

Increment 2 Report

Plan Your Shopping

Project Group 18

Vikas Kondapalli (39)

Swatvik Gunamaneni (27)

Gopi Kirshna Bodapati (8)

Introduction

Our project is a website which allows its users to plan their shopping well in advance before they actually go the stores. The website accepts the input from the user where he specifies the product he is looking for. Then the system displays a list of all the stores in the close proximity to the user with the distance from the user and the price of the product at each store. The user then chooses the product from the store of his choice considering the price and also the distance from his place. He can add this choice to his cart and start all over again searching for a new product. He continues in this fashion until he finds all the products he needs to shop for. Finally on completing his search he has all the items added to his cart. The cart gives him the total amount he is supposed to spend in buying these items and also segregates the list based on the store each item has to be brought from.

The major progress we have made from the first increment to this increment is our move towards implementation after the discussion and planning phases. For this phase we have come up with login and registration pages where the user on his first visit gets himself registered with the website. Then he tries to log in. We have connected the login and register pages to the database MongoDB using the Data API offered by MongoLab. So initially when the user registers himself, his details get stored in the Database and when he tries to log into the website his credentials are validated against the ones already stored in the database. After the user successfully logs into his account he is directed to his dashboard, which is the actual workplace where he performs search and operations and gets the results displayed. In this phase we are accepting inputs from the user like his State and City and then we display the list of all stores close to him. He then chooses a store and searches for the price of an item he needs at that store and then has a list of the same item from different producers with their prices.

Features

When the user visits the website he has two options. If he is a first time visitor he can move to the registration page in order to sign up. In the registration page he can give all the details he is asked for and create his account. All the details he gives are stored in the MongoDB database and are used to validate him when he tries to log into his account in the future. He then can move to the log in page and access his account.

If the user already has an account he can simply give his username and password to log into his account. The username and password will be validated against the ones in the database which he gave while creating the account. After successfully validating these details with those in the database he gets directed to his dashboard. All the interactions with the database have been carried out using the Data API.

The dashboard is like the home page of the user's account. Here the user chooses his state and city from a drop down box. For the current increment we have included only the cities in Missouri state and for the coming increment we will include all the other states and their cities. On giving his choice of state and city the user now gets a list of all the stores in the city he chooses as a result. This is accomplished using the Supermarket API.

Now the user selects a store and enters the store id of this store and the item he is searching for. Then he gets a list of the same item from all available manufacturers available at that store with the price in each case. Even this service is enabled using the Supermarket API.

In the next increment we will introduce speech input where the user can search for the products using voice inputs in addition to the existing text inputs from the drop down boxes.

Import Existing services/API:

SuperMarketAPI:

In this project (plan your shopping), the challenging part is to provide dynamic detail of product in store to user on daily basis. The main purpose is provides information about prices of item in store for this it necessary to retrieve data from store. To sever this, we have use super market API. Super market API is an open source API which present information about 9000 supermarket across USA includes Walmart, Costco etc. API allow user to search and retrieve item based on product name, item id, state zip code and store name.

Link on web: <http://supermarketapi.com/Default.aspx>

Google Maps API:

Other task in the project was to present the route direction to the store for this purpose we have use google maps API so, that it would be convenient to user to get item from the store in addition to this it provide number from the present location to stores. This help user to know which store is near which one is far. Google map API is open source API which display map based on the input provided by user. After selecting item from home page on click on direction button it direct to webpage where user is present with direction from current location to desired store. This will repeat for every selection of product from different stores.

Link for Google Maps API used in this project:

<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

MongoDB

To data we have use MongoDB which is store data in document oriented manner,it also popularly called as “NOSQL” un it can like traditional table based relational Database it has a flexible schema. Some of feature it includes are document-oriented, Adhoc queries, indexing, replication, load balancing, File storage. It important feature is that having replica data which store data in different machine provides high availability. Each user has separate account to store data. User can store use collection which is similar to table in relational database. MongoDB is enriched with data API which are default in disable state it can be used to get data from database, collection and document. Every user has its own API key that can used to access the APIs.

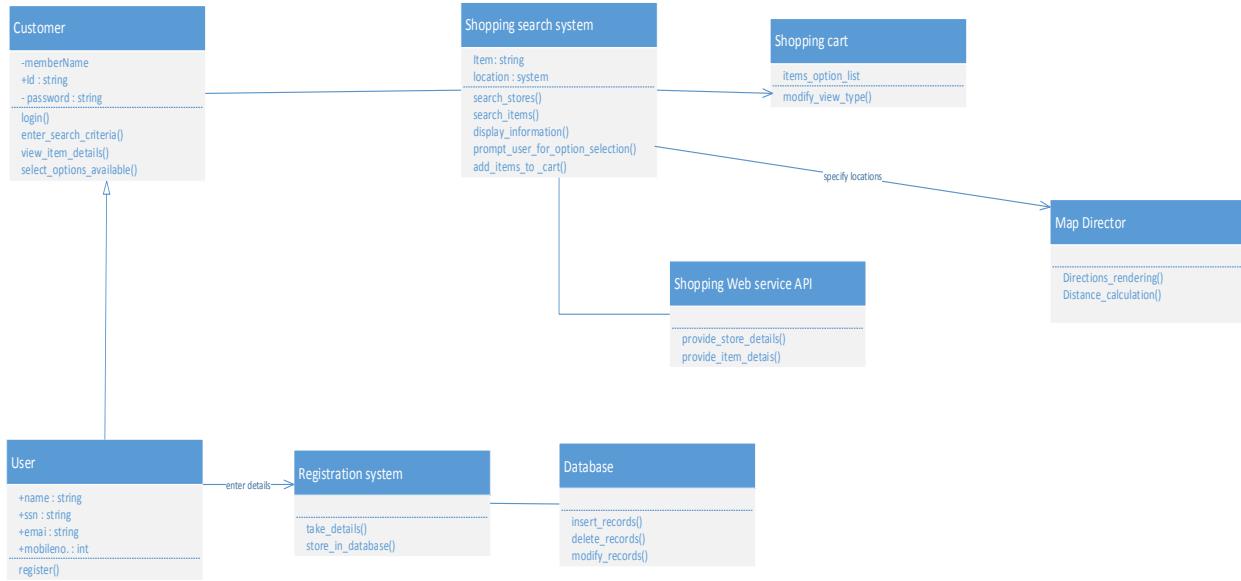
Link for MongoDB Data API used in this project:

https://api.mongolab.com/api/1/databases/vikas2/collections/users?apiKey=HxPILPvmcIj3SyZBMrPV38t7_BiBKUJ

Detail Design of Services (using tools)

UML Diagrams:

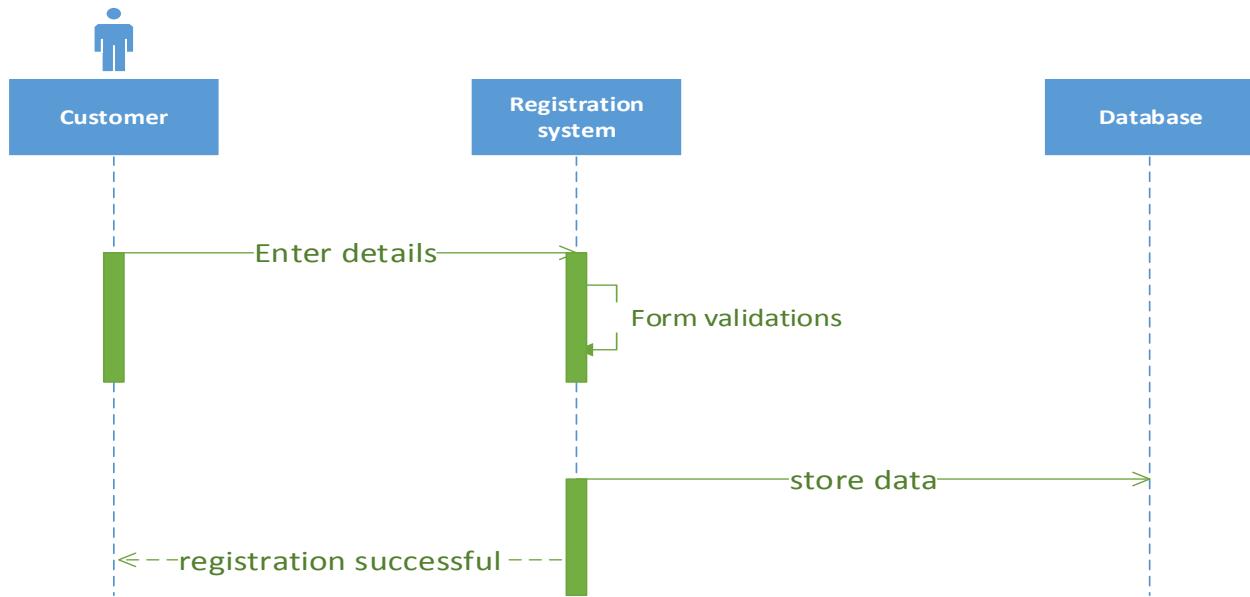
Class Diagram



In this class diagram customer, user, shopping system, cart, Registration are class. Here Customer is the super class of the user i.e for the normal user to become the customer of this application he has to register and store his details in database. Every user interact at least one registration class and detail of it store in Database. Customer then interacts with Shopping search system class to serach for the required stores and items. Shopping system class has websevice API which is a class based on information provided by user. Shopping class is relate to Map director to locate store.

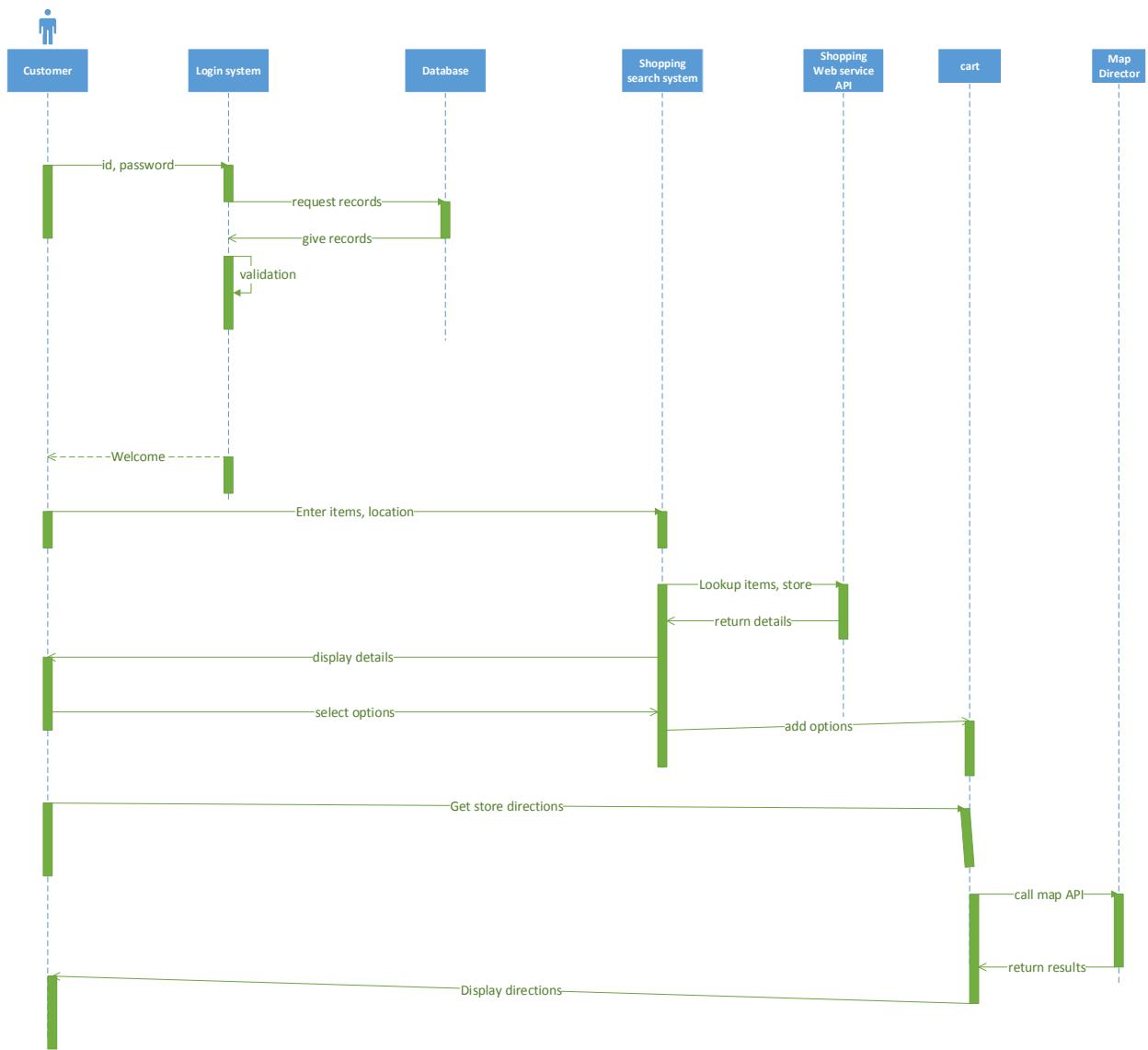
Sequence Diagram

For registration



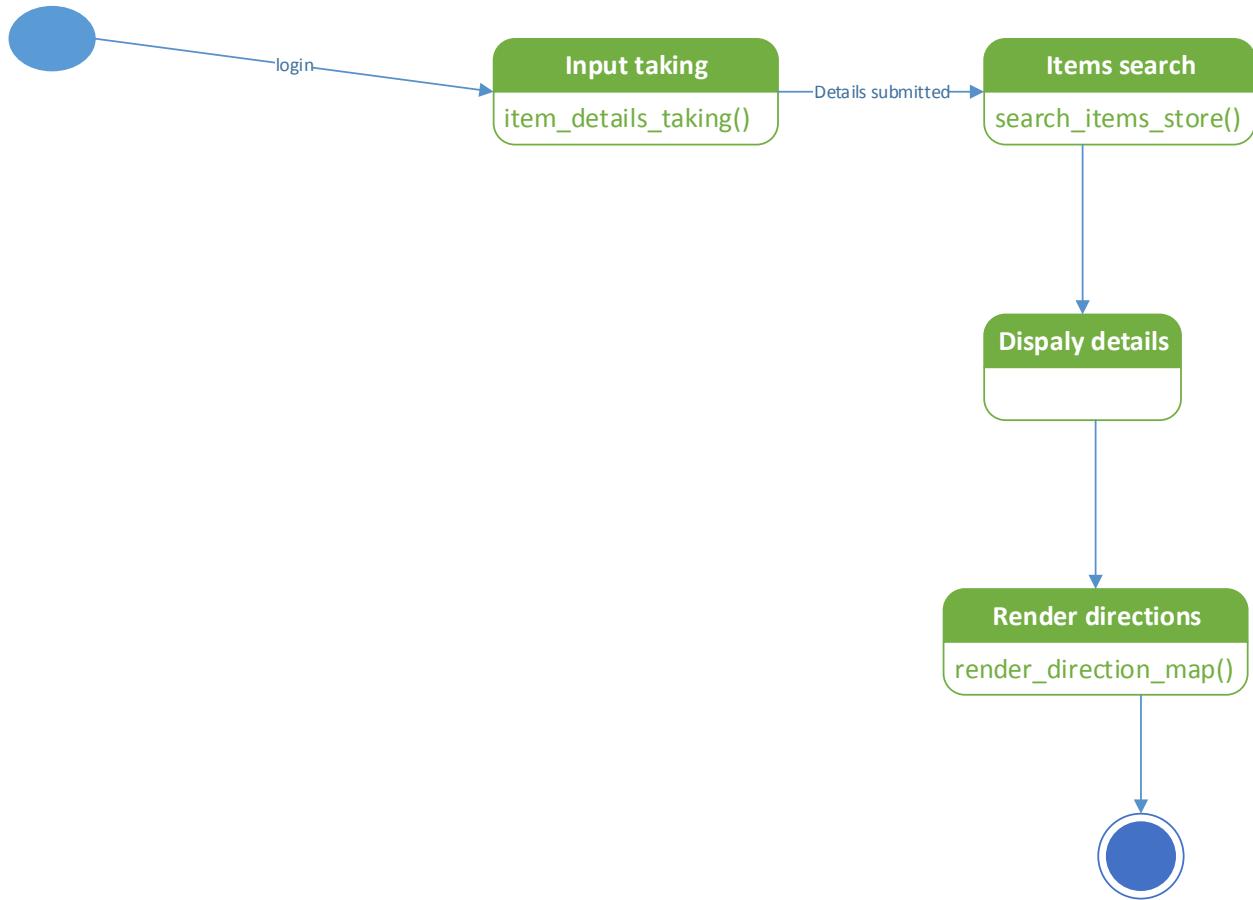
In this sequence diagram message are pass between actors. Customer present information about that is require for registration which is validates by registration page then store into database after storing detail returns successful message to customer.

For Search and display service



Customer login into system by provide information system then validate by comparing information registration provide by user. User in then navigate to searcher system where use is allow to select product and stores by using various API. After selection of item user is allow to add item into cart and show location of store in map.

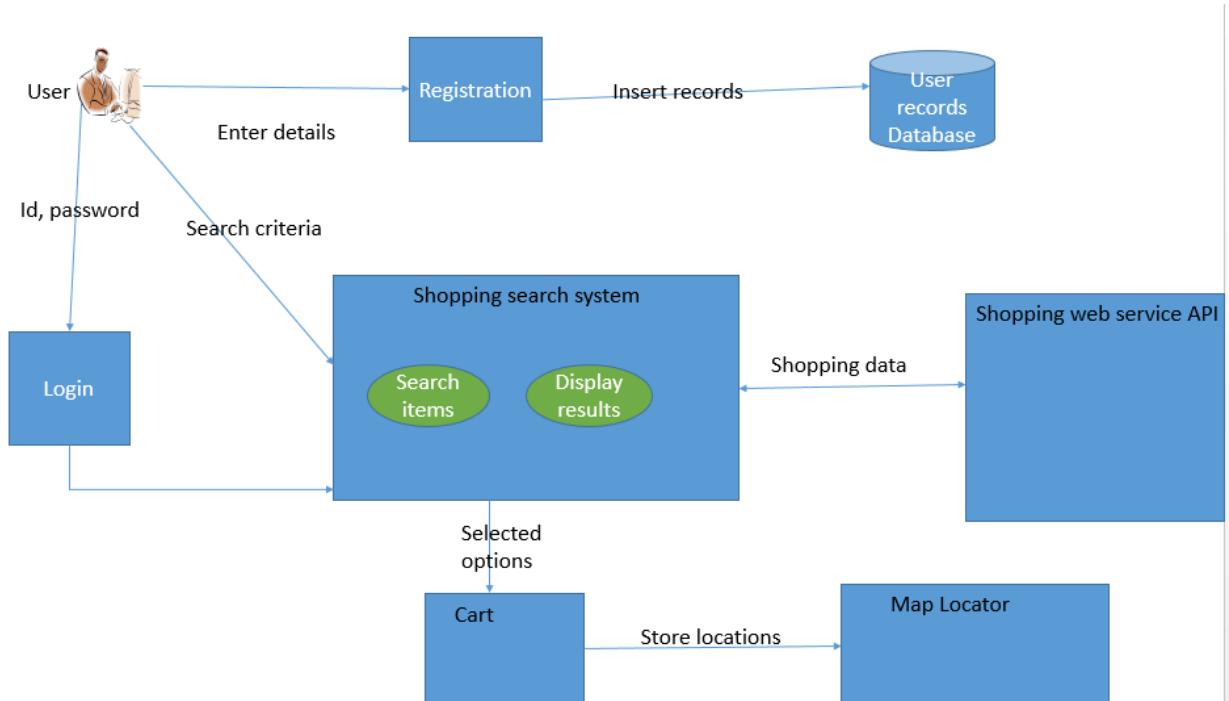
State Diagram



In state Diagram user is direct to system after providing valid login detail, each component in system is represent by state and states are controlled by either by external or internal event such as on selecting item from store system display details later shows direction to store specified by user.

After the login system enters into input_taking state where it accepts search input. After the input is read it enters into Searching state. Then it enters into Display_details state and Render_directions state.

System Architecture Diagram

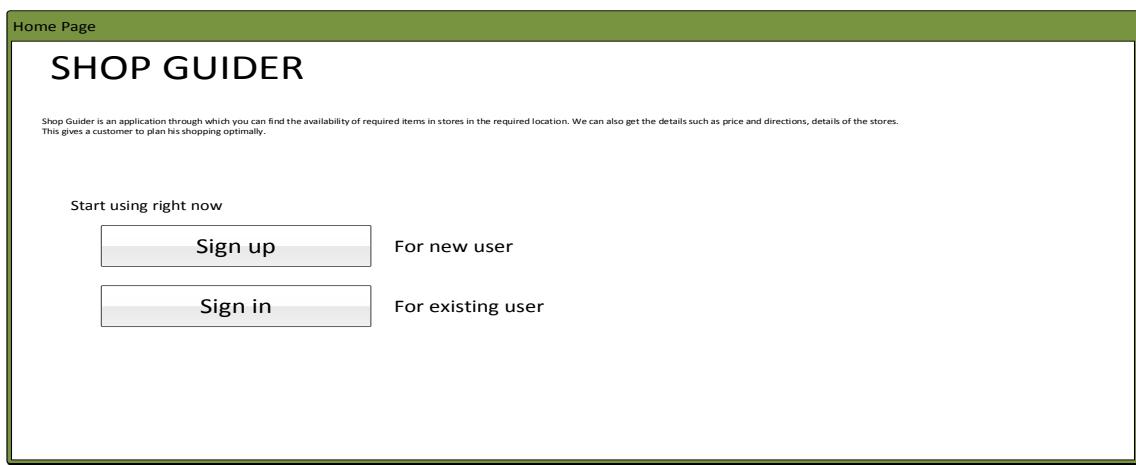


This system architectural overview where system is interacting with various API services to present information to user. New user access system by registrations mandatory information which is store in Database. Current user can use system through login to system to get desires data for example get prices of item then item is add into cart. Shopping search system has two processes search items and display results. Store location use map locator API is used to get detection to particular store.

WireFrames and Mockups:

Wireframes:

Home page



This is the home page where user need to provide valid information in order to access the service and for new user there will be redirect to registration page on clicking sign up button. Here we have use to two button sign in button user is sail to login page

Registration Page

Registration Page

Enter details for Sign up and Registration

First Name	<input type="text"/>	Last Name	<input type="text"/>
Email or Mobile no.	<input type="text"/>		
Re-enter Email or Mobile no.	<input type="text"/>		
Password	<input type="text"/>		
Re-enter Password	<input type="text"/>		

New User has to register by providing some of the personal detail on clicking button “create account” information of the particular user stores in database. Here user first name is used as login id. Here user need to enter their email id or mobile number, first name, last name, password in order to register into service.

Login Page

The diagram shows a wireframe of a login page. At the top, there's a green header bar with the text "Login Page". Below it is a white main area. In the center, the text "Sign in for Login" is displayed. Underneath this, there are two sets of labels and input fields: "Email ID" followed by a text input field, and "Password" followed by another text input field. At the bottom of this section are two buttons: "Forgot my password?" and a larger "Login" button.

Login is starting page of service where user need to provide user id and password to get into service. Hyperlink will redirect to “forget password” where user can retrieve password stored in database. On providing valid login id and password service will redirect to same page and passes invalid message to the user.

Dashboard

The screenshot shows a user interface titled "Dashboard". At the top, there is a header bar with the title "Dashboard". Below the header, the main content area is titled "Welcome User". It contains three input fields: "Enter Item:" with a placeholder "Enter Item", "Enter Location:" with a placeholder "Enter location", and a "Get details" button. Below these fields is a table with four columns: "Item", "Store name", "cost", and "location". The table has 10 rows, each with a checkbox labeled "Check to select" in the "location" column. At the bottom of the table is an "Add to cart" button.

System allow user to selecting of item or product in dashboard page in additional to this user are allow to select store or supermarket on clicking to button user redirect to cart page. In this page user are allow to add or delete product from list. On clicking button “Get Detail” detail of item in select store is show in the list so that user can select item based on their budget plan.

Cart Page

In cart page user get information of product selected by individual it also including cost and store he/she need to get from. After user is navigate to map locator page. Here selected item are dynamically add to list along with detail of products. The button sort by cost allow user to get list in ascending order based on prices of item.

Map Locator Page

Map Locator

[Get directions for](#)

[location1](#)

[location2](#)

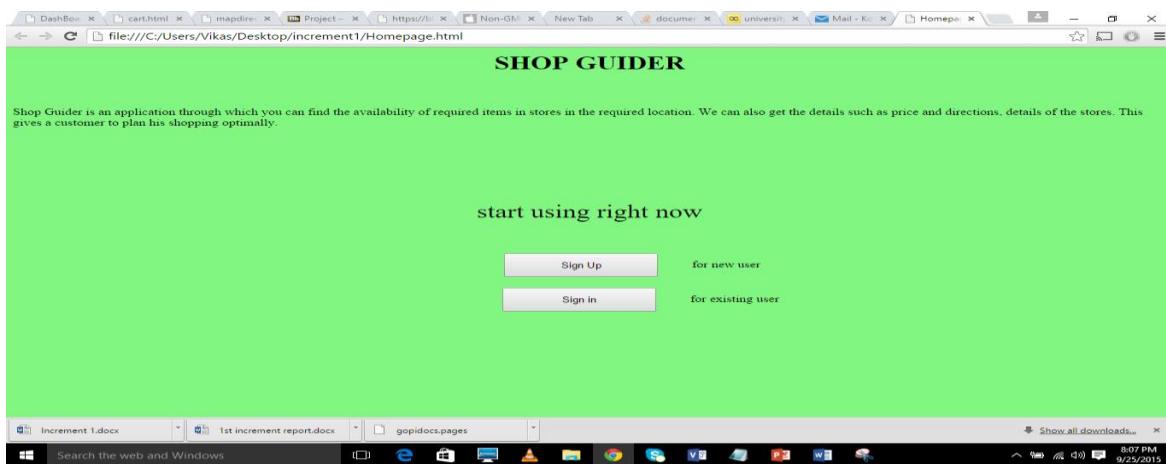
[location3](#)

Map Output

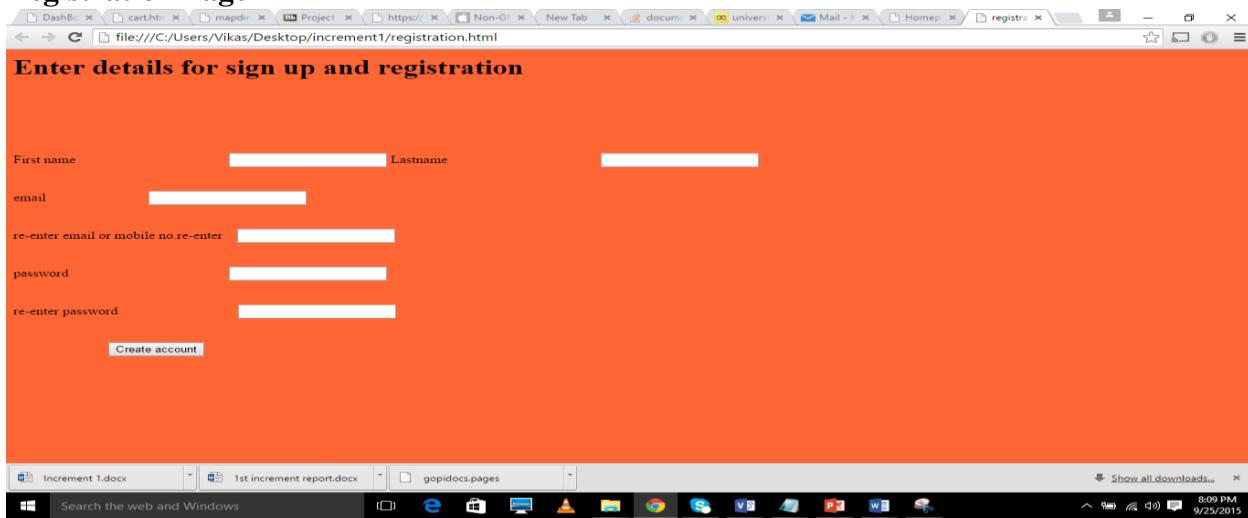
In this system provide location of store and distance from his/her current location. In this user is displayed with short distance from his/her current location to particular store so, user can save time which is prime motive of project.

Mockups

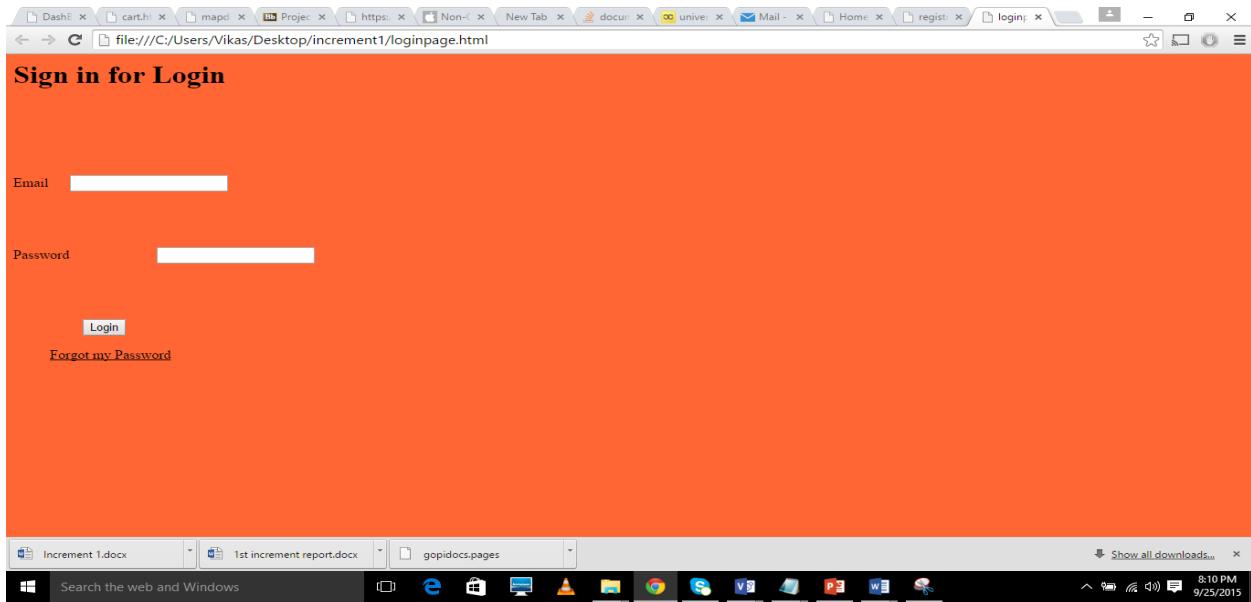
Home page



Registration Page



Login Page



Dashboard page

Welcome user

Enter Item: Location: Get Details

Item	store name	cost	Address	Select
milk	walmart	\$10	overland park, kansas city	<input type="checkbox"/>
eggs	walmart	\$12	overland park, kansas city	<input type="checkbox"/>
bread	walmart	\$14	overland park, kansas city	<input type="checkbox"/>
tissues	walmart	\$16	overland park, kansas city	<input type="checkbox"/>

Add to Cart

The screenshot shows a web browser window with a title bar "file:///C:/Users/Vikas/Desktop/increment1/dashboard.html". The main content area displays a "Welcome user" message, input fields for "Enter Item" and "Location", and a "Get Details" button. Below these is a table with columns "Item", "store name", "cost", "Address", and "Select". The table contains four rows of data: milk (\$10, overland park, kansas city), eggs (\$12, overland park, kansas city), bread (\$14, overland park, kansas city), and tissues (\$16, overland park, kansas city). Each row has a checkbox in the "Select" column. At the bottom of the table is an "Add to Cart" button. The browser's address bar shows the local file path. The taskbar at the bottom includes icons for various applications like Word, Excel, and File Explorer, along with system status icons.

Cart Page

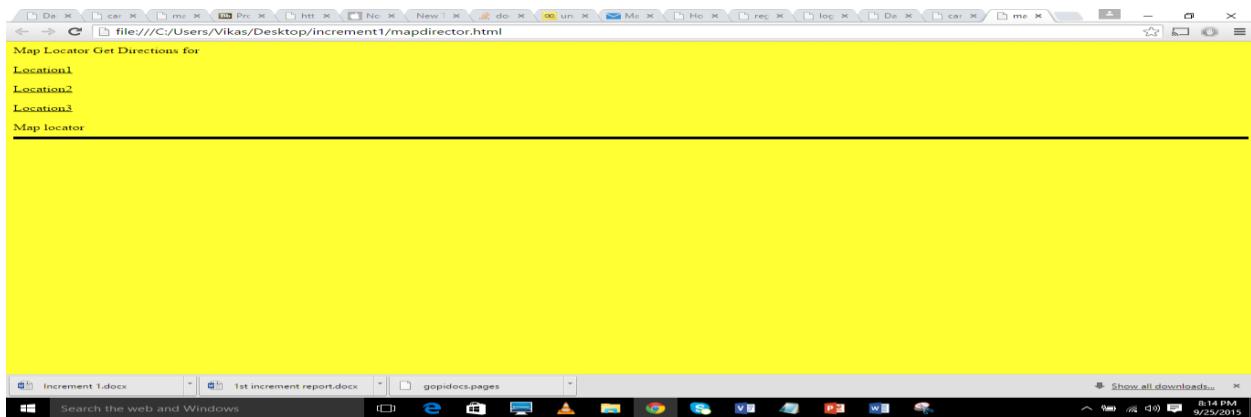
Your Selection

Item	store name	cost	Address
milk	walmart	\$10	overland park, kansas city
eggs	walmart	\$12	overland park, kansas city
bread	walmart	\$14	overland park, kansas city
tissues	walmart	\$16	overland park, kansas city

sort by Location
sort by Cost Go to map locator

The screenshot shows a web browser window with a title bar "file:///C:/Users/Vikas/Desktop/increment1/cart.html". The main content area displays a "Your Selection" message above a table of the same data as the previous screenshot. The table has columns "Item", "store name", "cost", and "Address". Below the table are two buttons: "sort by Location" and "sort by Cost", each with an associated input type="button". At the bottom right is a "Go to map locator" button with an input type="button". The browser's address bar shows the local file path. The taskbar at the bottom includes icons for various applications like Word, Excel, and File Explorer, along with system status icons.

Map Locator Page



STORYBOARD

In this increment

As a system, it must able to validate given login id and password provided by the user and displays message to the user

As a system, it must able to access detail from supermarket API and provide detail to user such as product id, store name, store location, store address, store id.

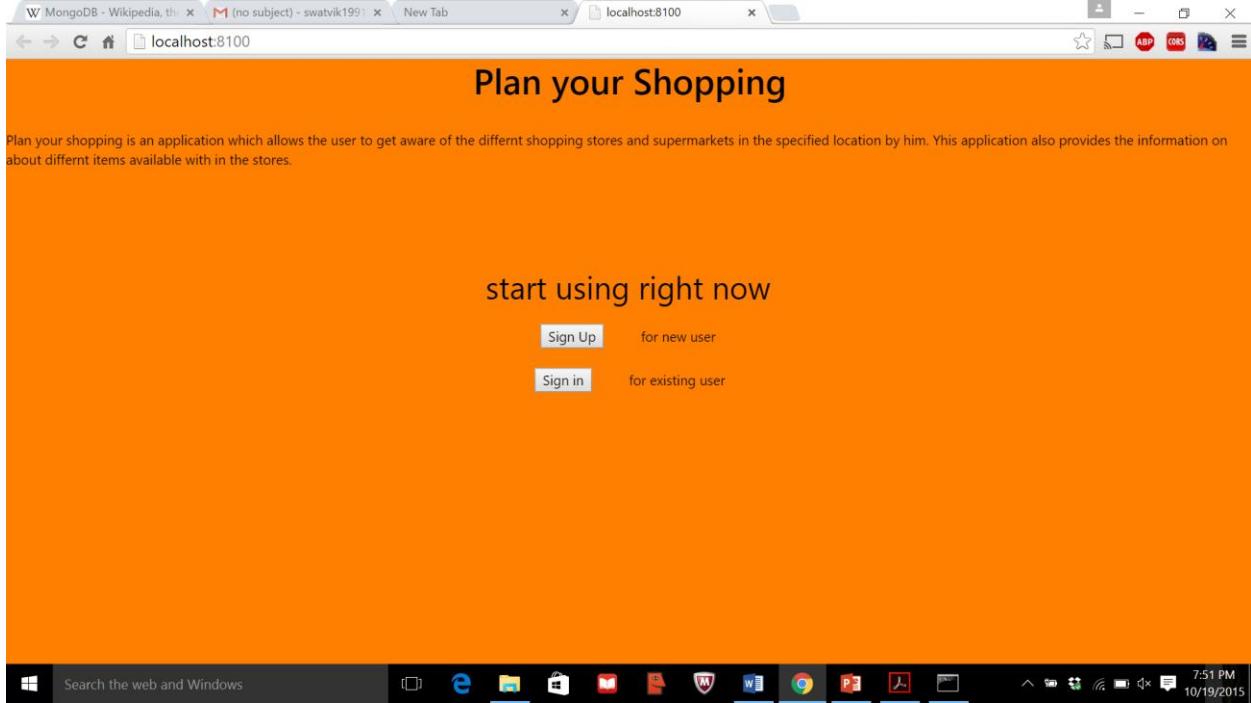
As a system, it must able to use google API and locate the store on the map that user has selected.

As a system, it must able to display name of user in dashboard by welcoming user with username.

As a user, there must allow to view list of item in particular store and able to select item from the list

Testing

Unit Testing (Manual)

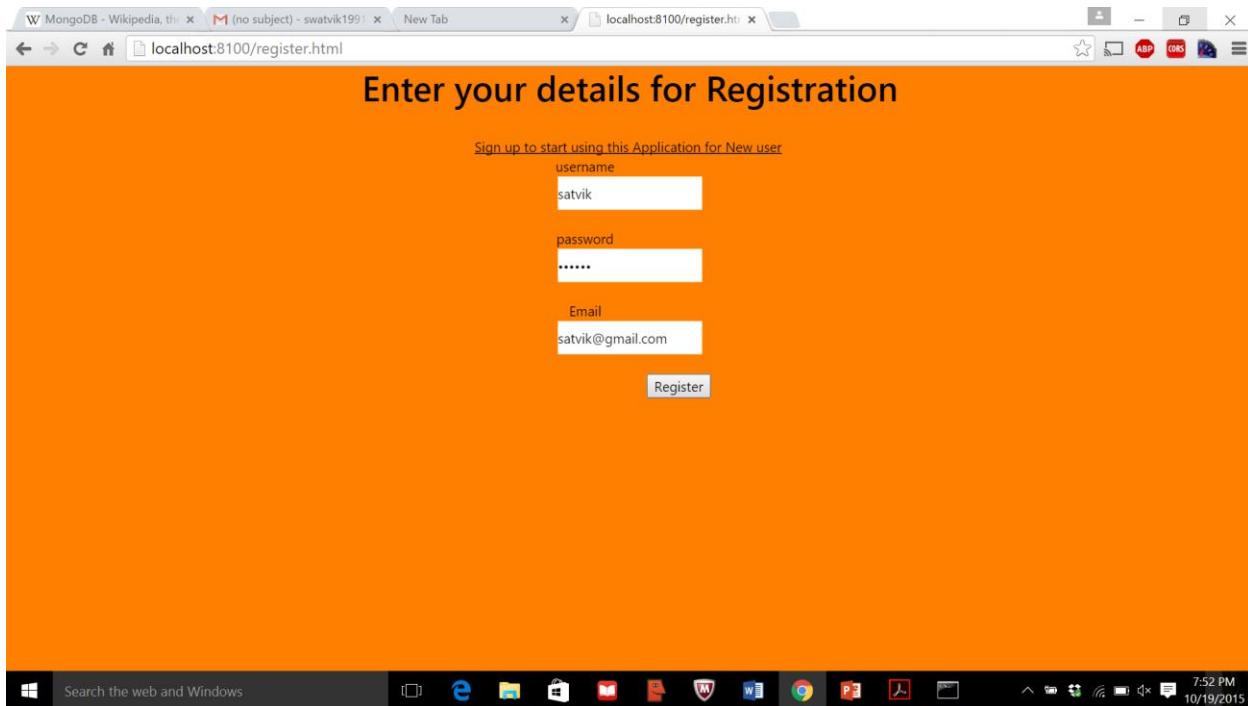


This is the home page of application with two buttons one to sign in and the other to sign up with the website.

Testing Sign up button

Expected Output: On clicking the Sign up button it should redirect the user to the Registration Page.

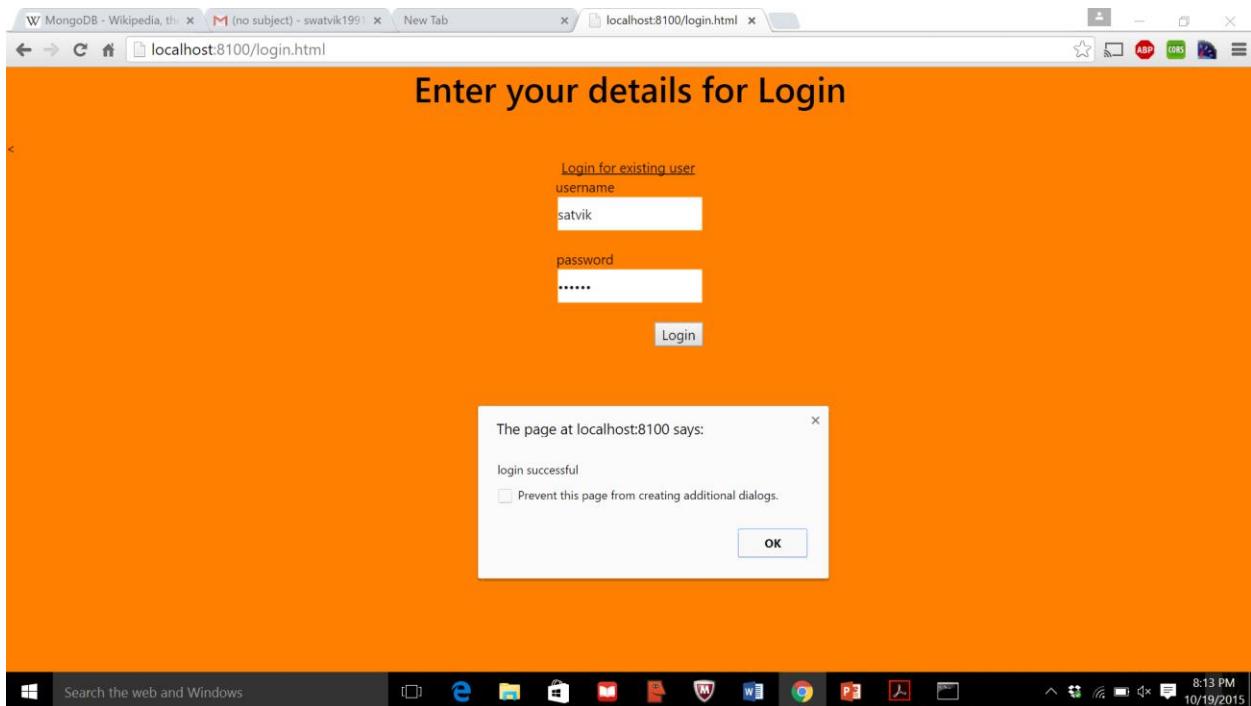
Actual output: On clicking Sign up button the user was redirected to the Registration Page.



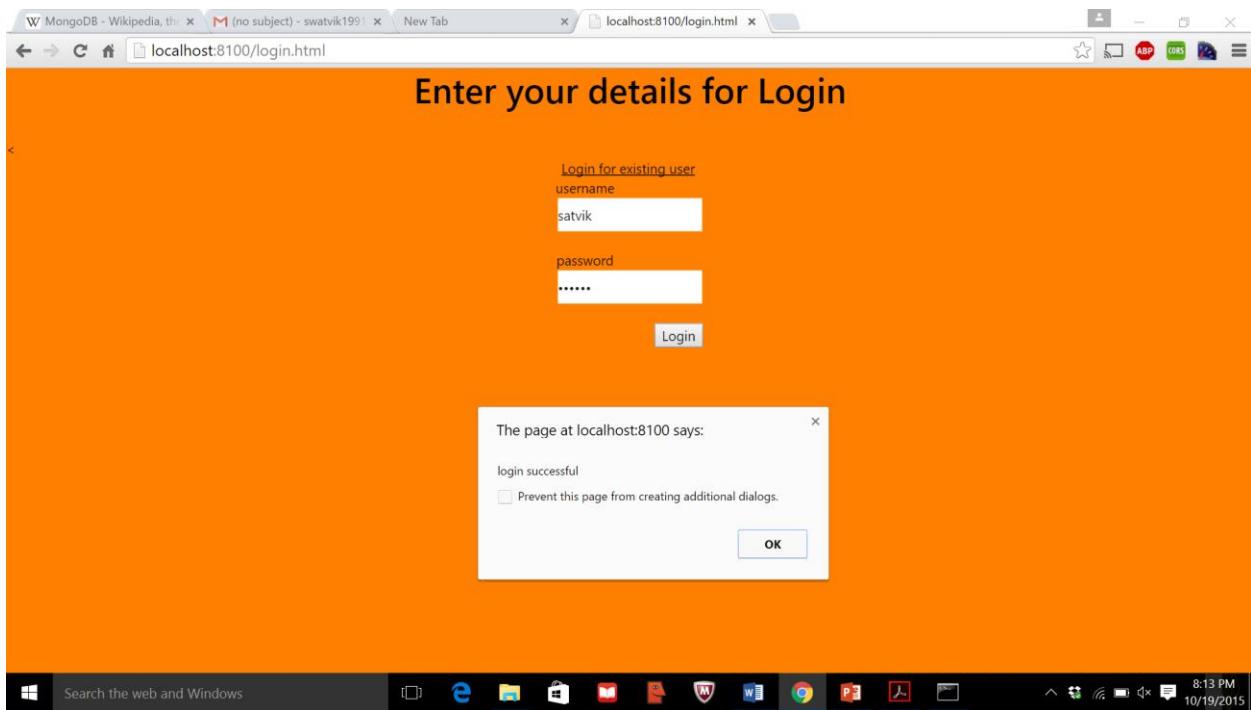
Testing Sign in button

Expected Output: On clicking Sign in button it should redirect the user to login Page.

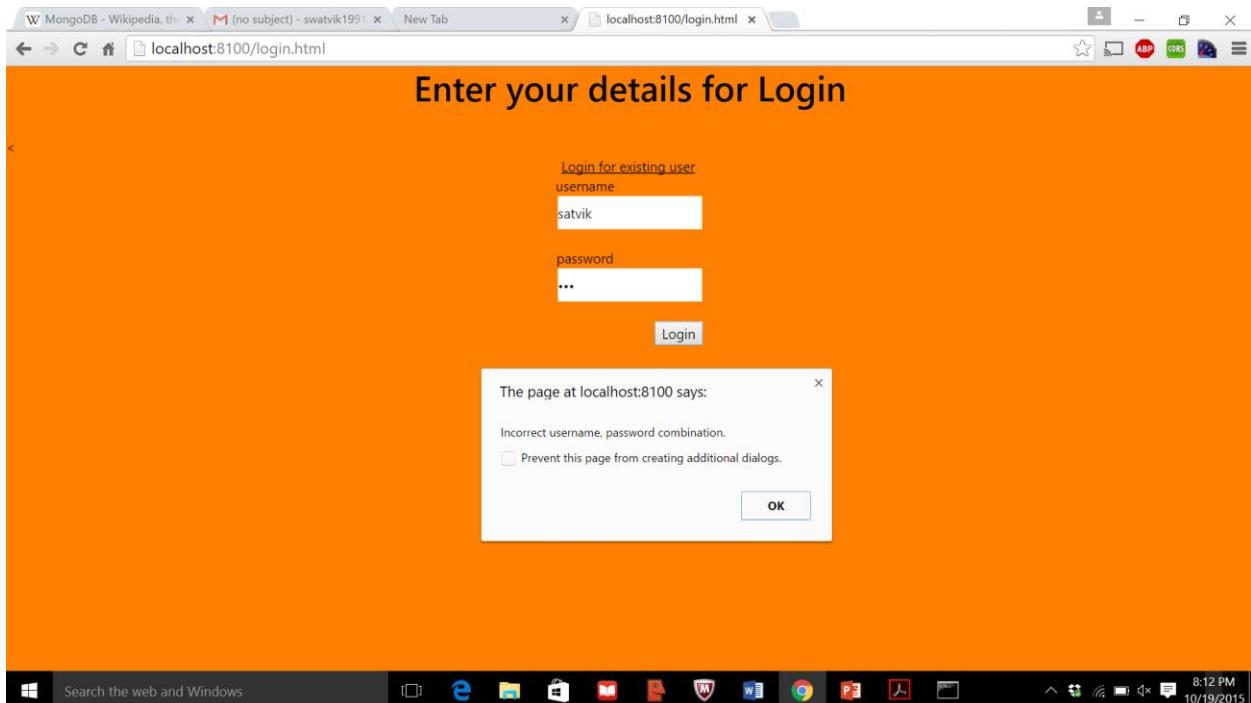
Actual output: On clicking Sign in button the user was redirected to the login Page.



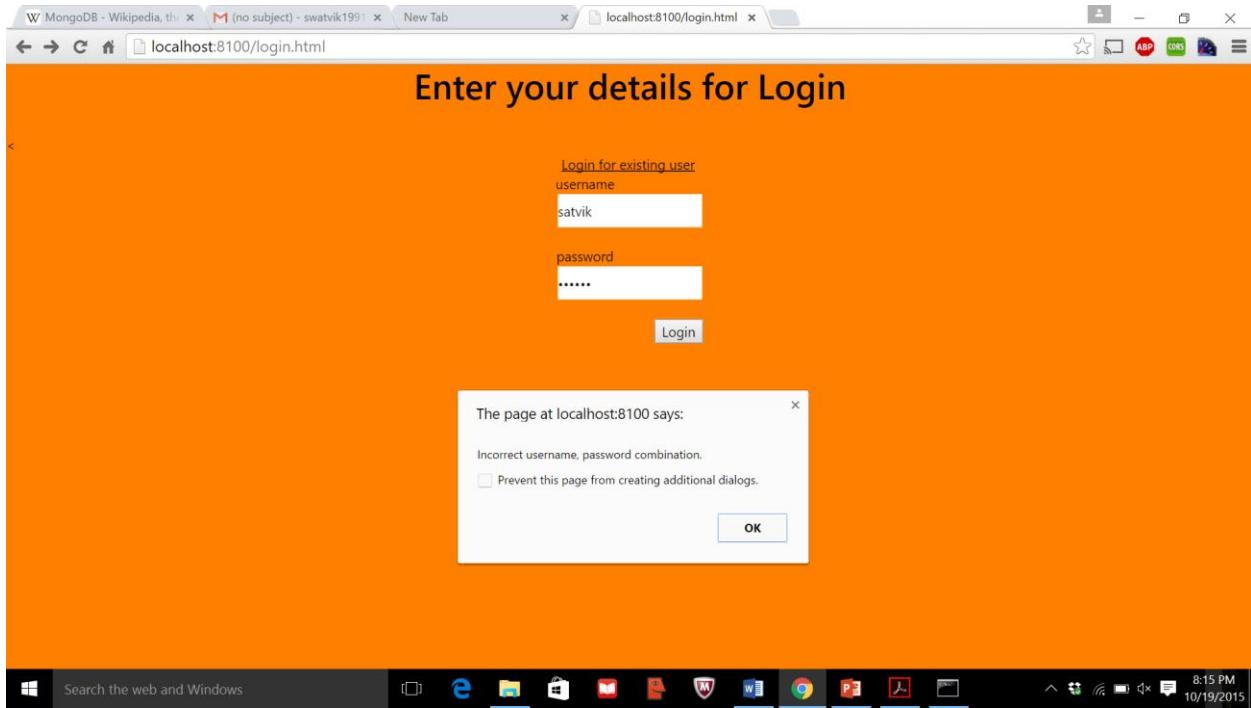
Testing Login page for correct username and password combination



In this login page we have given same username and password which was provide in registration page. It displays login successful for valid data.



This time we have given invalid password with different length of password then system responding with incorrect message



This time we have given password with same length as valid password for username then system displays with incorrect username and password combination message.

Welcome to Dashboard

Hello satvik

Select the state and city in which you want to search for stores and products

choose state: Missouri ▾ choose City: saint louis ▾ Display stores show location

Select the item to be searched in your desired store

Enter Item:

Enter StoreID:

Display items list

Storename	Address	City	State	Zip	StoreId
Global Foods Market	421 N Kirkwood Rd.	Saint Louis	MO	63122b8301f3e39	
Schnucks	5519 Oakville Shopping Ctr	Saint Louis	MO	63129e8e655ec2a	
Schnucks	4333 Butler Hill Rd.	Saint Louis	MO	63128c1aa97a512	
Schnucks	10650 Olive Blvd.	Saint Louis	MO	6314143f1982ad8	
Schnucks	3430 S Grand Blvd.	Saint Louis	MO	631188d6cd0c2b3	
Schnucks	1589 Sierra Vista Plz	Saint Louis	MO	6313849cc8eb44	
Schnucks	315 North 9th Street	Saint Louis	MO	6310112928c0291	
Schnucks	4171 Lindell Blvd.	Saint Louis	MO	63108cda176bc6f	
Schnucks City Plaza	3431 Union	Saint Louis	MO	631153c1ed94b4b	
Schnucks Kossmuth	4127 N Grand	Saint Louis	MO	631070397230453	
Schnucks	60 Hampton Village Plaza	Saint Louis	MO	6310964432e5786	
Schnucks	5055 Arsenal	Saint Louis	MO	63139d9ecbb6fa6	
Schnucks	7450 Hampton Ave.	Saint Louis	MO	63109b3c8709957	
Schnucks	7450 Hampton Ave.	Saint Louis	MO	63109b3c8709957	

After logging into the system we have the dashboard page of application where we select state and city

On clicking “display stores” button the system displays list of stores located in the given city.

W MongoDB - Wikipedia, thi x M (no subject) - swatvik199 x New Tab x localhost:8100/home.html x

Welcome to Dashboard

Hello satvik

Select the state and city in which you want to search for stores and products

choose state: Missouri choose City: saint louis Display stores show location

Select the item to be searched in your desired store

Enter Item:

milk

Enter StoreID:

e8e655ec2a

Itemname	ItemDescription	ItemCategory	ItemID	AisleNumber
Enfamil Enfacare Lipil Milk-Based Powder - 12.8 Oz Canister	Enfamil Enfacare Lipil Milk-Based Powder - 12.8 Oz Canister	Baby	75360	Aisle:17
Enfamil A.R. Lipil Milk-Based Powder w/Iron - 12.9 Oz Canister	Enfamil A.R. Lipil Milk-Based Powder w/Iron - 12.9 Oz Canister	Baby	75361	Aisle:17
Enfamil Lipil w/Iron Milk-Based Powder - 25.7 Oz Canister	Enfamil Lipil w/Iron Milk-Based Powder - 25.7 Oz Canister	Baby	75370	Aisle:17
Enfagrow Premium Next Step Lipil Milk Based Powder Unflavored - 24 Oz Canister	Enfagrow Premium Next Step Lipil Milk Based Powder Unflavored - 24 Oz Canister	Baby	75371	Aisle:17
Enfamil A.R. Milk-Based Powder W/Iron Enfamil A.R. - 24 Oz Canister	Enfamil A.R. Milk-Based Powder W/Iron Enfamil A.R. - 24 Oz Canister	Baby	78198	Aisle:17
Enfamil A.R. Infant Formula For Spit-Up Milk-Based Powder W/Iron Enfamil A.R. - 22.2 Oz Plastic Tub	Enfamil A.R. Infant Formula For Spit-Up Milk-Based Powder W/Iron Enfamil A.R. - 22.2 Oz Plastic Tub	Baby	78199	Aisle:17
Enfamil Lipil w/Iron Milk-Based Powder - 12.9 Oz Canister	Enfamil Lipil w/Iron Milk-Based Powder - 12.9 Oz Canister	Baby	78201	Aisle:17
Enfamil Premium 1 Infant Formula Milk-Based Powder W/Iron Enfamil Premium - 12.5 Oz Canister	Enfamil Premium 1 Infant Formula Milk-Based Powder W/Iron Enfamil Premium - 12.5 Oz Canister	Baby	78202	Aisle:17
Enfamil Premium Infant Formula Milk-Based Powder Enfamil Premium - 23.4 Oz Plastic Tub	Enfamil Premium Infant Formula Milk-Based Powder Enfamil Premium - 23.4 Oz Plastic Baby Tub	Baby	78203	Aisle:17
Enfamil Premium Lipil Milk-Based w/Iron Concentrated Liquid Enfamil Premium - 13 Fl Oz Can	Enfamil Premium Lipil Milk-Based w/Iron Concentrated Liquid Enfamil Premium - 13 Fl Baby Oz Can	Baby	78205	Aisle:17

Search the web and Windows 8:00 PM 10/19/2015

On typing item and stores id in text box from following list then clicking “display item list” it displayed item based on stores and item selected by user. Here we have selected milk and entered stored id in textbox.

W MongoDB - Wikipedia, thi x M (no subject) - swatvik199 x New Tab x localhost:8100/maplocator.html x

Show Directions and distance

Map Satellite

Kans A City

Overland Park

Lawrence

Olathe

Warrensburg

Sedalia

Columbia

Fulton

Hermann

Jefferson City

St Charles

Fergus

B

St Louis

Map data ©2015 Google Terms of Use Report a map error

Search the web and Windows 8:04 PM 10/19/2015

In this increment we only show only direction from current location to selected city further enhancement will be done in coming increment.

Performance Testing

The screenshot shows a web browser window with the URL `localhost:8100`. The page title is "Plan your Shopping". Below the title, a message says "start using right now". The browser's status bar indicates the URL is `http://localhost:8100/`. A YSlow extension is active, displaying a performance report. The grade is B, with an overall score of 85. The report lists several items under "Grade A" and "Grade F".

Grade B Overall performance score 85 Ruleset applied: YSlow(V2) URL: http://localhost:8100/

Grade A on Make fewer HTTP requests

This page has 4 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and Image maps.

[Read More](#)

YSlow extension interface:

- Home
- Grade
- Components
- Statistics

Rulesets: YSlow(V2)

Filter by: ALL (23) | [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

Social sharing: Tweet | Share

Windows taskbar at the bottom: Search the web and Windows, Start button, Task View, File Explorer, Edge, File Manager, Mail, Photos, Taskbar icons, System tray showing 8:47 PM, 10/19/2015.

The screenshot shows a web browser window with the URL `localhost:8100`. The page title is "Plan your Shopping". Below the title, a message says "start using right now". The browser's status bar indicates the URL is `http://localhost:8100/`. A YSlow extension is active, displaying a performance report. The grade is B, with an overall score of 85. The report lists several items under "Grade A" and "Grade F".

Grade B Overall performance score 85 Ruleset applied: YSlow(V2) URL: http://localhost:8100/

Grade A on Make fewer HTTP requests

This page has 4 external Javascript scripts. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and Image maps.

[Read More](#)

YSlow extension interface:

- Home
- Grade
- Components
- Statistics

Rulesets: YSlow(V2)

Filter by: ALL (23) | [CONTENT \(6\)](#) | [COOKIE \(2\)](#) | [CSS \(6\)](#) | [IMAGES \(2\)](#) | [JAVASCRIPT \(4\)](#) | [SERVER \(6\)](#)

Social sharing: Tweet | Share

Windows taskbar at the bottom: Search the web and Windows, Start button, Task View, File Explorer, Edge, File Manager, Mail, Photos, Taskbar icons, System tray showing 10:50 PM, 10/19/2015.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Plan your Shopping" and displays a message: "Plan your shopping is an application which allows the user to get aware of the different shopping stores and supermarkets in the specified location by him. This application also provides the information on about different items available with in the stores." Below this, there is a large orange banner with the text "start using right now".

The browser's address bar shows the URL "localhost:8100". The YSlow extension interface is overlaid on the page, showing a table of components and their details. The table has columns for TYPE, SIZE (KB), GZIP (KB), COOKIE RECEIVED (bytes), COOKIE SENT (bytes), HEADERS, URL, EXPIRES (Y/M/D), RESPONSE TIME (ms), ETAG, and ACTION.

TYPE	SIZE (KB)	GZIP (KB)	COOKIE RECEIVED (bytes)	COOKIE SENT (bytes)	HEADERS	URL	EXPIRES (Y/M/D)	RESPONSE TIME (ms)	ETAG	ACTION
doc (1)	2.7K									
js (4)	2108.2K									
css (2)	223.7K									
favicon (1)	0.02K									
font (1)	120.7K									

* type column indicates the component is loaded after window onload event
+ Another test would return this result for broken resources

The browser's taskbar at the bottom shows various pinned icons, and the system tray indicates the date and time as 10/19/2015 at 10:51 PM.

The above screenshots shows the results of performance testing on our website using YSlow.

From the results of the test we have come areas like usage of Content Delivery Network and Expiry Headers where the performance is low and needs improvement. We will fix these issues in the next increment.

Implementation

Server Side Implementation:

The server side implementation of this application mainly consists of processing of API used in this project.

The APIs used in this project are “Supermarketapi”, “GoogleMaps” API, “MongoDB” API.

The Login and Registration pages use MongoDB API. Registration page uses MongoDB API for storing the details of the user in Mongo Database, and Login Page uses MongoDB API for reading the details of the user.

“Supermarketapi” contains list of the stores along with details of their location. It contains the information regarding all the items in these stores. All these details are stored in the XML format in this API. This provides an easy way to perform search on stores and items by simple entry of location name and item name.

MapLocator page in this application uses the Google Maps API to locate the store selected by the user. It locates the address of the store in Google Maps and also it specifies the distance between the user and the store.

Link for Supermarketapi used in this project:

<http://supermarketapi.com/Default.aspx>

Link for MongoDB Data API used in this project:

https://api.mongolab.com/api/1/databases/vikas2/collections/users?apiKey=HxPILPvmcIj3SyZBMrPV38t7_BiBKUJ

Link for Google Maps API used in this project:

<https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=true>

Client Side Implementation:

Registration page implementation:

Registration page allows the user on his/her to enter his details and sign up for start using this Application. This page reads the details of the user and access the MongoDB API and stores the details of the user in the MongoDB database.

Login Page implementation:

Login Page allows the user on his/her device to login into the Application to start using this Application. This page reads username and password of the user and validates them. If the user has entered the correct combination of username and password then it redirects him to the Home page. Here, validation is performed by comparing the details entered by the user with details existing in the MongoDB database by accessing the MongoDB API.

Home Page implementation:

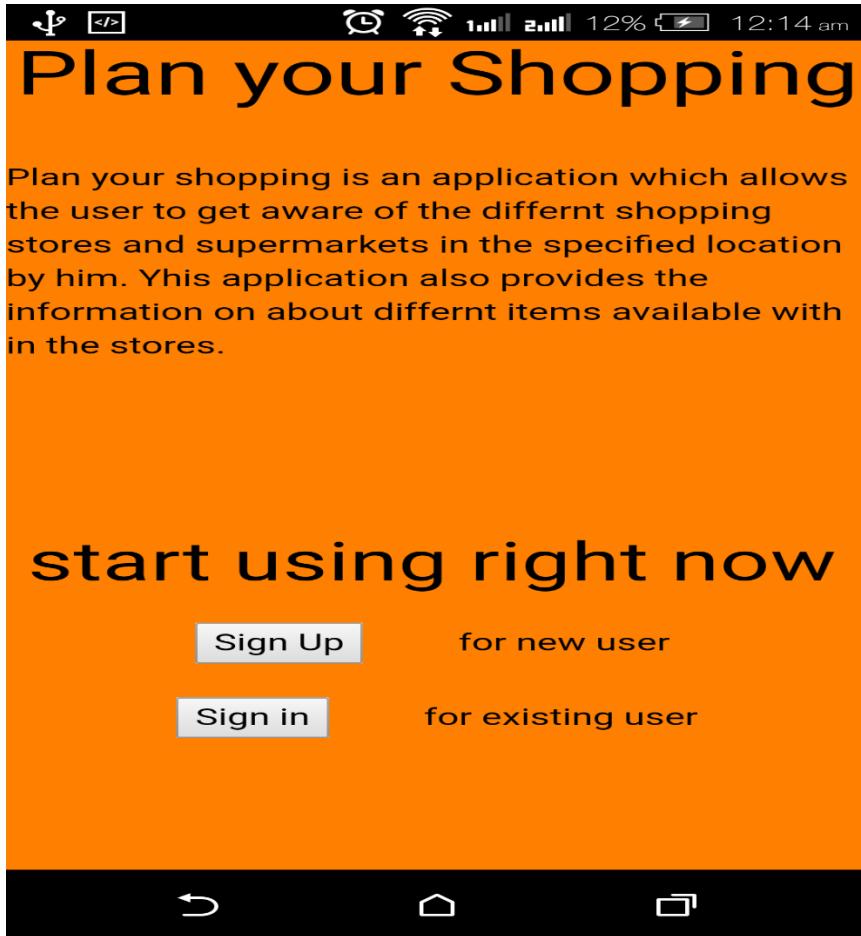
Home page allows the user to select the State and City in which he want to search for stores. For second increment the state and city option are limited to Missouri state and Saint Louis, Woodhavens cities respectively. This will be updated to all the states and cities in further increments. Once the user selects the required state and city, all the store existing in that city are displayed. Home Page also allows the user to search for a specific item in any of the store displayed in the list. All this is performed by accessing the “Supermarketapi” API and performing search on this API and fetching the required details from the server and displaying it on the client.

MapLocator Page implementation:

Maplocator Page reads the location selected by the user and traces direction from the user current location to the store location and displays the distance to be travelled. MapLocator page does this by accessing the Google Maps API and giving the user specified location as parameters to it.

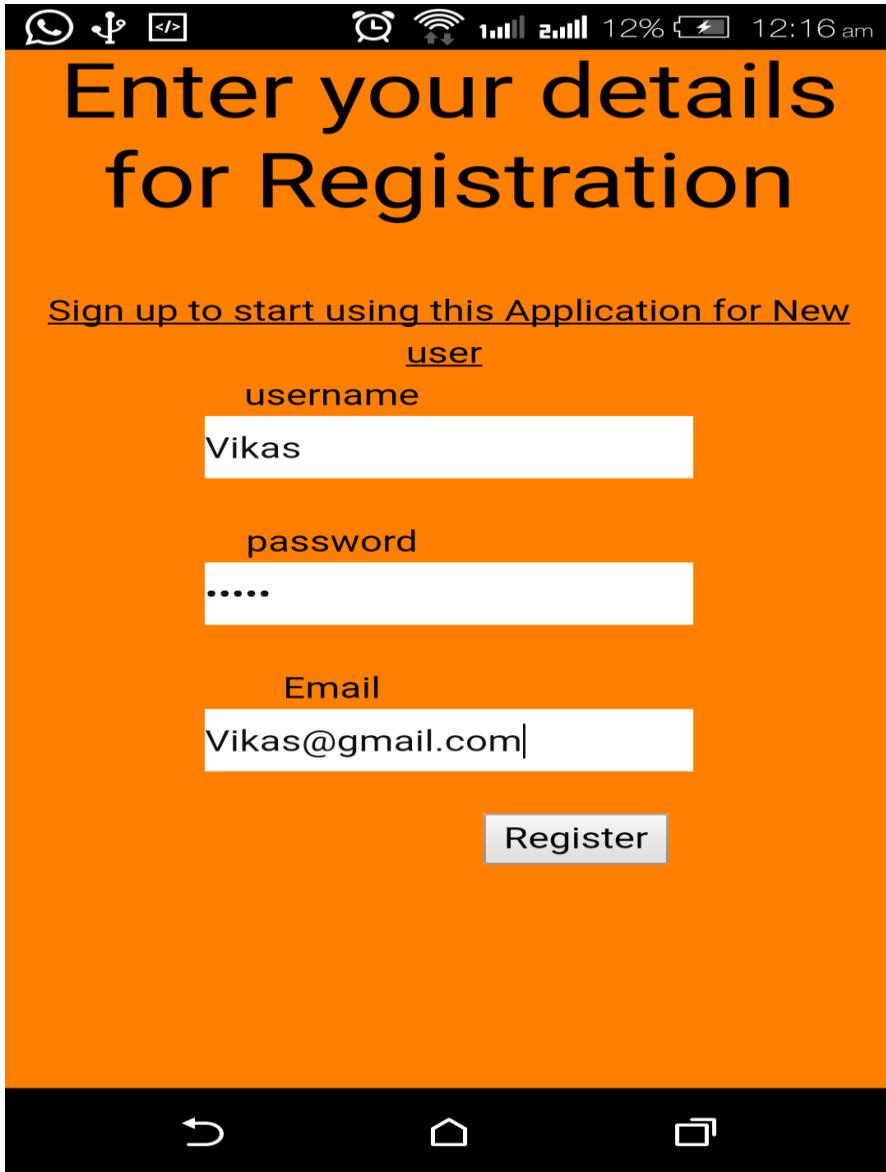
Screenshots of Implementation of Application:

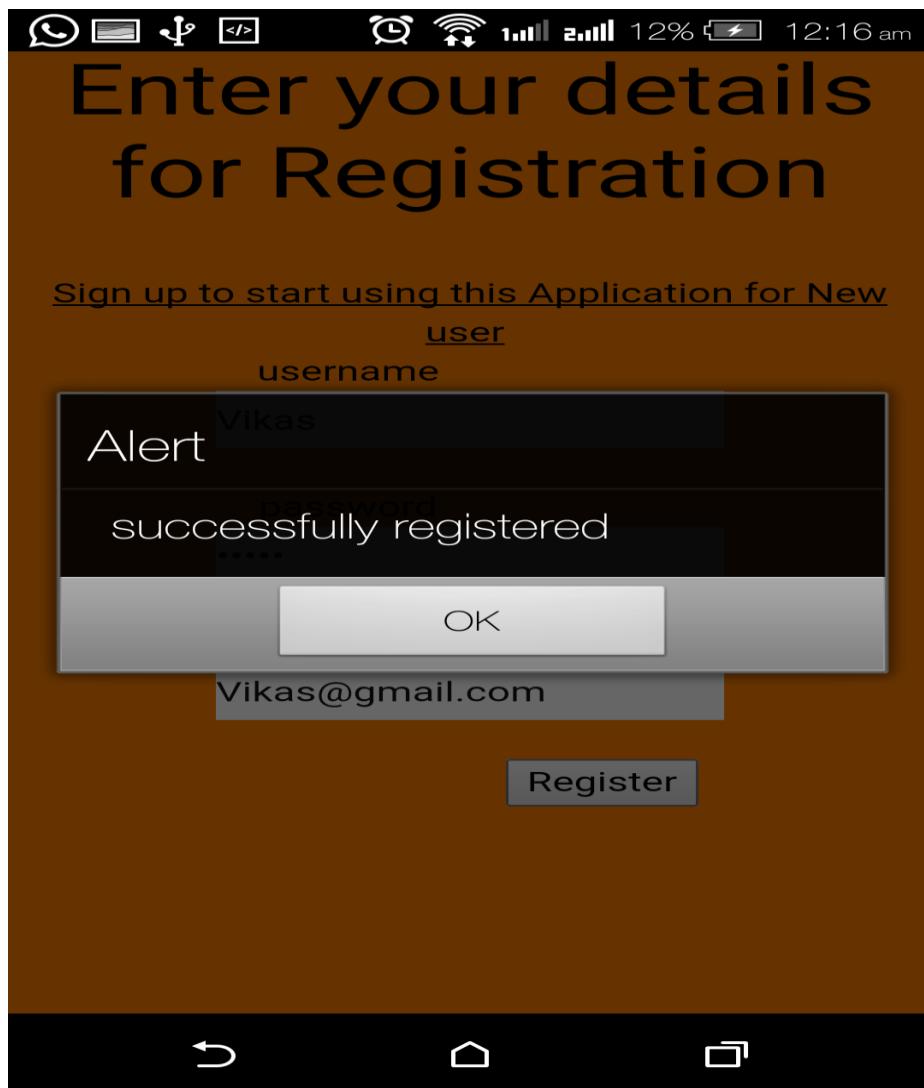
Index Page:



Index Page is the first page of the Application which the user sees. This Page displays the name of the Application and basic information about application. It also contains two button Sign Up and Sign In. On clicking Sign Up user will be navigated to Registration page and on clicking Sign In user will be navigated to Login Page.

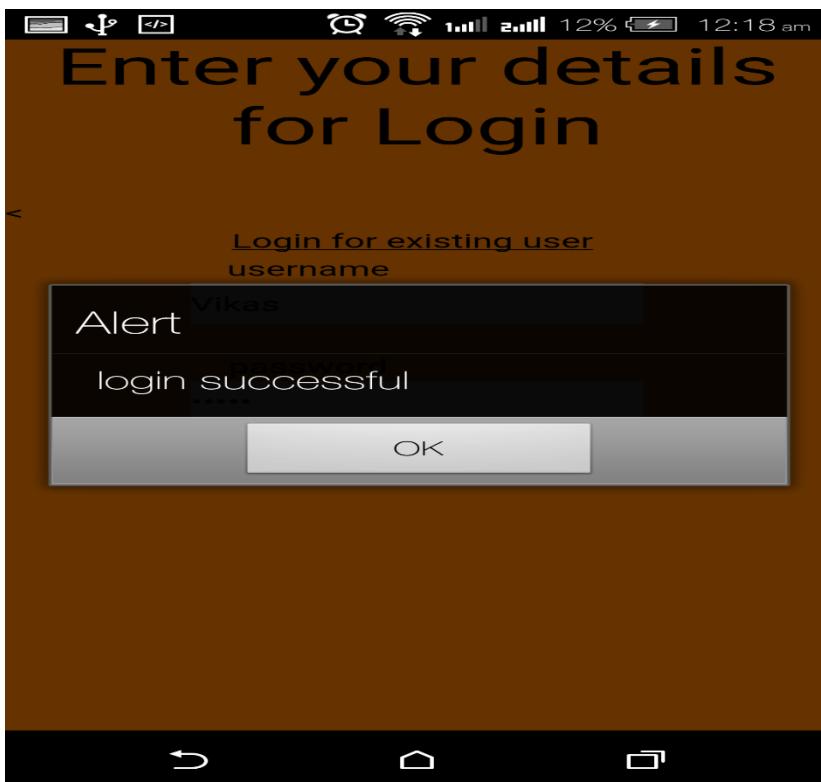
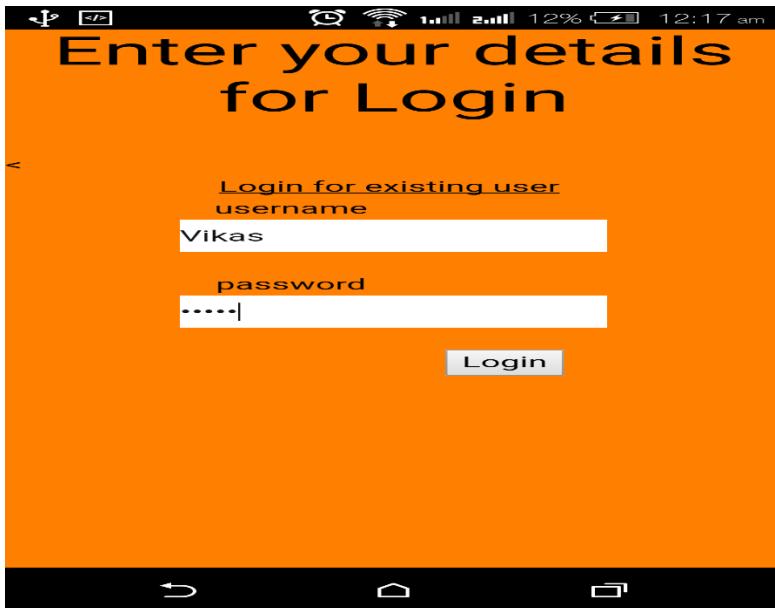
Registration Page:





Registration Page contains Text boxes for entering Username, Password and E mail-ID of the user. On clicking Register button the details will be stored into the Database and the user will be provided with acknowledgement of successful registration.

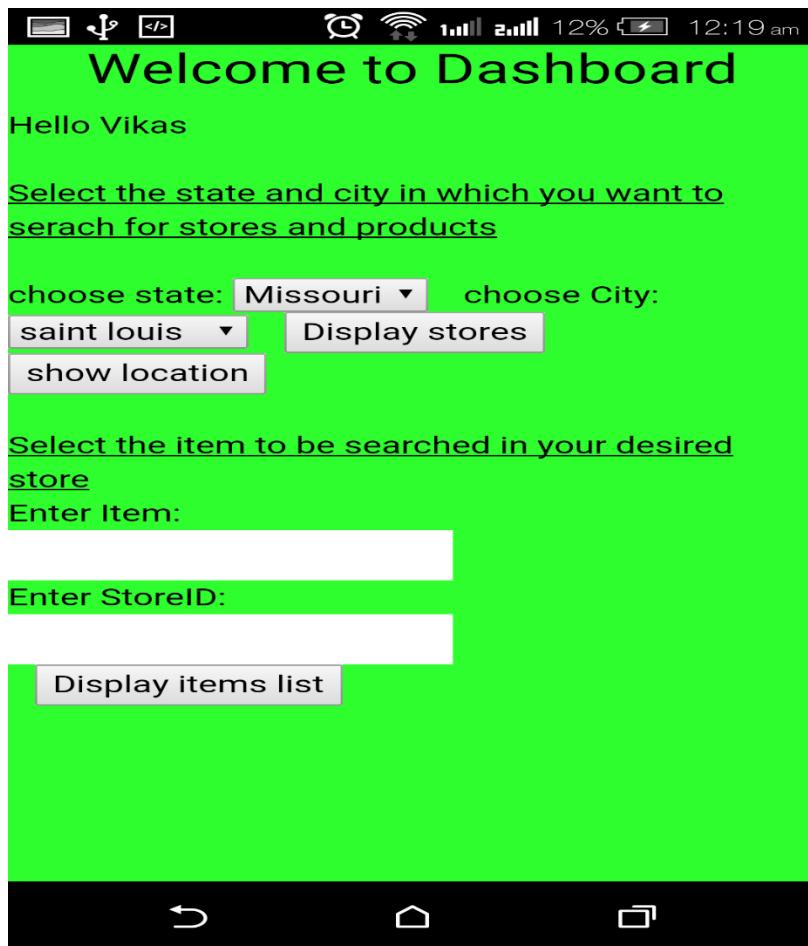
Login Page:



Login Page contains fields for entering Username and Password of the user. On clicking the Login button the user details are validated and if the details are found to be correct the an alert message that login has been successful is provided and he will be directed to home page.

Home Page:

Home Page of the user prompts the user to select state and city in which he want to search for the stores. On clicking Display Stores button the details of the stores in that city are displayed in a tabular format.





Hello Vikas

Select the state and city in which you want to
search for stores and products

choose state: Missouri ▼ choose City:

saint louis ▼

Display stores

show location

Select the item to be searched in your desired
store

Enter Item:

Enter StoreID:

Display items list

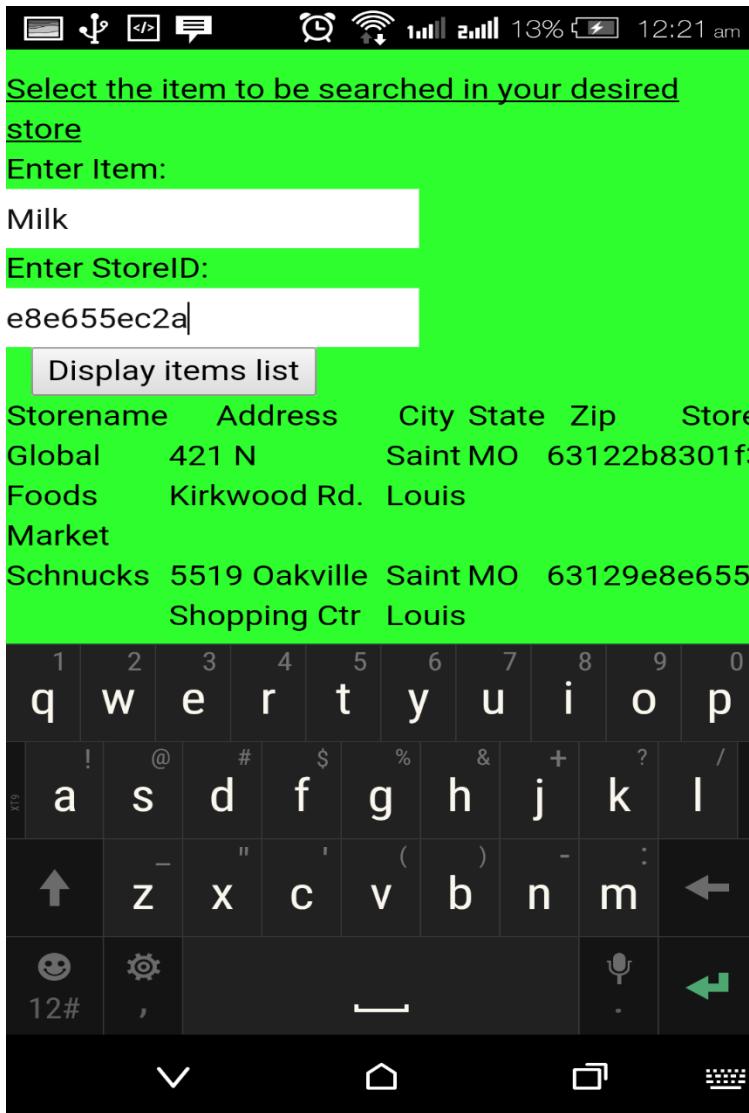
Storename	Address	City	State	Zip	StoreID
Global Foods Market	421 N Kirkwood Rd.	Saint Louis	MO	63122b8301f3e8e655e1aa97a	b8301f3e8e655e1aa97a
Schnucks	5519 Oakville Shopping Ctr	Saint Louis	MO	63129e8e655e1aa97a	e8e655e1aa97a
Schnucks	4333 Butler	Saint Louis	MO	63128c1aa97a	c1aa97a



The image shows a smartphone screen displaying a list of store locations. The top status bar shows various icons including signal strength, battery level at 13%, and the time 12:20 am. Below the status bar, a green header bar says "Enter StoreID:". The main content area has a title "Display items list" and a table with columns: Storename, Address, City, State, Zip, and Store. The data is as follows:

Storename	Address	City	State	Zip	Store
Global Foods Market	421 N Kirkwood Rd.	Saint Louis	MO	63122b8301f3	
Schnucks	5519 Oakville Shopping Ctr	Saint Louis	MO	63129e8e655e	
Schnucks	4333 Butler Hill Rd.	Saint Louis	MO	63128c1aa97a	
Schnucks	10650 Olive Blvd.	Saint Louis	MO	6314143f1982	
Schnucks	3430 S Grand Blvd.	Saint Louis	MO	631188d6cd0e	
Schnucks	1589 Sierra Vista Plz	Saint Louis	MO	631384f9cce8	
Schnucks	315 North 9th Street	Saint Louis	MO	6310112928c0	
Schnucks	4171 Lindell Blvd.	Saint Louis	MO	63108cda1761	
Schnucks	3431 Union City Plaza	Saint Louis	MO	631153c1ed94	
Schnucks	4127 N Grand	Saint Louis	MO	631070397230	

The Home Page also provides the user to search for a specific product with in a required store with in the list displayed. The user need to a keyword pertaining to the product and the storeID to search for that product in the specified Store.



Select the item to be searched in your desired store

Enter Item:

Milk

Enter StoreID:

e8e655ec2a

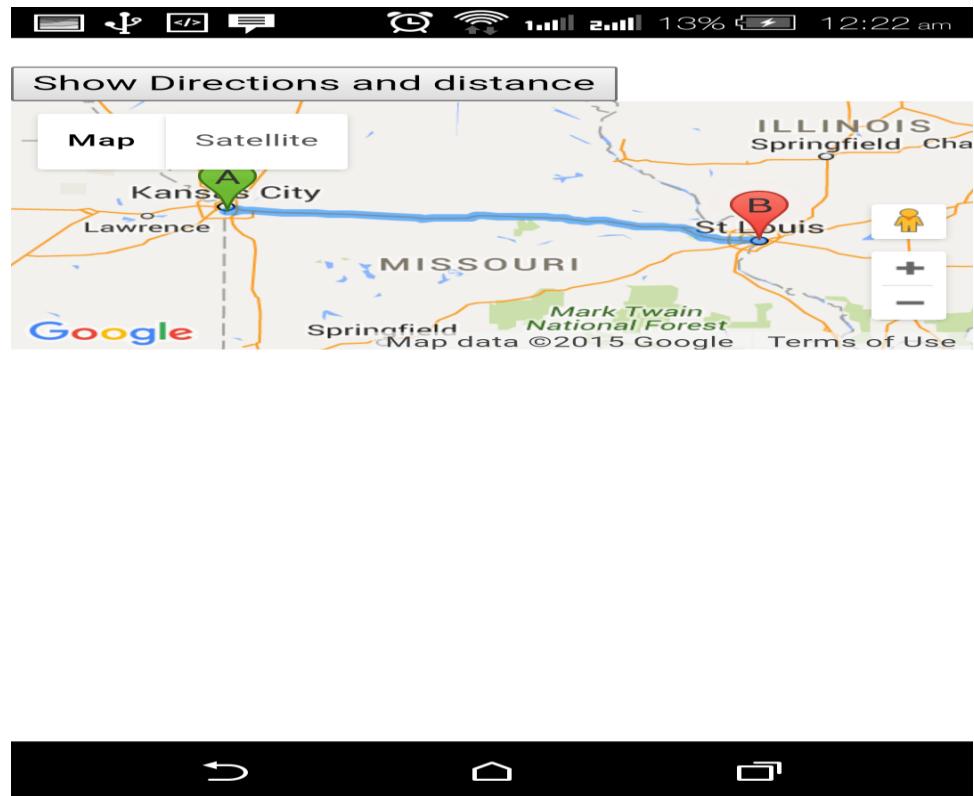
Display items list

Itemname	ItemDescription	ItemCategory	ItemID
Enfamil	Enfamil	Baby	75360 A
Enfacare	Enfacare Lipil		
Lipil Milk-Based	Milk-Based Powder - 12.8 Oz		
Powder - 12.8 Oz	Oz Canister		
Enfamil A.R.	Enfamil A.R.	Baby	75361 A
Lipil Milk-Based	Lipil Milk-Based Powder w/Iron -		
Powder - 12.9 Oz	12.9 Oz Canister		
Enfamil Lipil w/Iron Milk-	Enfamil Lipil w/Iron Milk-	Baby	75370 A

◀ ◁ ◁

Map Locator Page:

Map Locator Page Shows the directions from the user location to the city specified by the user for store location.



Project Management

This Project has been managed using the Kanban tool. This project has been planned to be completed in four increments. The first increment has already been completed.

For the second increment, the tasks to be done were already planned and they were at the Waiting stage at the time of first increment. These tasks are eventually completed and moved to the Done phase.

The tasks designed for the second phase are

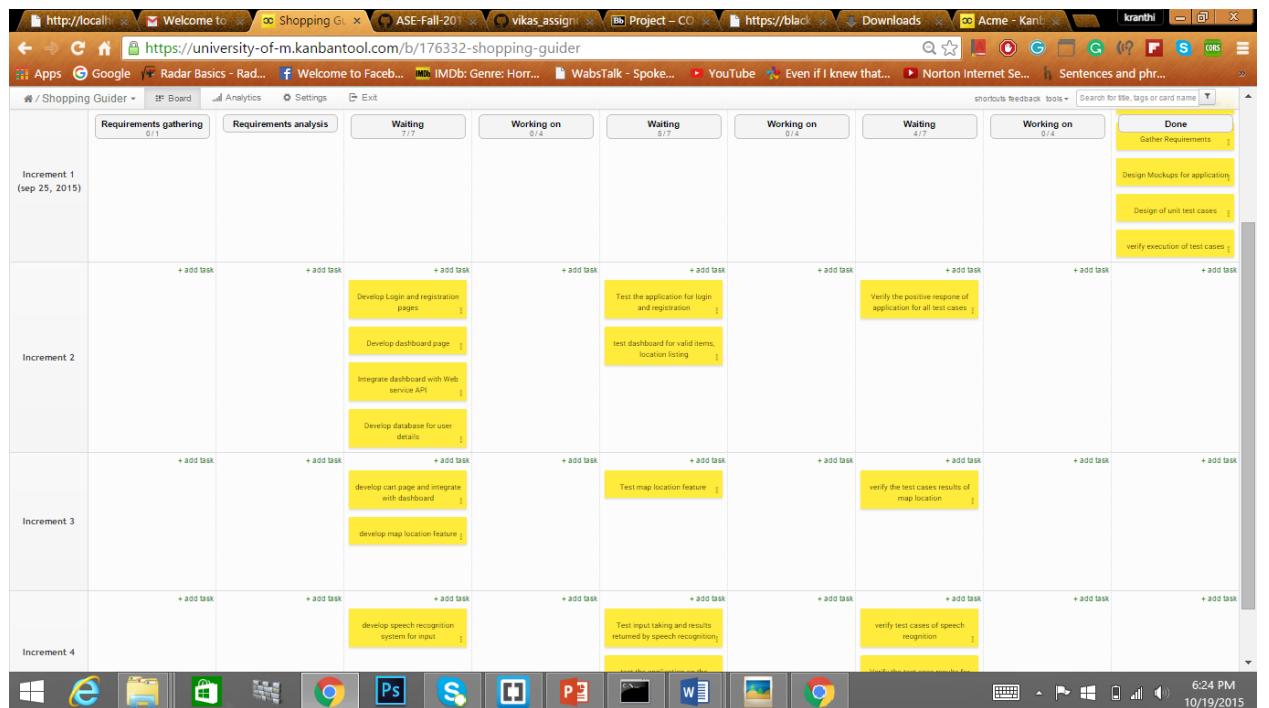
- **Development:**

1. Develop Login and Registration pages.
2. Develop Database for storing user details.
3. Develop Dashboard (Home) page.
4. Integrate the Dashboard page with SuperMarket API.

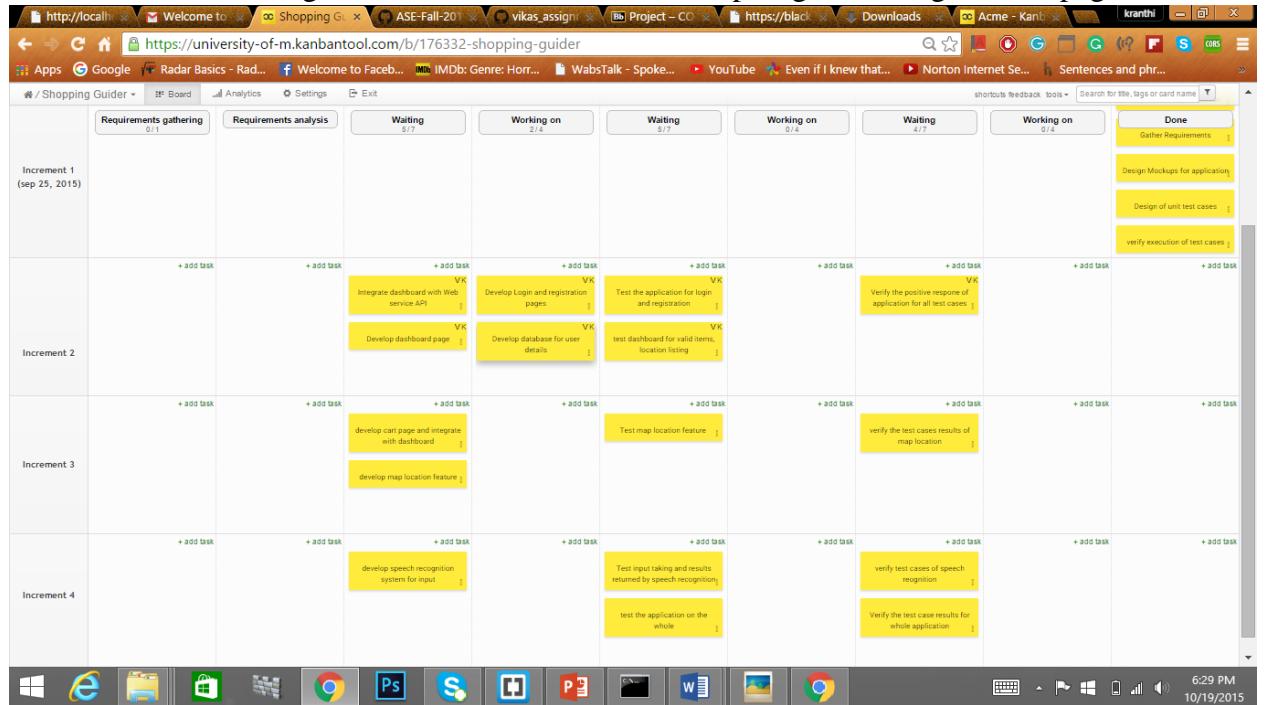
- **Testing:**

1. Test the functionality of Login Pages.
2. Test the functionality of Registration Pages.
3. Test the Dashboard page for displaying of list of stores for the selected city and state.
4. Test the Dashboard page for displaying of list of items available for the specified keyword for item in the specified store.

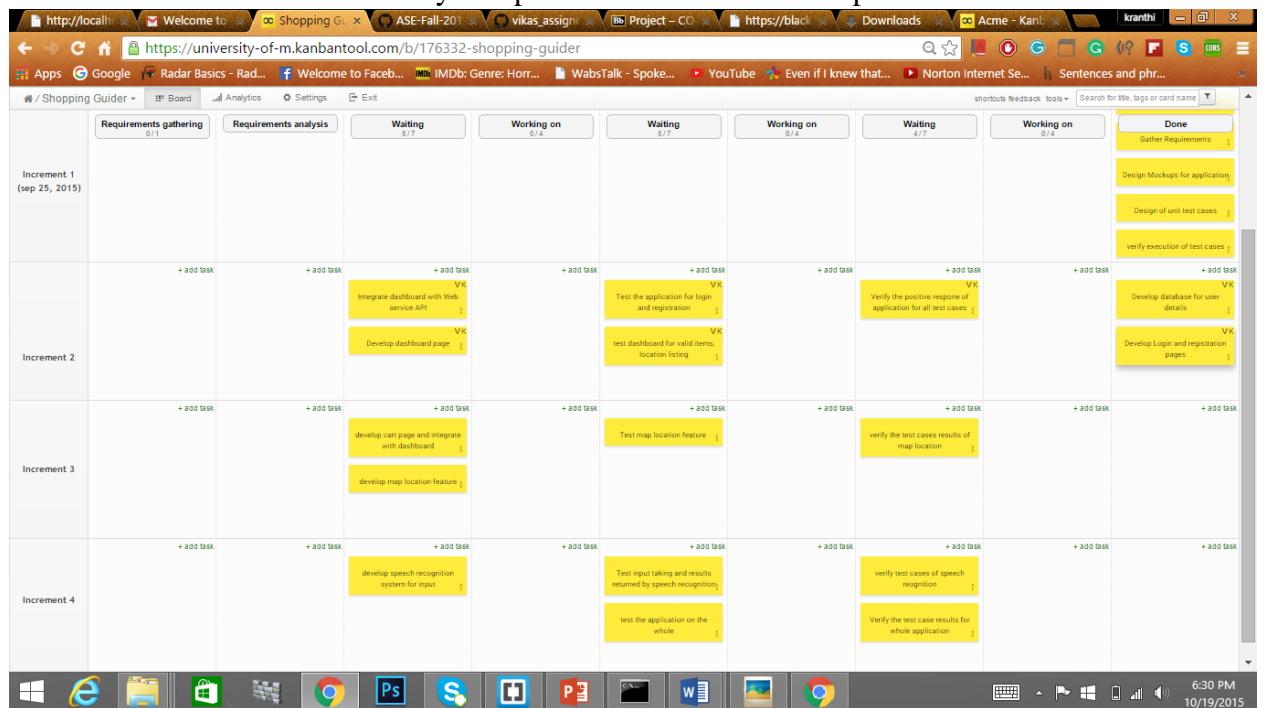
Initially all these tasks were in the waiting phase at the beginning of second increment.



The first task which is carried out was the developing of MongoDB database and getting a DataAPI for accessing it. Then next task was to develop Login and Registration pages



Both these tasks were successfully completed and moved to done phase.



The next tasks were Developing the Dashboard page and integrating the dashboard page with Supermarket API. Both these tasks were worked on simultaneously.

The screenshot shows a Kanban board with columns: Requirements gathering, Requirements analysis, Waiting, Working on, Waiting, Working on, Waiting, Working on, and Done. The board is divided into four horizontal sections labeled Increment 1 through Increment 4. Each section has a header row with column names and a body row where tasks are listed. Tasks are represented by yellow cards with descriptions and status indicators (e.g., 'VR' for verified). The 'Done' column on the right lists completed tasks: 'Gather Requirements', 'Design Mockups for application', 'Design of unit test cases', and 'verify execution of test cases'. The bottom of the screen shows a Windows taskbar with various icons and the system tray indicating the date and time as 10/19/2015 at 6:33 PM.

Both these tasks were successfully completed and moved to done phase.

This screenshot shows the same Kanban board after the tasks have been completed. The 'Done' column now includes additional items: 'Develop database for user details', 'Develop Login and registration pages', 'Develop dashboard page', and 'Integrate dashboard with Web service SuperMarketAPI'. The rest of the board structure remains the same, with increments and tasks listed in their respective columns.

The next tasks were to test the login and Registration pages, test the stores, items list displaying functionality of Dashboard (home page).

The screenshot shows a Kanban board with columns: Requirements gathering, Requirements analysis, Waiting, Working on, Waiting, Working on, Waiting, Working on, and Done. The board is divided into four increments:

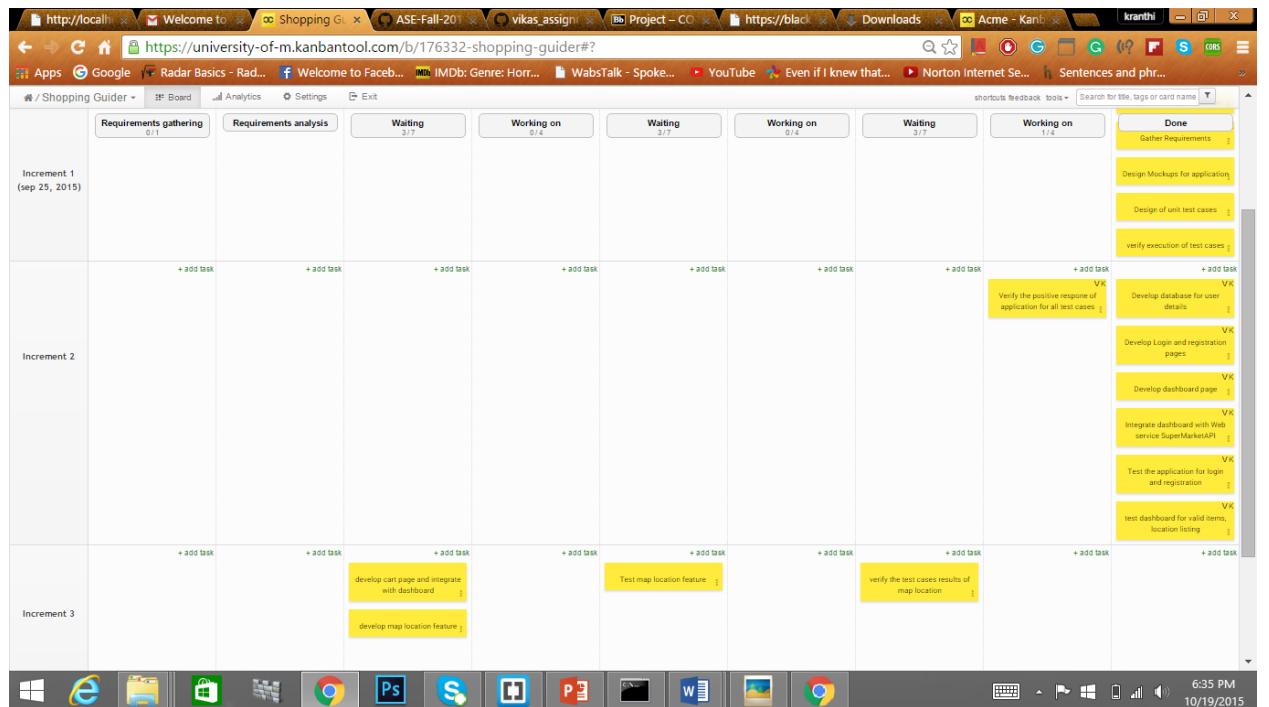
- Increment 1 (sep 25, 2015):** Contains one card in the Requirements gathering column.
- Increment 2:** Contains cards in the Requirements analysis, Waiting, Working on, Waiting, and Done columns. Tasks include "Test the application for login and registration" and "Verify the positive response of application for all test cases".
- Increment 3:** Contains cards in the Requirements analysis, Waiting, Working on, Waiting, and Done columns. Tasks include "develop car page and integrate with dashboard" and "develop map location feature".
- Increment 4:** Contains cards in the Requirements analysis, Waiting, Working on, Waiting, and Done columns. Tasks include "develop speech recognition system for input" and "Test input taking and results returned by speech recognition".

The Done column contains completed tasks: "Gather Requirements", "Design Mockups for application", "Design of unit test cases", and "verify execution of test cases". The Windows taskbar at the bottom shows various open applications like Google Chrome, Microsoft Word, and Project.

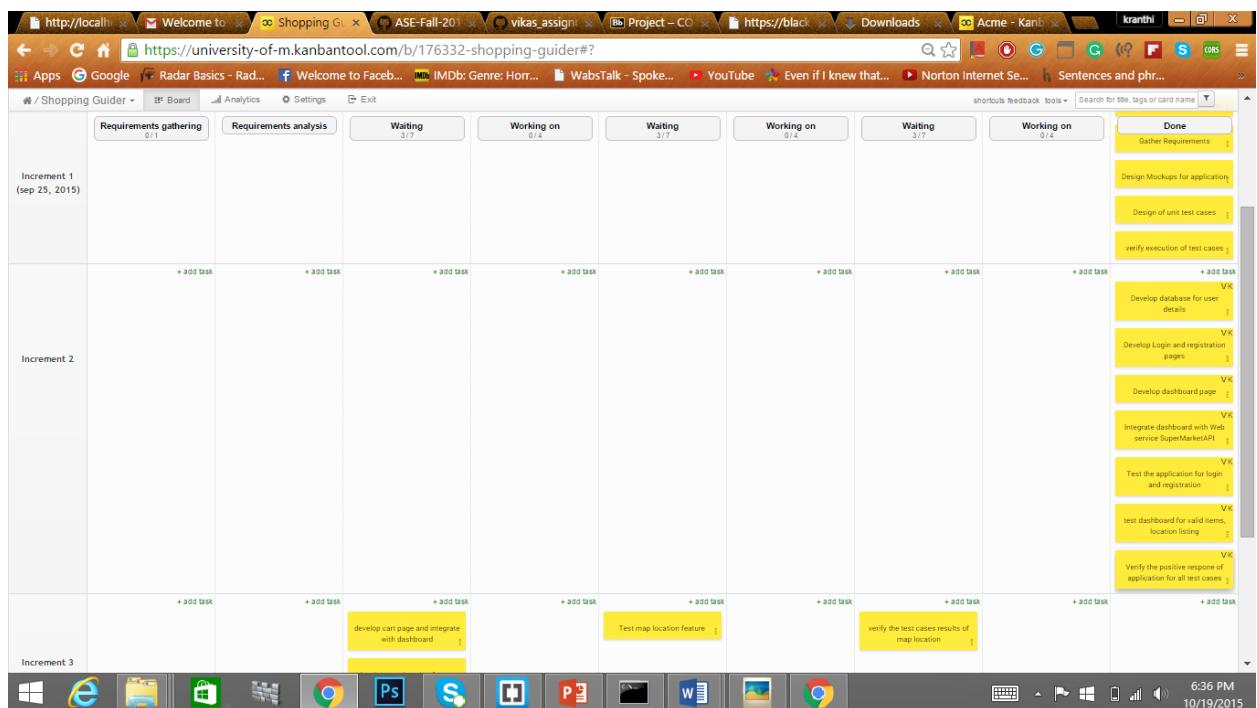
Both the testing tasks were successfully completed and moved to the done phase.

The screenshot shows the same Kanban board after the testing tasks have been moved to the Done column. The board structure remains the same, but the tasks in the Done column now include additional items: "Develop database for user details", "Develop Login and registration pages", "Develop dashboard page", "Integrate dashboard with Web-service SuperMarketAPI", "Test the application for login and registration", and "test dashboard for valid items, location listing". The Windows taskbar at the bottom shows the same set of open applications as in the previous screenshot.

The next task was verifying the test case outputs with the expected outputs in the test case design.



The above task was successfully completed and moved to done phase.



Work completed:

Description:

- Developed Login and Registration pages and their functionalities.
- Developed Database for storing user details in MondoDB and integrated it with Login and Registration Pages.
- Develop Dashboard (Home page) that displays list of stores and items based on user input.
- Integrated the supermarket API with the Dashboard page.
- Integrated the Google Maps API with Dashboard Page.
- Tested the Login and Registration pages for correct functioning.

Responsibilities:

- **Vikas kondapalli :**
Developing login page by connecting to MongoDB, Developing Dashboard page, Integration and accessing Supermarket API, Google Maps API, testing, documentation.
- **Gopi Krishna Bodapati:**
Developing registration page, developing MongoDB database and access it using data API , testing, documentation.
- **Swatvik Gunamaneni:**
Developing map locator page using Google Maps API, testing, documentation.

Time taken:

- Developing login page, Registration page: 90 hours
- Developing Dashboard page and integrating it with web service APIs: 110 hours
- Testing : 20 hours
- Documentation: 20 hours

Contributions:

- Vikas kondapalli : 33.33%
- Gopi Krishna Bodapati: 33.33%
- Swatvik Gunamaneni: 33.33%

Work to be completed:**Description:**

- Developing a cart page and integrating it with Dashboard page.
- Improvising the Map Location page with desired store location to be more specific on Map.
- Improvising the Dashboard page for letting the user perform search on stores in all the States in the USA as the current page only deals with Missouri state.
- Testing the cart page.
- Testing the Map location page.
- Developing a speech recognition feature for letting the user enter his input through his voice.

Responsibilities:

- Vikas kondapalli :
Improvising the Dashboard page for letting the user perform search on stores in all the States in the USA, testing, documentation.
- Gopi Krishna Bodapati:
Developing a cart page and integrating it with Dashboard page, testing, documentation.

- Swatvik Gunamaneni:
Improvising the Map Location page with desired store location to be more specific on Map, testing, documentation.

Time to be taken:

- Improvising Dashboard page: 60 hours
- Developing cart page: 110 hours
- Improvising the Map Location page: 60 hours
- Testing: 20 hours
- Documentation: 20 hours

Link for Kanban tool (project management tool):

<https://university-of-m.kanbantool.com/b/176332-shopping-guiders#?>

Username and password are required for accessing it.

Bibliography

Link for Supermarketapi used in this project:

<http://supermarketapi.com/Default.aspx>