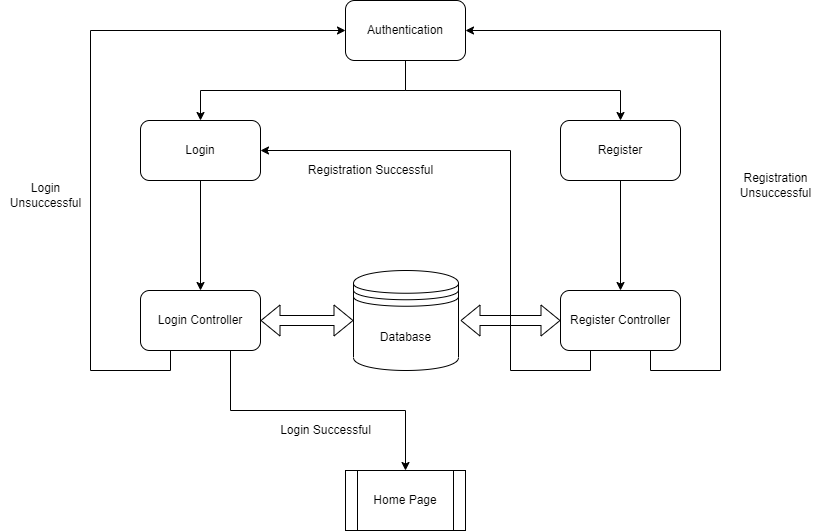
Actors for this feature –

* Users – They can use login/register
* Admins (unauthorized) – they can only use login feature, admins can’t register by themselves.
* Admins (Authorized) – they can login & after login they can register new admins.

4.1.2 Stimulus/Response Sequences

User Actions –

* Register
* Login



4.1.3 Functional Requirements

* Registeration: Any customer can register to Electronics Mart after filling necessary details in registration form, the email-id should be unique which means there can’t be any existing account with that email id (which is checked at the time of registration). After successful registration user should be redirected to login page, if registration is not successful user should be redirected to User authentication page having both options of login/register.

Input fields & their validation checks in registeration –

* Name
* Email-id – valid email address
* New password –
* atleast 8 letters or number
* atleast one lower case & one uppercase letter
* atleast one number & a special character
* Re-enter password – should be same as new password
* Address – House No., Landmark, City, State, Pin code (number only)
* Interested in categories – user can choose his/her interested categories of products from most popular categories, this is an optional input.

Register might fail in these cases – duplicate email-id, weak password (should pass password validation check), both passwords should match (new pass & re-enter new pass), if any necessary filed is empty last & least possible in case server is down then user will get an error message corresponding to particular error.

After successful registeration user will receive an email containing a unique account activation link, after successful activation of account user can login with his user id & password.

* Login: On login page any user has to enter correct email-id & password, then that email is checked in database if it’s existing then password that user entered is verified with the password in database. If its correct then login is successful & user is redirected to home page of web site where user has multiple functionalities as mentioned in next points.

Login might fail in these cases – If he has entered wrong email-id, wrong password it shows a message saying ‘Invalid Id or Password’ or if user leaves a field, a dialog box shows a warning that a field is empty.

On any new device login, user will get an email regarding security of his account.

If user has enabled Two factor Authentication, then after entering correct email-id & pass then user will be redirected to check OTP page, OTP main will be sent via email on user mail-id, after entering correct otp user can access his account.

User can use forgot password option in case he/she is not able to login due to password issues.

User will have 2 options for recover account –

Recover by email – user will get a recovery email & then user can enter new password with same password validation check as mentioned at the time of registeration.

Recover by old password – if user don’t know current password, then he/she can recover account by previous password & can set new password with same password validation check as mentioned at the time of registeration.

Email regarding successful account recover & new password will be sent to user’s email.

4.1.4 Modules used –

## Browse & Search Products

4.2.1 Description and Priority

This is one of the most important features of the website which enables the customer to browse products in an optimized way, hence having high priority. User can search products by applying many filters and combinations of different filters too.

Actors for this feature –

* All Users
* All Admins

4.2.2 Stimulus/Response Sequences

User Action –

* View all products
* View by major categories (like Electronics – phone, laptop, earphones, etc.)
* View by categories
* View by price range
* View by customer ratings
* View new arrivals
* Search products by name
* View by categories & customer ratings (combination of 2 filters)
* Search products by name in specific category (combination of 2 filters)
* View by rating & categories (combination of 2 filters)
* View by price range & categories (combination of 2 filters)
* View by interest (only for registered users if they have selected any interested category while registration process)
* View specific product with its details.

4.2.3 Functional Requirements

* View all products – users can see all products with mix categories & mix price range
* View by major categories – Products can be sorted via 3 major categories Electronics, Appliances and Sports.

These major categories include these products –

Electronics – Mobile phones, cover & screen protectors, power banks, tablets, headphones & earphones, smart home, laptops, computer accessories, game zone & software, smart watches

Appliances – TV, speakers, camera & DSLR, gaming consoles, AC, heaters, refrigerator, washing machine

Sports – cricket, badminton, cycling, football, gym equipment, sport shoes,

* View by categories – Users can select a specific category from the categories dropdown and view products.
* View by price range – Users can also browse products in between or less than specific price ranges.

Price ranges –

* Under ₹5,000
* ₹5,000 - ₹10,000
* ₹10,000 - ₹20,000
* ₹20,000 - ₹35,000
* ₹35,000 - ₹50,000
* Over ₹50,000
* View by customer ratings – Users can view product based on rating as well.

Ratings - 5, 4.5, 4, 3.5, 3 stars, on selecting any of these rating, the products having more than or equal to that rating will be displayed.

* View new arrivals – Products can also be browsed by new launched/arrived products, basically those products which were last stored in data base.
* Search products by name – A particular product can also be viewed by searching.

If that specific product is not present in the database, then the products related or having similar name will be displayed otherwise there will be no change in products.

* View by categories & customer rating – 2 filter of category & price range are applied together for better product searching.
* Search products by name in specific category – user can search products by name in a specific category.
* View by rating & categories – user can browse products by rating in a category or major category.
* View by price range & categories – user can browse products by price range in major categories (Electronics, appliances, sports).
* View by interest – users can see products by their interested category, this option is only available to registered users. Products will be show on the basis of the interested category they selected at the time of registeration, if they selected none then no change in products.
* View specific product with its details – users can view detailed view of a specific product like -
* Product description
* customer rating & reviews
* can search specific keywords in reviews
* If product’s quantity is less than 10 in stock then it’s also displayed or if it’s out of stock

\*Only 9 products will be shown on one screen. for view more user have to go to next page.

4.2.4 Modules used -

## Order Products

4.3.1 Description and Priority

This feature will help logged in users to order any product in stock. This feature is using

an external plugin dependency for payment. After successful payment the product is

ordered. This is one of the key features of shopping website & involves payments due to

which its having high priority & risk.

Actors for this feature –

* Registered User
* Unauthorized Admins
* Authorized Admins

4.3.2 Stimulus/Response Sequences

User Action – Order products, cancel or return product & view previous orders.

Unauthorized Admins – can view all orders

Authorized Admins – can view all orders & update order status

4.3.3 Functional Requirements

* Order Products – User can order any product in stock after successful payment, payment

Implemented by an external plugin.

User will get an email after successful order, which includes order Id.

* Cancel/return Order – user can cancel any order or return it after mentioning a valid.

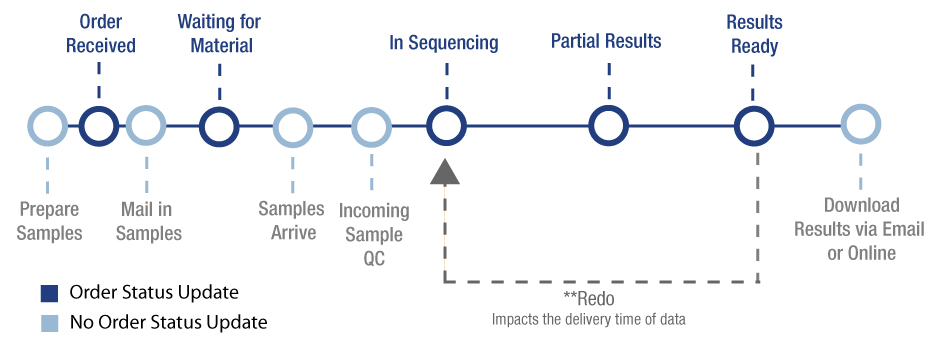
Reason, in case he has returned the quantity of the product will be restored & updated in

Database. In case of return it should be sent to different department.

* View my orders – a user can also see his previous orders.
* View all orders – all admins can view all orders in optimized way.
* View orders of a single users – enter user id
* View orders by their status – choose an order status
* View order of a particular product – enter product id

\*Only 9 orders will be visible on screen at once, for view more user have to go to next page.

* Update Order Status – authorized admins can update order status on completion of different stages of Order.



Pending — Customer started the checkout process but did not complete it. Incomplete orders are assigned a "Pending" status and can be found under the More tab in the View Orders screen.

Awaiting Payment — Customer has completed the checkout process, but payment has yet to be confirmed. Authorize only transactions that are not yet captured have this status.

Awaiting Fulfillment — Customer has completed the checkout process and payment has been confirmed.

Awaiting Shipment — Order has been pulled and packaged and is awaiting collection from a shipping provider.

Awaiting Pickup — Order has been packaged and is awaiting customer pickup from a seller-specified location.

Partially Shipped — Only some items in the order have been shipped.

Completed — Order has been shipped/picked up, and receipt is confirmed; client has paid for their digital product, and their file(s) are available for download.

Shipped — Order has been shipped, but receipt has not been confirmed; seller has used the Ship Items action. A listing of all orders with a "Shipped" status can be found under the More tab of the View Orders screen.

Cancelled — User can cancel the order until it’s out for delivery or Seller can cancel an order, due to a stock inconsistency or other reasons. Stock levels will automatically update depending on your Inventory Settings. Cancelling an order will not refund the order. This status is triggered automatically when an order using an authorize-only payment gateway is voided in the control panel before capturing payment.

Returned — Seller has used the Refund action to refund the whole order. A listing of all orders with a "Refunded" status can be found under the More tab of the View Orders screen.

4.3.4 Modules used –

## Manage My Account

4.4.1 Description and Priority

This feature will allow logged in user to view, update, delete account. This includes user data so this is a medium priority feature. Risk regarding data leak or hacking of account should be handled properly for security.

Actors for this feature –

* Registered users
* All admins

4.4.2 Stimulus/Response Sequences

* View my account
* Update account details like –
* Name
* Email id
* Password
* address
* Enable/disable two factor authentication
* Delete account

4.4.3 Functional Requirements

* View my account – user can view his/her account details i.e., name, email-id, address, two factor authentication enabled or disabled.
* Update account details –
* Name – user can change his/her name
* Email id – user can change his/her email id, the new email id should not be registered to any other Electronics Mart account & should be a valid email id.

Email will be sent to the new email id for successful update.

* Password – user has to enter the old password of account in order to change password, the new password validation is same as password validation at the time of registeration.

Email will be sent to the user’s email id about change of password, for security in case the real owner hasn’t updated the password then he/she should recover account as soon as possible.

* Address – user can update address, user can change whole address or any single attribute of address on his/her choice.
* Enable/disable two factor authentication - user can enable of disable this extra account security feature, in this if user logins then a 6-digit OTP will be sent to user’s email id and he/she has to enter that OTP at the time of login.

Email will be sent to user’s email id about enabling or disabling of Two factor authentication.

* Delete account – user can delete his/her account any time they want, at the time of deletion they have to enter correct password of their account for verification.

Email will be sent to the user’s email id about deletion of account.

The user data will be stored in another collection for one year of time span.

4.4.4 Modules used –

## Manage My Cart

4.5.1 Description and Priority

This feature will allow logged in users to add, delete items in cart. Users can directly place order

From my cart if that product is in stock. This is an additional feature so its having low

Priority.

4.5.2 Stimulus/Response Sequences

* Add items to cart
* View my cart
* Update items in cart
* Delete items in cart
* Place order

4.5.3 Functional Requirements

* Add items to cart – user can add any product to cart, initially quantity of product in cart is one.
* View my cart- user can view items added into cart, only 9 items will be shown on a page, for more user has to go to next page.
* Update items – user can increase or decrease quantity of items in cart.
* Delete items in cart –
* Delete one item – user can delete any single item in cart.
* Delete all items – user can delete all items in cart, basically user cart will be emptied.
* Place order –
* Order one item – user can order any single item in cart if that item is in stock.
* Order all items – user can also order all items in cart, but if any item is out of stock, then user has to order other items one by one.

4.5.4 Modules used –

## Manage Products

4.6.1 Description and Priority

This feature will allow logged in admins can add, update, delete product after successful login.

This will provide an Interface to manage products so this is a low priority feature.

Actors for this feature – Authorized admins

4.6.2 Stimulus/Response Sequences

* Add product
* Update product
* Delete product

4.6.3 Functional Requirements

* Add Product – Admins can add a single product at a time.

Input fields & their validation checks –

* Product name
* Product description
* Product Image link – should be a valid image link
* Product price – should be greater than or equal to 0
* Product Quantity – should be greater than or equal to 0
* Product Category

If any of the field is left empty product can’t be added.

* Update Product – Admins can update different attributes of product
* Update product name, price, description, image, category with same validation checks which were used at the time of adding product
* Update product quantity – admins may update quantity more often than product other attributes so update quantity is separated from update of other attributes of products, Admins have to enter the change in product quantity and total quantity of product should be greater than or equal to zero otherwise update will be unsuccessful.
* Delete product – admins can delete any product from database if required.

4.6.4 Modules used –

## Manage Admins

4.7.1 Description and Priority

This feature will allow logged in authorized admins to view admins, authorize/unauthorize admins / delete admins. This feature is having high priority because there's risk of administration power.

Actor for this feature - Authorized admins

4.7.2 Stimulus/Response Sequences

* View admins
* Authorize/Unauthorize admins
* Delete admins

4.7.3 Functional Requirements

* view admins - admin can view all admins, for optimized search admin can search admins by -
* Search by admin id
* Search by email
* Search by name
* View a specific admin - complete details of the admin will be shown

\*Only 9 orders will be visible on screen at once, for view more user have to go to next page.

* authorize/unauthorize admins - admins can Authorize or unauthorize other admin as per requirements, option for authorizing or unauthorizing other admins will be provided in detailed view of that admin

Email will be sent to that admin's email id who is authorized or unauthorized.

* delete admin - admin can delete any admin as per requirements, this feature is also provided in detailed view of that admin.

Email will be sent to that admin's email id that his account has been deleted if it looks wrong, he/she can contact the administration.

4.7.4 Modules used –

## Manage Users

4.8.1 Description and Priority

This feature will allow logged in admins to view & search users in an optimized way & delete any user if required. This feature is helping admins to manage users so it a medium priority Feature.

Actors for this feature

* Unauthorized admins - can view & search any user.
* Authorized admins - can view & search any user and delete.

4.8.2 Stimulus/Response Sequences

* + View users
  + Delete users

4.8.3 Functional Requirements

* View users - admins can view or search users in an optimized way
* View all users
* View by name
* View by email id
* View by user id
* View by city
* View by state
* View by pin code

\*Only 9 users will be visible on screen at once, for view more user have to go to next page.

* Delete users – only authorized admins can perform this action, if any user is violating any T&C, then his/her account can be deleted.

Email will be sent to user’s email id that his account has been deleted if it looks wrong, he/she can contact the administration.

4.8.4 Modules used -

## Rate & Review products

4.9.1 Description and Priority

This feature will allow logged user to rate & review product so that they can share their experiences of using products with other buyers. This feature helps other users to select better product so, this is feature is having low priority. Any user can only rate or review product when he/she has ordered that product & order status is delivered. Otherwise, the use user can’t rate or review it.

Actor for this feature – Registered user

4.9.2 Stimulus/Response Sequences

* Rate product
* Edit rate
* Review product
* Edit review

4.9.3 Functional Requirements

* Rate Product – User can rate the product once the product is delivered to him/her, rate the product from 1.0 to 5.0 where higher rating means good product.
* Edit rating – User can edit rating of the product he/she has given earlier.
* Review product – user can review the product once the product is delivered to him/her, in review user can share his/her experience in a paragraph with a word limit of 200 words.
* Edit review - User can edit review of the product he/she has given earlier.

4.9.4 Modules used -

## News Letter service

4.10.1 Description and Priority

This feature will allow users to receive news letter emails from website & logged in admins to send newsletter email & view previous newsletter emails. This is an additional feature of website so it’s having low priority.

Actors of this features –

* All users
* All admins

4.10.2 Stimulus/Response Sequences

* Activate newsletter service
* Send newsletter
* View previous newsletters

4.10.3 Functional Requirements

* Activate newsletter Service – any user can activate newsletter service by entering their correct email address in newsletter section of the website, already registered user need not to activate this service because they will receive all newsletter regarding exiting deals & sales.
* Send newsletter – any admin can send newsletter which require a subject & body.
* View newsletter – any admin can view previous newsletter, the subject, body, reciver’s list of the newsletter.

4.10.4 Modules used -

# Other Nonfunctional Requirements

## Performance Requirements

* Use best servers & techniques for data transfer to achieve better performance.
* Use light-weight frontend so that load on end user device is reduced which will give better performance.
* Minimize redirects on your site by frontend AJAX call through which user will get required data without redirecting.
* Use of projection will remove unnecessary data like createdAt, updatedAt, etc. while sending data from backend to frontend which will reduce size of data to be transferred & therefore less loading time hence achieving better performance.
* Use separate database to store images & save the links of images in there corresponding objects which will enhance performance because the images will only load when end user visits the website & will not travel from database to controller & then controller to frontend.
* Use Indexing on product search as it will gives result faster than normal searching.
* Referral traffic knowledge who sends you the most traffic whether its google or your advertisement – can be very valuable in deciding how to best target the majority of your visitors.

## Safety Requirements

* Maintain a backup of data because in case database server gets crashed or something else happen data could be recovered easily.
* Continues Monitoring is required for proper maintenance of website, in case load on website is at its peak the servers should be upgraded by vertical or horizontal scalability as per requirement & we are using a distributed system so it can be achieved easily.

## Security Requirements

* The password must be encrypted while saving in data base.
* The token-id should expire after 2 Hours.
* A middleware should be plugged in before other routes which requires user’s details for authentication which will authenticate users because we hare using HTTPS protocol which is connectionless.
* Another middleware for admins before admins routes (can only view users, orders, products) and another middleware for authorized admins before authorized admin routes (can manage products, users, orders).
* Another middleware if user has activated Two factor authentication which should check whether user has confirmed otp before entering routes which requires user details.

## Software Quality Attributes

* Testing: The software should be tested for both backend, frontend & after integrations another integration testing should be performed.
* Updates: Updates can be easily managed and joined with distributed node JS system.
* Portability: The software should be portable from one device to another i.e., laptop, mobile, etc.
* Availability: The software is 24X7 available except backend servers may get crashed for that we may deploy the website’s backend on different servers after dividing into different nodes so that whole application doesn’t go down only one or two functions may not work in case server gets crash.

## Business Rules

* User will get mailed on every login so if it’s not the user then the user must change the account password for security as soon as possible.
* The admin has rights to delete any users account if the user is violating any rules.
* The admins can update order status and can cancel the order in case the product is unavailable due to some reason.
* If a user cancels any order, then the product quantity, he cancelled should get added to orignal product quantity.

# Modules

User/admin Modules

Check email

Input – email

Description – This module will check whether the user/admin entering an email for registration is already present in the database or not.

Output – If email is already present, the user will get an email already used message. Else user can complete further registration process.

Register

Input – name, email id, password, address: {house no, city, state, pin code, landmark}, interested in categories

Description – A new user or an authorized admin for registration of a new admin will be using this module to create an account. User/admin needs to provide all the input fields and may also select favorite categories in case of user. a unique id will be generated for each user/admin by using his/her name and this unique id and all the information about the user/admin will be stored in the database collection.

Before saving a password in the database, it should be encrypted for security.

A unique key should be generated & stored in the database, also mapped with the account activation link for generating a unique account activation link for each user/admin & send to them after successful registration on their email id.

Few keys of a new user/admin object should have some by default value –

* account\_activated = 0
* two\_factor\_auth = 0
* otp\_checked = 0
* old\_pass = password

Output – If all the validations match and data is successfully stored in the database, the user will get a successful prompt and get a successful mail on the entered mail id. Else will get a failure message.

Activate account

Input – key (from the query)

Description – After registering user will get a unique account activation link, once the user clicked on the link the key in the link will be verified from the user object in the database and the account\_activated key will be equal to 1.

Output – if an account is successfully created, a confirmation mail be sent & a successful message will be displayed on the screen. In case of any errors, the user will get an unsuccessful prompt.

Login

Input – email id, password

Description – this module will verify whether the email id entered is present in the database or not, if its present then it will compare the password with database encrypted password with a 3rd party dependency, if it’s matched it will return the document, same will be checked whether the password is matching to old\_pass, if matched it will return the document. If not matched, then the user/admin has entered the wrong password or email.

After getting a document, this module will check whether the account is activated or not.

If activated It will check whether the user/admin has activated two-factor authentication, if activated a 6-digit OTP will be generated, saved in the user/admin document & send to the user/admin email for verification, and redirected to verify OTP page.

If two-factor authentication is not activated a token will be generated with the help of a generate token module and user/admin name and token will be sent.

If the old password has matched, then the name will be sent.

Output – if no document is found with a password or old password then the user/admin will get an invalid email or password prompt message. If user/admin has entered an old password, user/admin will get an old password entered prompt message. If user/admin has entered correct email id, password & activated two-factor auth but due to some error unable to send OTP user/admin will get error prompt message with a token, user/admin name, if user/admin has entered correct email id, password & activated two-factor auth, OTP is sent to registered email then user/admin will get successful login prompt message & email, token, user/admin name. if user/admin entered correct email id, password & two-factor auth is not activated then user/admin will get successful login prompt message & email, token, user/admin name

Check OTP

Input – OTP, token

Description – If user/admin has enabled two-factor authentication, this module receives OTP from user/admin for verification, the OTP will be verified from the database which was sent to user/admin at the time of login to his/her email id. If it is correct, then key “otp\_verified” will be updated to 1.

Output – If the OTP is correct, the user will get an OTP verified message and will be able to use website functionalities. Else, the user will get an instant security message on the mail id and the wrong OTP popup on the screen.

Forgot pass

Input – email id

Description – If the user/admin has forgotten their password then this module will help them to recover their account. They need to enter their registered email id, then the application will verify whether this email is in our database or not. If it's present in the database, then the user/admin will get two methods to recover his/her account, by recovery email or recovering account by the previous password. If the user opted for a recovery email, then a unique key will be generated & stored in the database and mapped with account recovery link & sent to the user/admin through email or else if opted for recovery by old password the user/admin will be redirected to account recovery through old password page.

Output – If the email is in the database & user/admin opted for account recovery through email & email is sent then user/admin will get an email successfully sent a prompt else if the email isn’t sent then a failure prompt, if user/admin opted for account recovery though old pass then they will be redirected to another page, if the email is not found in database, then email not found prompt message will be displayed.

Account recovery by email

Input – key, new password

Description – this module will help the user/admin to recover his/her account through the unique account activation link sent to his registered mail. The application will get the key from the link & the new password entered by the user/admin and will verify the key if the key is correct then the current password will become the old password and the new password will be saved as the current password after encryption in the database. Else if the key is not correct then the account can’t be recovered by this method.

If successful, then the user/admin will also get an email for successful account recovery.

After successful account recovery, the user/admin can log in through this new password.

Output – If the key is not correct then the user will get an account recovery unsuccessful prompt message or else if the key is correct then the user will get the account successfully recovered, password updated prompt & an email will be sent.

Account recovery by old password

Input – email, old password, new password

Description – this module will help user/admin to recover his/her account with an old password if he/she remembers, the user/admin has to enter email id, old password & new password so, if the old password matches the encrypted old password saved in that user/admin document in database after decryption, then his old password will be updated by current password & new password will become current password after encryption & saved in the database and an email will be sent about successful account recovery. Else If the old password doesn’t match then the account can’t be recovered by this method.

Output – If the old password is correct & the password has been updated successfully then the user will get a successfully account recovered prompt message & a mail else if the old password doesn’t match user will get an account recovery unsuccessful prompt message.

View my account

Input – token

Description - User/admin can view the account information as well. This module will receive a token to get the user/admin id and will search that id in the database for complete information about the user/admin to be displayed.

Output – User/admin will be displayed the complete account information if successfully found in the database. In case not found then user/admin will get data not found prompt.

Activate two-factor auth

Input – token (for user-id)

Description – User can enable two-factor authentication by this module, user/admin id will be retrieved from token, then “two\_factor\_auth” key will be equal to 1 in database & after successfully activating the two-factor authentication a mail will be sent to user/admin email id.

Output – If two-factor authentication is enabled, the user will get a confirmation mail and a successful prompt & new token will be set & saved. In case of an error, the user will get an unsuccessful message.

deactivate two-factor auth

Input – token (for user-id)

Description – User can disable two-factor authentication by this module, user/admin id will be retrieved from token, then “two\_factor\_auth” key will be equal to 0 in database & after successfully deactivating the two-factor authentication a mail will be sent to user/admin email id.

Output – If two-factor authentication is disabled, the user will get a confirmation mail and a successful prompt & new token will be sent & saved. In case of an error, the user will get an unsuccessful message.

Update name

Input – token (for user-id), name

Description – User/admin can update his/her name. By this module name of that user/admin will be updated to the new name in the database.

Output – If the update is successful, the user will get a successful prompt and a new token will be sent & saved. Else, the user will get an unsuccessful prompt message.

Update email

Input – token (for user-id), email

Description – User/admin can update his/her registered email. By this module email of that user/admin will be updated to the new email in the database. A confirmation mail will be sent to the new email of the user/admin. User/admin will be only able to log in through this email after a successful update.

Output – If the update is successful, the user will get a successful prompt and a mail on the mail id new token will be sent & saved. Else, the user will get an unsuccessful message.

Update password

Input – token (for user-id), new password, old password

Description – this module will help to update the password of an account. User/admin must enter old password & new password, email id of user/admin will be retrieved from token & the old password will be matched to the encrypted password in database after decryption, if the password is correct then the old password will be updated to current password & current password will be updated to a new password, a confirmation mail will be sent to user/admin email id about password update. User/admin will be only able to log in through this new password after a successful update.

Output – If the old password is correct & the update is successful, the user/admin will get a successful prompt and a mail on the mail id new token will be sent & saved. Else, the user/admin will get an unsuccessful message accordingly.

Update address

Input – token (for user-id), house no, city, pin code, landmark

Description – In this module user/admin can update his/her address details. The address will be updated to the new address in the database.

Output – If the update is successful, the user/admin will get a successful prompt. Else, the user/admin will get an unsuccessful message.

Delete User

Input – token (for user-id), password

Description – This module will enable a user/admin to delete his/her electronics mart account. If the password is matched with the encrypted password in the database after decryption, then the user/admin account will be deleted, and this user/admin data will be stored in another collection for 1 year. & key “deleted\_by” will have a value “USER\_SELF”.

The user’s cart will be permanently deleted, all the orders done by this user will be updated and a key will be added to all his/her orders-

"ACC\_DELETED": "this user's account has been deleted"

Output - If delete is successful, user/admin will get a successful prompt and a mail on the registered mail id & user/admin will be redirected to the home page. Else, the user/admin will get an unsuccessful message.

View all users

Input –

Description – An admin can view all the users in the database who have registered on the site. An authorized admin can also view all admins.

Output – If users are found in the database, all the users will be displayed on the page. In case, there is an error or users are not found, the admin will get an unsuccessful prompt message.

View by user-id

Input – user-id

Description – An admin can view user details by searching his/her user-id in the database. An authorized admin can also view admin with admin id.

Output – If a user is found in the database with that user-id, the user details will be displayed on the page. In case, there is an error, or a user is not found, the admin will get no such user found prompt message.

View by email id

Input – email id

Description – An admin can view user details by searching his/her email id in the database. An authorized admin can also view admin with that email id.

Output - If a user is found in the database with that email id, the user details will be displayed on the page. In case, there is an error, or a user is not found, then the admin will get no such user found prompt message.

View by name

Input – name

Description – An admin can view the user's list by searching their names. all the users having similar or related names will be displayed in the list. An authorized admin can also view all admins with that name.

Output – If users are found in the database with that name, the user's list will be displayed on the page. In case, there is an error or users are not found, the admin will get no such user found prompt message.

View by pin code

Input – pin code

Description – An admin can view the user's list by searching the pin code, all the users having that pin code in address will be displayed in that list.

Output – If users are found in the database with that pin code, the user's list will be displayed on the page. In case, there is an error or users are not found, then the admin will get no such user found prompt message.

View by city

Input – city

Description – An admin can view the user's list by searching the city, all the users having that city in address will be displayed in that list.

Output – If users are found in the database with that city, the user list will be displayed on the page. In case, there is an error or users are not found, the admin will get no such user found prompt message.

View by state

Input – state

Description – An admin can view the user's list by searching the state, all the users having that state in address will be displayed in that list.

Output – If users are found in the database with that state, the user's list will be displayed on the page. In case, there is an error or users are not found, the admin will get no such user found prompt message.

Delete any user/admin

Input – user-id/admin id, token

Description – An authorized Admin can delete any user/admin by entering the user’s user-id or admin’s admin id if that user/admin violates the T&C of the website. The user/admin will get an email if the account is deleted by an admin. And the user/admin data will be saved in another collection for 1 year.

Admin’s data like admin id, name, email from token will also be saved who has deleted this user/admin account in that another collection for reference.

The user’s cart will be permanently deleted, all the orders done by this user will be updated and a key will be added to all his/her orders-

"ACC\_DELETED": "this user's account has been deleted"

Output – On successful deletion, the admin will get a successful message. Else an unsuccessful message

Authorize admin

Input – token, admin id

Description – an authorized admin can authorize any unauthorized admin if required, the admin’s name, admin id, email id who authorized, will be saved in the newly authorized admin document in the database for reference. After the successful update, the newly authorized admin will get a confirmation email.

Output – if admin is authorized then a successful prompt message will be displayed & newly authorized admin will get an email, if the process was not completed due to any issue unsuccessful prompt message will be displayed.

Unauthorize admin

Input – token, admin id

Description – an authorized admin can unauthorize any authorized admin if required, the admin’s name, admin id, email id who unauthorized, will be saved in a newly unauthorized admin document in the database for reference. After the successful update, the newly unauthorized admin will get a confirmation email.

Output – if admin is unauthorized then a successful prompt message will be displayed & newly unauthorized admin will get an email, if the process was not completed due to any issue unsuccessful prompt message will be displayed.

Products Modules

Add product

Input – product name, category, price, image URL, Description, quantity

Description – An admin with all the input details can add a product in the database in which a unique product id is generated by a unique id, product rating & rated by is by default 0.

Output – If products are added successfully, the admin will get a message product is added. Else, a failure message.

View all products

Input – rating (optional)

Description – All the products available in the database will be displayed. If rating Is also mentioned, then the product more than that rating will be displayed otherwise products above rating 0 will be displayed.

Output – The user will be displayed all the products. In case products are not displayed, the user will get a failure prompt.

View by interest

Input – Interested in categories from user data by token, rating (optional)

Description – During the registration process, the user is asked for selecting favorite categories. The products belonging to those categories will be displayed. If rating Is also mentioned, then the product more than that rating will be displayed otherwise products above rating 0 will be displayed.

Output - The user will be displayed all the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by name

Input – the name of product, rating (optional)

Description – The user can view a product by searching for the product name. the name of the product is searched in the database with case-sensitive syntax for best results. If rating Is also mentioned, then the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output - The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by a single category

Input – the name of the category, rating (optional)

Description – The user can view a product by searching for the product category. If rating Is also mentioned, then the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output - The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by multiple categories

Input – the name of all the categories, rating (optional)

Description – User can view a product by searching for multiple product categories. If rating Is also mentioned, then the products more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output - The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by price

Input – price range, rating (optional)

Description – The user can view a product by selecting a price range. If rating Is also mentioned, then the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output - The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by rating

Input – rating

Description – The user can view a product by selecting a rating.

Output - The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

Search by name in a category

Input – name, category, rating (optional)

Description – The user can search for a product by entering a name in the search box and selecting a category. If rating Is also mentioned, the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output – The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

Search by name in multiple categories

Input – name, categories, rating (optional)

Description – User can search a product by entering a name in the search box and selecting multiple categories. If rating Is also mentioned, the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output – The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by price & categories

Input –price, categories, rating (optional)

Description – The user can view a product by selecting a category and a price. If rating Is also mentioned, the product more than that rating will be displayed otherwise products above rating 0 will be displayed and there will be no change in products on the screen.

Output – The user will be displayed the products. In case products are not displayed, the user will get a failure prompt and there will be no change in products on the screen.

View by product id

Input –product id

Description – User can view a product detail by this feature, the front-end application will automatically send a product id when any product is selected by the user to view.

Output – The user will be displayed the products. In case products are not displayed, the user will get a failure prompt.

Rate product

Input – product id, rating

Description – In this module a user can rate a product once the product is delivered to his address. user details like user-id will be retrieved from token, this module also checks whether that user has already rated that product or not, if not then the average rating of that product is updated by simple Mathematical equation, and product rated by is also increased by 1. The rating given by the user is saved in the user’s order for future reference if the user wants to change the rating given by him.

Output – If the rating is successful, the user will get a successful prompt. Else if a product is already rated, then the user will get an already rated prompt message. Or else an unsuccessful prompt.

Edit rating

Input – product id, rating

Description – User can only edit rating given by him/her earlier, user details like user-id will be retrieved from token, his previous rating is obtained from product order as saved earlier and the average product rating will be updated by a simple mathematical equation which will be deleting the previous rating from average and adding a new rating.

Output – If the rating is successful, the user will get a successful prompt. Else if a product is not rated earlier, then the user will get a product not rated prompt message. Or else an unsuccessful prompt.

Review product

Input – product id, review

Description – In this module, a user can review a product once the product is delivered to his address. user details like user-id will be retrieved from token, any review will be added to the list of the reviews of the product which stores a review mapped with that user’s id.

Output – If the review is successful, the user will get a successful prompt. Else if a review is already added, then the user will get an already reviewed prompt message. Or else an unsuccessful prompt.

Edit review

Input – product id, review

Description – The user can only edit a review of a product given by him earlier by adding a new product review which will overwrite the previous review given by him. user details like user-id will be retrieved from the token.

Output – If the review is successful, the user will get a successful prompt. Else if a product is not reviewed, then the user will get a product not reviewed prompt message. Or else an unsuccessful prompt.

Order Product Modules

Order product

Input –Token, product id, product quantity

Description –When a user orders a product, order details like product id, user-id, quantity and retrieve email address, full name, user-id from token, unique order id is generated by unique id despondency & stored in the orders collection in the database and from the products database, product quantity is reduced.

Output – If an order is successful, the user will get a successful prompt and a mail on the registered id with the order in the mail body, and if the order can’t be placed, the user will get a prompt that the order is unsuccessful.

View One Order (by user)

Input – order id, token (for user-id)

Description – A user can check the status of order with the order id which is automatically filled by the front end. The application will check the order id with that user-id from the orders database.

Output – If an Order is found, the user will get all the order details on the screen. Else if there is a mistake in the order id, the user will get an order not found prompt.

View orders (by user)

Input – token (for user-id)

Description - A user can check all the orders. The application will return all the orders placed by that user-id.

Output– If Orders are found, the user will get all the order details on the screen. If unsuccessful, the user will get an order not found prompt.

Change order status (by user)

Input – token (for user-id), order id, new order status (CANCEL/RETURN)

Description – A user can change the order status to CANCEL/RETURN of the order if he wants to cancel or return an order. Mail is also sent to the user email giving details about the new order status. If an order is canceled the product quantity is increased by the quantity of product in this order.

Output – If the status is changed, the status will get changed. Else the user will get a prompt that it is unsuccessful.

View all orders (by admin)

Input – none

Description – An admin can view all the orders that are placed by all the users combined

Output- If successful, all the orders will be displayed, else the admin will get an unsuccessful prompt message on the screen.

View orders by the user (by admin)

Input – user-id

Description – An admin can view all the orders placed by a particular single user in this module.

Output – If successful, all the orders will be displayed, else the admin will get an unsuccessful prompt message on the screen.

View orders by product id (by admin)

Input – product id

Description – An admin can view all the orders placed for a specific product.

Output – If successful, all the orders will be displayed, else the admin will get an unsuccessful prompt message on the screen.

View single order (by admin)

Input – order id

Description – An admin can view a specific order with details by looking for the order id in the database.

Output – If successful, order will be displayed, else admin will get an unsuccessful prompt message on the screen.

Search Order by order status

Input – order status

Description – Admin can view all the orders with order status.

Output – If successful, all the orders will be displayed, else admin will get an unsuccessful prompt.

Update order status (by admin)

Input – order id, new order status

Description – Admin can change the order status of an order as described in the requirement. The user will get an email about an update in order status. If an order is canceled the product quantity is increased by the quantity of product in this order.

Output - If successful, order status will be changed for both user and admin and the user will also receive a mail with updated status. Else admin will get an unsuccessful prompt.

Cart Modules

Add to cart

Input – product object (containing product id & quantity), token

Description – an empty user cart is already created at the time of user registration; cart id is the same as user-id for efficiency & easy mapping that will be stored in cart object. product object is pushed into products array of cart object. A cart object has an array that contains product objects. If the user click adds to cart it first checks whether that product is in stock for that much quantity user-selected & then looks for that product is already in its cart of that user or not. If it is existing in the array, then it will simply increase the quantity of that product in the cart else that product is pushed into the product's array.

Output – If the product is added to the cart, the user will get a successful prompt. Else if the product is not in enough quantity user will get a product out of stock prompt. Or else an unsuccessful prompt.

View my cart

Input – token

Description – The user can view his cart items with the help of this module.

Output – User will be displayed cart. In case the cart is not displayed user will get an unsuccessful prompt.

View one cart item

Input – product id, token

Description – cart id is the same as user id which will be retrieved from token. Users can view a detailed view of any cart item.

Output – User will be displayed that cart item. If not, then the user will get an unsuccessful prompt.

Search in cart

Input – name, token

Description – User can search any product in his/her cart by name.

Output – user will be displayed that cart item. If there is no product in the cart with that named user will get a not found prompt. Or else an unsuccessful prompt.

Delete cart item

Input – product id, token

Description - cart id is the same as user id which will be retrieved from token & the product with this product id which is selected by the user will be deleted (pop) from the products array of the cart.

Output – If a cart item is deleted user will get a successful prompt. Or else an unsuccessful prompt.

Delete user cart

Input – token

Description - cart id is the same as user id which will be retrieved from token. Users can delete their all-cart items at once if needed with the help of this module.

Output - if all cart items are deleted user will get a successful prompt. Or else an unsuccessful prompt.

Newsletter Modules

Activate Newsletter Service

Input – email address

Description – After users enter the email address, a function checks whether the email address is present in the newsletter collection of the database. If not present, it will store emails in the collection and activate the service & it will mail the user with a success message.

Output – If activated it will prompt a successful message, Else if the service is already activated, it will prompt the user that the service is already activated.

Send Newsletters

Input – Subject, Body, Email addresses, token

Description – An admin (after being verified through token) can add subject, body, and send the newsletter to all the email addresses which are present in newsletter collection & to the emails of all registered emails (will not send an email twice if an email is in both collections). After sending it will save the newsletter, list of recipients & who send this email in another collection.

Output – If successful, it will prompt the admin that emails were sent successfully sent, else it will prompt those emails were not sent.

View All

Input – none

Description – An admin can check all the emails that were sent earlier to the users.

Output – This module returns all the emails that were sent to the users as the output

Common Modules

Authentication

Input – token from headers under authorization key.

Description – it checks whether the token is empty or not, if not empty it verify the token with the help of another module.

Output – if the token is valid, the user can go further, else the user will be displayed an unauthorized prompt & redirected to the login page for authentication.

Two-factor authentication

Input - token from headers under authorization key.

Description – If the user has activated two-factor authentication & has verified the latest OTP only then this module will let the user access the website features that require user data. if two-factor authentication is not activated the user can go further.

Output – if OTP is not verified user will get an OTP not verified prompt & will be redirected to the OTP verification page. Else user can access services.

Admin authentication

Input - token from headers under authorization key.

Description – This module verifies whether a person is an admin or not with the help of verify admin module. if get success through verify admin module that means this is an admin who can access admin functionalities.

Output – if not the user will get an unauthorized prompt & redirected to the login page.

Admin authorization

Input - token from headers under authorization key.

Description – This module verifies if this admin is authorized or not with the help of verify authorized admin module. If it gets success through verify authorized admin module that means this is an authorized admin who can access all admin functionalities.

Output - if not, the admin will get an unauthorized prompt & redirected to the login page.

Encrypt

Input – password or encrypted password

Description – this module has 2 functions. One function is for converting passwords into encrypted passwords & another one will decrypt & check the encrypted password with the password for security & verification.

Output – it will return encrypted password for password as input, true or false based on comparison after decryption.

Nodemailer

Input – email, subject, body

Description – this module is used for mail service.

Output – sends mail or if there’s some error it will show an error in the console.

Generate token

Input – user details, secret string

Description – by using 3rd party dependency (JWT) this module will generate a token from user details & a secret string, this secret string should be unique so that the token should not conflict with any other website token. This token will expire in 2 hours for better security.

Output – unique token

Verify Token

Input – token, secret string

Description – this module will verify the user/admin by token after decoding it with the help of 3rd party dependency, will check whether the token contains user/admin object & our secret string and should not be expired.

Output – if the token is correct and not expired it will return true, else if the token is invalid in any way or expired it will return false.

Verify Admin Token

Input – token, secret string

Description – This module will verify the admin by token after decoding it with the help of 3rd party dependency, will check whether the token is containing an admin object & our secret string and should not be expired as well.

Output – if the token is containing the admin object & is not expired then it will return true, else if the token is invalid or is expired, it will return false.

Verify Authorized Admin

Input – token, secret string

Description – This module will verify the authorized admin by token after decoding it with the help of 3rd party dependency, will check whether the token is containing an admin object with “isadmin” key equal to “1”, our secret string, and should not be expired.

Output – if the token is correct & isadmin is equal to 1 and not expired then it will return true, else if it’s invalid, isadmin is not equal to 1 or not found or is expired then it will return false.

Generate key

Input – secret string

Description – This module helps to create a unique key with the help of 3rd party dependency for functionalities like account activation or account recovery where a unique key is required & should get expired in a limited time after 2 hours.

Output – If the key is generated successfully then it returns a unique key, else at least possible it may give some error.

Verify key

Input – key, secret string

Description – This module will verify the key with the help of 3rd party dependency whether it is correct or not or it is containing the secret string & should not be expired.

Output – if the key is correct & not expired it will return true else if the key isn’t correct or is expired it will return false.

Get doc

Input – token, secret string

Description – this module will decode the token with the help of 3rd party dependency. If the token is correct, then it will return the user/admin object containing their information. By this, we are reducing I/O operations through the database which will improve performance & reduce loading time also.

Output – If the token is correct & not expired, it will return the complete user/admin object, else error

# Other Requirements

Electronics Mart shall handle expected & unexpected errors in ways that prevent loss in information & long downtime period.

Appendix A: Glossary

* ODM – object data manipulation which is basically manipulating data when it's transferred from database to application & vice-versa.
* Middleware – It’s a functionality which can be added to the application anywhere it’s required whose task is to perform a specific action.
* Unauthorized Amin – basically these are new admins who have rights to only read database like they can view users, products, orders.
* Authorized Admins – these admins have full rights to do anything, they can register any admin, authorize/unauthorize any admins, manage products, modify order status, manage users, etc.
* Token – token is used to verify the correct user as we are using HTTPS protocol which is connectionless approach, this token contains an object of user details, so that we can get user info more quickly from this token rather than doing database operation for each & everything.

Appendix B: Analysis Models

Electronics Mart Database

