

CORONA KIT MANAGER

IIHT

Time To Complete: 3 hrs

1. BUSINESS-REQUIREMENT:

1.1 PROBLEM STATEMENT:

The purpose of this application is to allow the visitors to create and order a customized Corona Prevention Kit.

1.1.1 Visitor Module:

The **Visitor Module** allows you to:

1. Access Portal's home page
2. Will be required to provide details before creating the kit.
3. Allow to enlist multiple items available to be added to kit (eg: Face mask, Sanitizers, immunity boosters medicines etc)
4. Allow to create the kit and add selected items in kit
5. Visitors shall be allowed to define the quantity of each item in the kit
6. Calculation of total amount for kit will be done automatically
7. Visitor shall be able to place order of the kit by providing his/her details with address
8. Visitor shall be able to view the order summary

1.1.2 Admin Module:

The **AdminModule of application** allows you to:

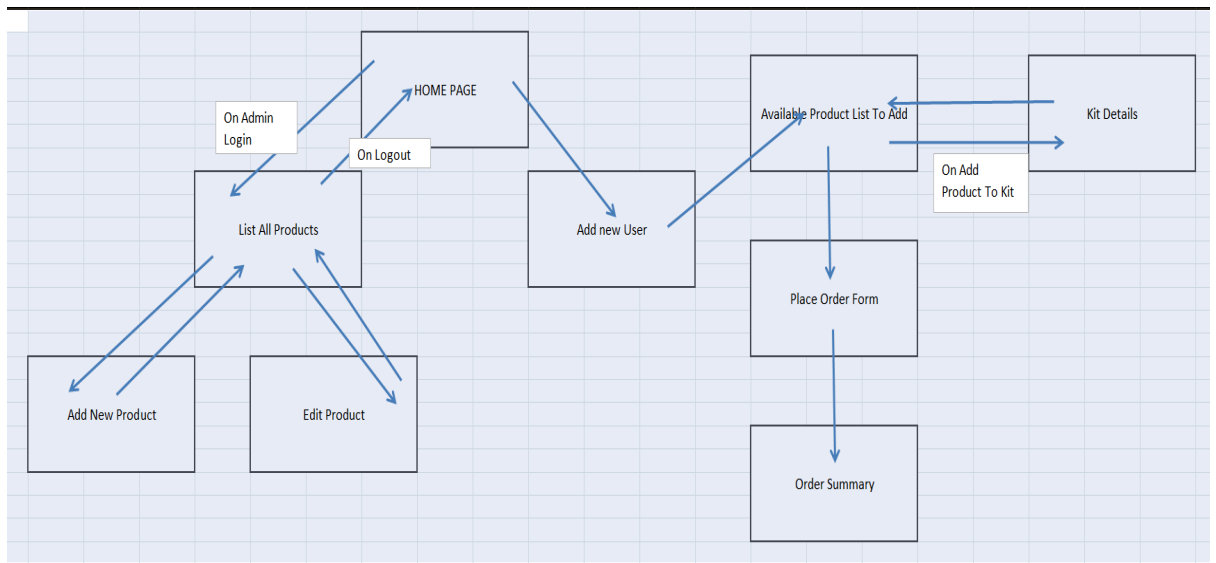
1. Login with predefined/hardcoded credentials
2. Admin will be able to add new items (with description, cost , name etc) to be shown to visitors, which they will use to create the prevention kit.
3. Admin will be able to edit and delete product as well

2. RESOURCES AVAILABLE :

| Description | View Pages Name | Remarks |
|----------------------------------------------------------|------------------------|-------------------------------------------------------------------------------|
| Common UI | | |
| Home Page | index.jsp | Contains Login form for Admin and link to new user page |
| Error Page | errorPage.jsp | |
| Header | header.jsp | |
| Footer | footer.jsp | |
| Not Found Page | notfound.jsp | If url contains invalid 'action' parameter than this page should be reflected |
| | | |
| Admin Services UI | | |
| Shall contain Entry Form for new product entry | newproduct.jsp | Must use input field names as pname, pcost, pdesc |
| Shall contain Entry Form for editing product detail | editproduct.jsp | |
| Shall use to show all products | listproducts.jsp | |
| | | |
| | | |
| User Services UI | | |
| Shall contain Entry Form for new user entry | newuser.jsp | Must use input field names as pname, pemail, pcontact |
| Shall show list of products available to add to kit | showproducts toadd.jsp | |
| Shall show list of products added to kit | showkit.jsp | |
| Shall contain Entry Form to input address to place order | placeorder.jsp | Must use input field name address |
| Shall show the final Order Summary | ordersummary.jsp | |

| | |
|----------------|-------|
| Admin Login Id | admin |
| Admin Password | admin |

3 PAGE FLOW :



4 SQL FOR APPLICATION :

```
Create database coronakit;  
use coronakit;
```

```
CREATE TABLE ProductMaster( Id INT(11) AUTO_INCREMENT,  
  ProductName VARCHAR(32),  
  Cost VARCHAR(32),  
  ProductDescription VARCHAR(100),  
  CONSTRAINT key1 PRIMARY KEY (Id)  
);
```

```
CREATE TABLE Kit( Id INT(11) AUTO_INCREMENT,  
  PersonName VARCHAR(32),  
  Email VARCHAR(32),  
  ContactNumber VARCHAR(32),  
  TotalAmount INT(11),  
  DeliveryAddress VARCHAR(100),  
  OrderFinalized BIT,  
  OrderDate VARCHAR(32),  
  CONSTRAINT key1 PRIMARY KEY (Id)  
);
```

```
CREATE TABLE KitDetail( Id INT(11) AUTO_INCREMENT,  
  CoronoKitId INT(11),  
  ProductId INT(11),  
  Quantity INT(11),  
  Amount INT(11),  
  CONSTRAINT key1 PRIMARY KEY (Id)  
);
```

5 EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.
3. To build your project use command:
`mvn clean package -Dmaven.test.skip`
4. To launch your application, run the following command to run the application:
`java -jar target/dependency/webapp-runner.jar
target/coronokit.war --port 8081`
5. This editor Auto Saves the code.
6. If you want to exit(login) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B**-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
7. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
8. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
9. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

Note: The application will not run in the local browser
10. Default credentials for MySQL:
 - a. Username: **root**
 - b. Password: **pass@word1**

11. To login to mysql instance: Open new terminal and use following command:

- a. `sudo systemctl enable mysql`
- b. `sudo systemctl start mysql`

NOTE: After typing the second sql command (`sudo systemctl start mysql`), you may encounter a warning message like :

System has not been booted with systemd as init system (PID 1).

Can't operate. Failed to connect to bus: Host is down

>> Please note that this warning is expected and can be disregarded.
Proceed to the next step.

- c. `mysql -u root -p`

The last command will ask for password which is '`pass@word1`'

12. Mandatory: Before final submission run the following command:

`mvn test`

13. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.