All-in-one E-Commerce Website

CSE3002 – INTERNET & WEB PROGRAMMING

J COMPONENT PROJECT DOCUMENT

FALL 2018 - 2019

GROUP/TEAM PROJECT

16BCE0779	QUEENIE DAS
16BCE0979	PRIYANSH JAIN

UNDER THE GUIDANCE OF
PROF. NAVEEN KUMAR N

SCOPE

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ■ CHENNAI
www.vit.ac.in

School of Computer Science and Engineering

DECLARATION

We hereby declare that the J Component project entitled **“All-in-one E-Commerce Website”** submitted by us to the School of Computer Science and Engineering, VIT University, Vellore in partial fulfillment of the requirements for the **Internet and Web Programming (CSE3002)** course, is a record of bonafide work carried out by us under the supervision of **Naveen Kumar N, Assistant Professor (Sr).** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Reg.No: 16BCE0779
Name : Queenie Das

Reg.No: 16BCE0979
Name : Priyansh Jain

TABLE OF CONTENTS

List of Tables:

List of Figures:

List of Abbreviations:

HTML	:	Hyper Text Mark-up Language
CSS	:	Cascading Style Sheets
JS	:	JavaScript
UI	:	User Interface
DOM	:	Document Object Model
PHP	:	Hypertext Preprocessor
AJAX	:	Asynchronous JavaScript and XML
CDN	:	Content Delivery Network
JSON	:	JavaScript Object Notation
NPM	:	Node Package Manager
EC	:	E-commerce

ABSTRACT

The Web is one of the most revolutionary technologies that has changed the business environment and has had a dramatic impact on the future of electronic commerce (EC). The immense popularity of the Internet in recent years has been fueled largely by the prospect of performing business online. Electronic commerce is officially defined as buying and selling of product, services or information via computer networks, mainly the Internet.

In this project, we will develop an E-commerce website that implements all the functionalities of such a website. We will develop this using HTML, CSS, JavaScript, PHP and some other recent technologies. Also, the E-commerce website will provide solutions to some problems that the existing E-commerce websites have.

INTRODUCTION

1.1.Aim

The rapid adoption of E-Commerce as a commercial medium has caused firms to experiment with innovative ways of doing business. A large number of E-commerce platforms have emerged in the recent years and yet, there are several problems that are faced by the customers, sellers and developers of the platforms. The aim of this project is to develop an all-in-one E-Commerce platform that makes use of recent technological advancements to provide solutions to the problems that exist for the same, and which will be described below.

1.2. Objective

The Objectives of this project are

- To create a responsive website that looks equally appealing on different screen sizes.
- To provide a well-designed User Experience to all potential customers and this includes the freedom to view and choose.
- To optimize the load-times of images and the base webpages, control the amount of data transferred and to not over bloat the website UI.
- To provide a reliable, persistent database service for handling the transactions performed on the website.
- To deploy the website in a zero-downtime manner using recent technological advancements.

1.3. System Requirements

- Web Browser - Chrome/Firefox/Microsoft Edge
- Computer - Ubuntu Linux(16.04/18.04), Windows
- Docker Community Edition
    - Docker Engine
    - Docker Swarm
    - Docker Compose

SOLUTION FOR EXISTING PROBLEMS

1. Image size compression wrt devices for faster loading

1.1. Problem description and its Solution

Problem Description:

An E-commerce website is filled with images of products. The fast loading of these images is essential for the customer to view these products. A lot of E-commerce websites fail to fulfill this especially when viewed on a mobile phone.

Solution:

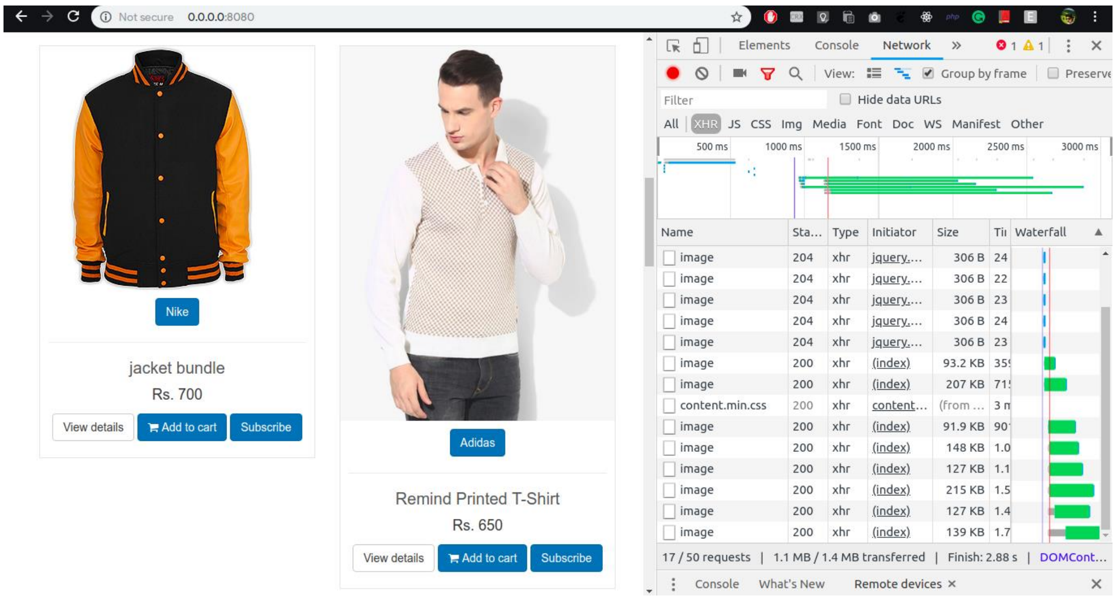
We use asynchronous requests called via JavaScript(AJAX) to another server that serves images as encoded strings. These strings are then rendered on the front-end by using Document Object Model in JavaScript. The server is a Node.JS server, and has high concurrent request serving capability. The image is resized to an optimal level based on the size of the screen the user is viewing, one of three categories - Laptop/Monitor, Tablet, Mobile Phone. The resized images have lower sizes, thus making the website load faster on devices that don’t need extra resolution pics to get a clear view.

1.2. Sample code

```
      <script>
            var wid = screen.availWidth;
            var level = 0;
            if (wid <= 425) level = 3;
            else if (wid <= 768) level = 2;
            else if (wid <= 1440) level = 1;
            else level = 0;
            $(document).ready(function () {
                $.ajax({
                    url: 'http://0.0.0.0:3000/image',
                    type: 'POST',
                    data: JSON.stringify({ image_name: '" . $pro_img1
. "', level: level }),
                    contentType: 'application/json; charset=utf-8',
                    success: function (response) {

document.getElementById('imageId_$pro_id').src = response.data;
                    }
                }));
            });
        </script>
```

1.3.Sample Screenshot

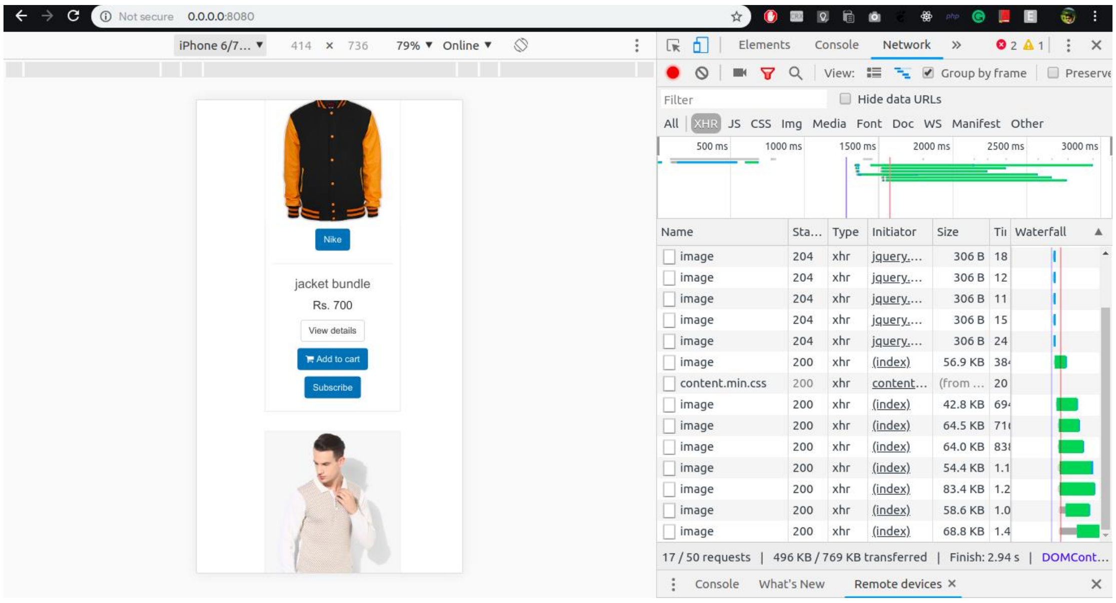
For a normal screen



Name	Sta...	Type	Initiator	Size	Ti	Waterfall
image	204	xhr	jquery....	306 B	24	
image	204	xhr	jquery....	306 B	22	
image	204	xhr	jquery....	306 B	23	
image	204	xhr	jquery....	306 B	24	
image	204	xhr	jquery....	306 B	23	
image	200	xhr	(index)	93.2 KB	35	
image	200	xhr	(index)	207 KB	71	
content.min.css	200	xhr	content...	(from ...	3 m	
image	200	xhr	(index)	91.9 KB	90	
image	200	xhr	(index)	148 KB	1.0	
image	200	xhr	(index)	127 KB	1.1	
image	200	xhr	(index)	215 KB	1.5	
image	200	xhr	(index)	127 KB	1.4	
image	200	xhr	(index)	139 KB	1.7	

17 / 50 requests | 1.1 MB / 1.4 MB transferred | Finish: 2.88 s | DOMCont...

For a mobile screen. Notice the difference in sizes.



Name	Sta...	Type	Initiator	Size	Ti	Waterfall
image	204	xhr	jquery....	306 B	18	
image	204	xhr	jquery....	306 B	12	
image	204	xhr	jquery....	306 B	11	
image	204	xhr	jquery....	306 B	15	
image	204	xhr	jquery....	306 B	24	
image	200	xhr	(index)	56.9 KB	38	
content.min.css	200	xhr	content...	(from ...	20	
image	200	xhr	(index)	42.8 KB	69	
image	200	xhr	(index)	64.5 KB	71	
image	200	xhr	(index)	64.0 KB	83	
image	200	xhr	(index)	54.4 KB	1.1	
image	200	xhr	(index)	83.4 KB	1.2	
image	200	xhr	(index)	58.6 KB	1.0	
image	200	xhr	(index)	68.8 KB	1.4	

17 / 50 requests | 496 KB / 769 KB transferred | Finish: 2.94 s | DOMCont...

2. Notify customers when there a price change of the product they wanted to buy

2.1. Problem description and its Solution

Problem Description:

A customer usually puts an item in the wishlist as he/she waits for a discount or price drop on the product. Whenever there a sale on the website, customers view their wishlist and purchase their products. However, in order to do so, they need to open and view their wishlist constantly. A customer can miss the opportunity of buying their product on sale because they failed to view their wishlist that day.

Solution:

To solve this problem, we give customers the option to “Subscribe” to a product instead of having a wishlist. Whenever there is a change in the price of a product, all the customers that are subscribed to the product are notified through email. To do this, we created a Subscribers table in our SQL database where the product ID and the email of the subscriber is stored. When the administrator makes a change in a product price, all the customers subscribed to that particular product are sent an email from another server, requests to which are made from the website.

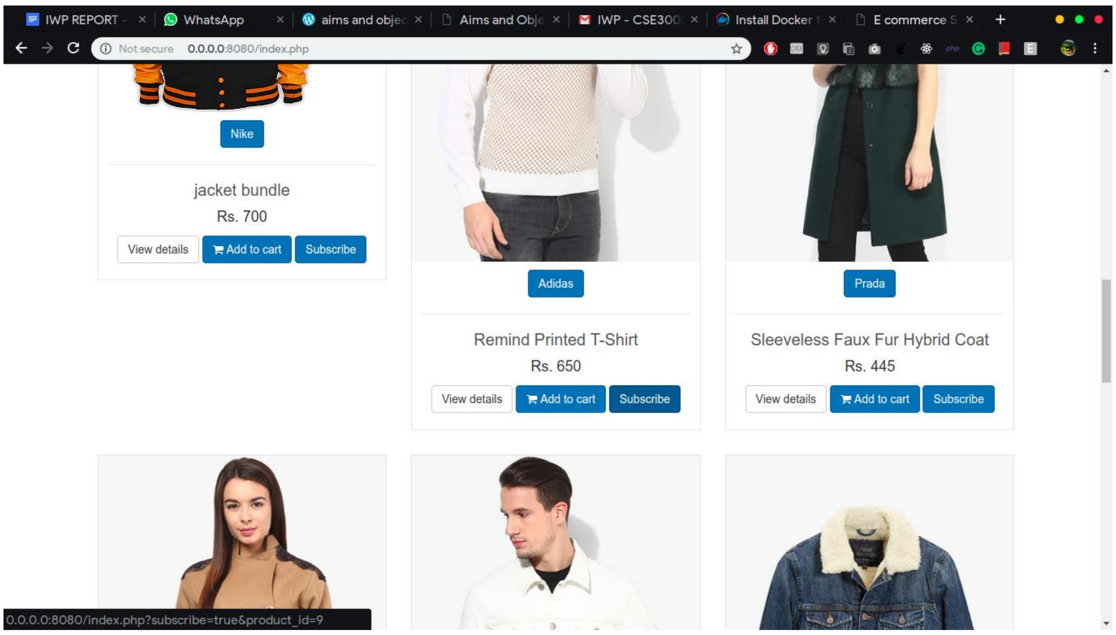
2.2. Sample code

```
function addSubscriber($product_id, $customer_id)
{
    echo                       "<script>alert('$product_id,
$customer_id')</script>";
    global $db;
    $query = "insert into subscribers values($product_id,
'$customer_id')";
    $result = mysqli_query($db, $query);
    if ($result) {
        echo  "<script>alert('You  have  subscribed  to  the
Product!')</script>";
    } else {
        echo "<script>alert('Failed!')</script>";
    }
}
```

Mail Code

```
$query = "select subscriber_id from subscribers where product_id
= $p_id";
           $run_mail = mysqli_query($con, $query);
            if ($run_mail) {
                $subscribers = mysqli_fetch_all($run_mail);
                foreach ($subscribers as $key => $value) {
                    echo
                        "
                    <script>
                        $.ajax({
                            url: 'http://mysql:3000/mail',
                            type: 'POST',
                            data:  JSON.stringify({  to_id:  '"  .
$value[0]  .  "',  product_title  :  '$product_title',  'new_price':
'$product_price' }),
                            contentType:        'application/json;
charset=utf-8',
                            success: function (response) {
                                console.log(response);
                            }
                        });
                        </script>
                        ";
                }
            }
```

2.3. Sample Screenshot



3. Complete seller details at one place

3.1. Problem description and its Solution

Problem Description:

There are different sellers for the same product available on E-commerce website. The existing websites select a seller by default for a customer using their recommendation algorithms. Usually, the seller selected is the one that is providing the product at the lowest price. However, another seller might be providing the product at an earlier delivery time (which might be the need of the customer) with a slightly higher price or there may be a seller that is more reliable having a higher rating. To view other sellers, the customer usually has to select the product and then open each seller in order to see their delivery time or ratings. This is tedious and is often overlooked by buyers.

Solution:

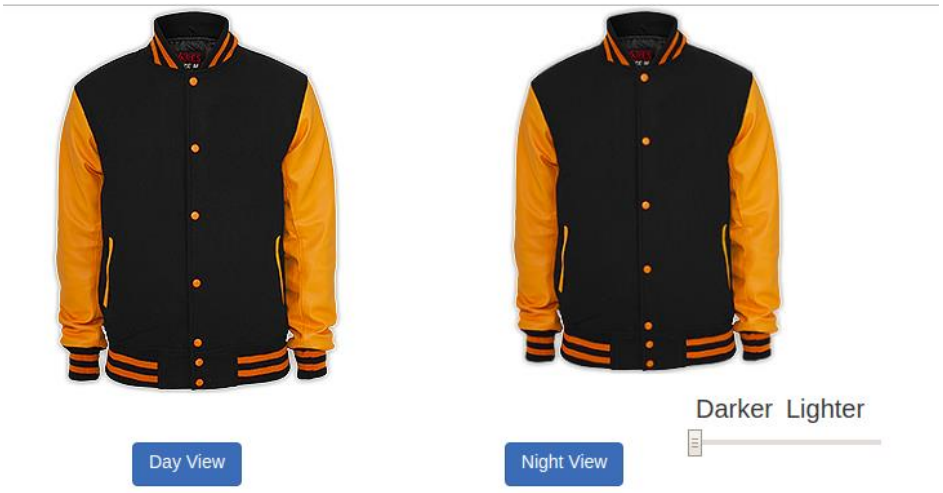
` To solve this problem, we created a “Seller” table in the SQL database that stores the seller id, the product they sell, their delivery charge, estimated shipping time and rating. When a product is viewed, all the details of the sellers selling that product are displayed right below the product. That way, the customer can choose a seller according to their need whether it be the one with the lowest price, earliest shipping time or highest ratings.

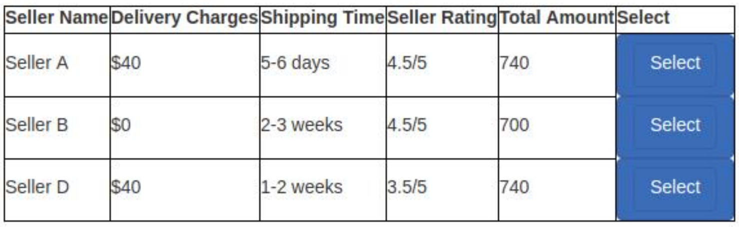
3.2. Sample code

```
//The Seller Table
CREATE TABLE IF NOT EXISTS `seller` (
`seller_id` int(10) NOT NULL,
  `seller_title` text NOT NULL,
  `seller_price` int(10) NOT NULL,
  `seller_time` text NOT NULL,
  `seller_rating` text NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=8;

//Sample values of the seller table
INSERT INTO `seller`
       (`seller_id`,`seller_title`,`seller_price`,`seller_time`,`selle
       r_rating`) VALUES
(4, 'Seller C',50, '1-2 weeks','4/5' ),
(4, 'Seller B',0, '2-3 weeks','4.5/5' ),
(5, 'Seller A',40, '5-6 days','4.5/5' ),
(6, 'Seller B',0, '2-3 weeks','4.5/5' ),
(7, 'Seller A',40, '5-6 days','4.5/5' ),
(7, 'Seller B',0, '2-3 weeks','4.5/5' );

//Displaying the seller details for a particular product
$get_prod = "select * from products where product_id=9";
$run_prod = mysqli_query($db, $get_prod);
   while($row_prod = mysqli_fetch_array($run_prod)){
       $pro_id= 9;
       $pro_price=$row_prod['product_price'];
}

$get_seller = "select * from seller WHERE seller_id=5";
$run_seller = mysqli_query($db, $get_seller);
echo "<table>
       <tr><th>Seller Name</th><th>Delivery
              Charges</th><th>Shipping Time</th><th>Seller
              Rating</th><th>Total Amount</th><th>Select</th></tr>";
while($row_seller = mysqli_fetch_array($run_seller)){
       $s_price=$row_seller['seller_price'];
       $amt=$s_price+$pro_price;
       echo   "<tr><td>".$row_seller['seller_title']."</td><td>$"
              .$row_seller['seller_price']."</td><td>".$row_seller['se
       ller_time']
```

```
        ."</td><td>".$row_seller['seller_rating']."</td><td>".$a
mt."</td>     <td class='btn btn-primary sel'>
<a href='../cart.php?itemId=$pro_id&quantity=1&price=$amt
        &size=Medium' class='btn btn-primary' style='text-
decoration:   none;'>Select</a></td></tr>";
}
echo "</table>";
```

3.3. Sample Screenshot





Seller Name	Delivery Charges	Shipping Time	Seller Rating	Total Amount	Select
Seller A	$40	5-6 days	4.5/5	740	Select
Seller B	$0	2-3 weeks	4.5/5	700	Select
Seller D	$40	1-2 weeks	3.5/5	740	Select

4. Zero Downtime Ensurance

4.1. Problem description and its Solution

Problem Description:

A lot of E-commerce websites are pulled down when they are under maintenance. This is very inconvenient to customers. A website can be down due to several reasons, such as database migration.

Solution:

We integrate a containerisation software along with a clustering tool, that maintains the state of the multiple services such as apache, php, mysql and the static image server. These services are run independently, and can be scaled to have as many replicas as required amongst manager and worker nodes. The clustering tool is Docker Swarm and the containerization software is Docker. Thus by having more than 1 replica, in the case of crash or server maintenance, the older server instance can be kept running and once the updates are performed, they can be updated separately.

4.2. Sample code

- The shell/bash script for setting up the ensemble.

```
#!/bin/bash
# set -e
docker swarm leave -f

docker swarm init
docker network create --attachable --driver overlay --
subnet=10.200.1.0/24 testnet

docker stack deploy -c docker-compose.yml test
```

- The DockerFile of php that is used to build the images for further use

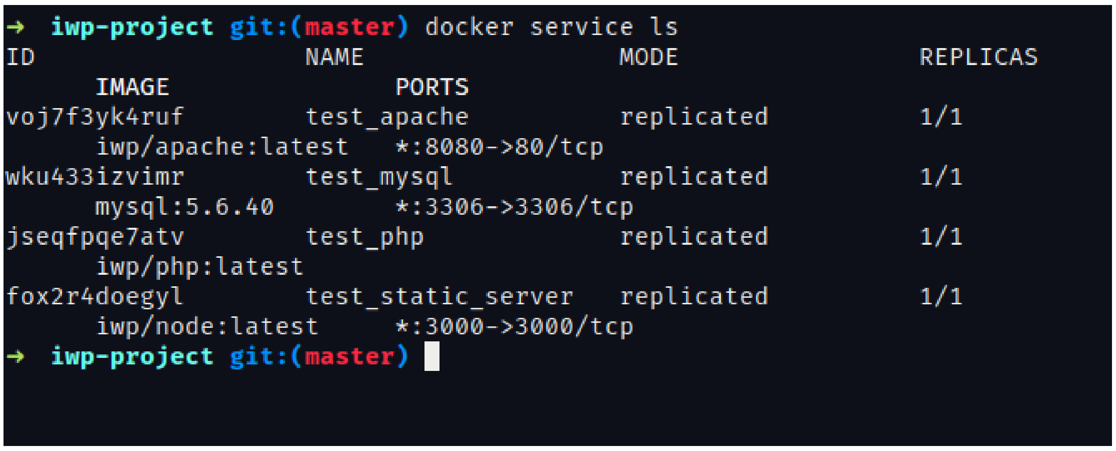
```
FROM php:7.2.7-fpm-alpine3.7
```

```
RUN apk update; \
apk upgrade;
RUN docker-php-ext-install mysqli
```

- The docker-compose file to specify deployment constraints

```
  php:
    deploy:
      replicas: 1
      restart_policy:
        condition: on-failure
        delay: 5s
        max_attempts: 3
    image: iwp/php
    networks:
      testnet:
        aliases:
          - php
    hostname: php
    volumes:
      - ./public_html/:/var/www/html/
```

4.3. Sample Screenshot



```
➜ iwp-project git:(master) docker service ls
ID                  NAME                MODE                REPLICAS
      IMAGE               PORTS
voj7f3yk4ruf        test_apache         replicated          1/1
      iwp/apache:latest   *:8080->80/tcp
wku433izvimr        test_mysql          replicated          1/1
      mysql:5.6.40        *:3306->3306/tcp
jseqfpqe7atv        test_php            replicated          1/1
      iwp/php:latest
fox2r4doegyl        test_static_server  replicated          1/1
      iwp/node:latest     *:3000->3000/tcp
➜ iwp-project git:(master)
```

5. Day and Night view of products

5.1. Problem description and its Solution

Problem Description:

One of the main drawbacks of online shopping is that a customer has to analyze whether a product will look good on them by just looking at pictures. Usually, all clothing products listed in E-commerce websites have a white background that gives the buyers an idea as to how the product will look in daytime and normal lighting. They can't know how the product will look in the evening or at night or in dim lightings.

Solution:

In order to assist a buyer when they're deciding if clothing product will suit them or not, we have provided a Day and Night View of the product. The 'day view' is how a product is usually listed on all websites, in a white background. The 'night view' provides a view of the product in different lightings of evening and night. This is done by placing a blue-coloured partially transparent element on top of the product image. The buyer can change the shade of the semi-transparent blue layover on the product image so as to get a view of the product in different lightings. A division with reduced opacity is placed over the product image. The background colour of the division is given the red and green value as 0 (from RGB values). The value of blue can be changed from 0 to 255 by the user using a slider to change the shade of blue.

5.2. Sample code

```
//JS for changing shade of blue
var slide = document.querySelector('#night'),
    b = document.querySelector('#b'),
    b_out = document.querySelector('#b_out');

function setColor(){
  var b_hex = parseInt(b.value,10).toString(16);
      if(b_hex.length<2)
```

```
         {
           b_hex="0"+n;
         }
      var hex = "#0000" + b_hex;     //RBG value set where R=0, G=0,
        B=input
  slide.style.backgroundColor = hex;
}

b.addEventListener('input', function() {
  setColor();
});
```

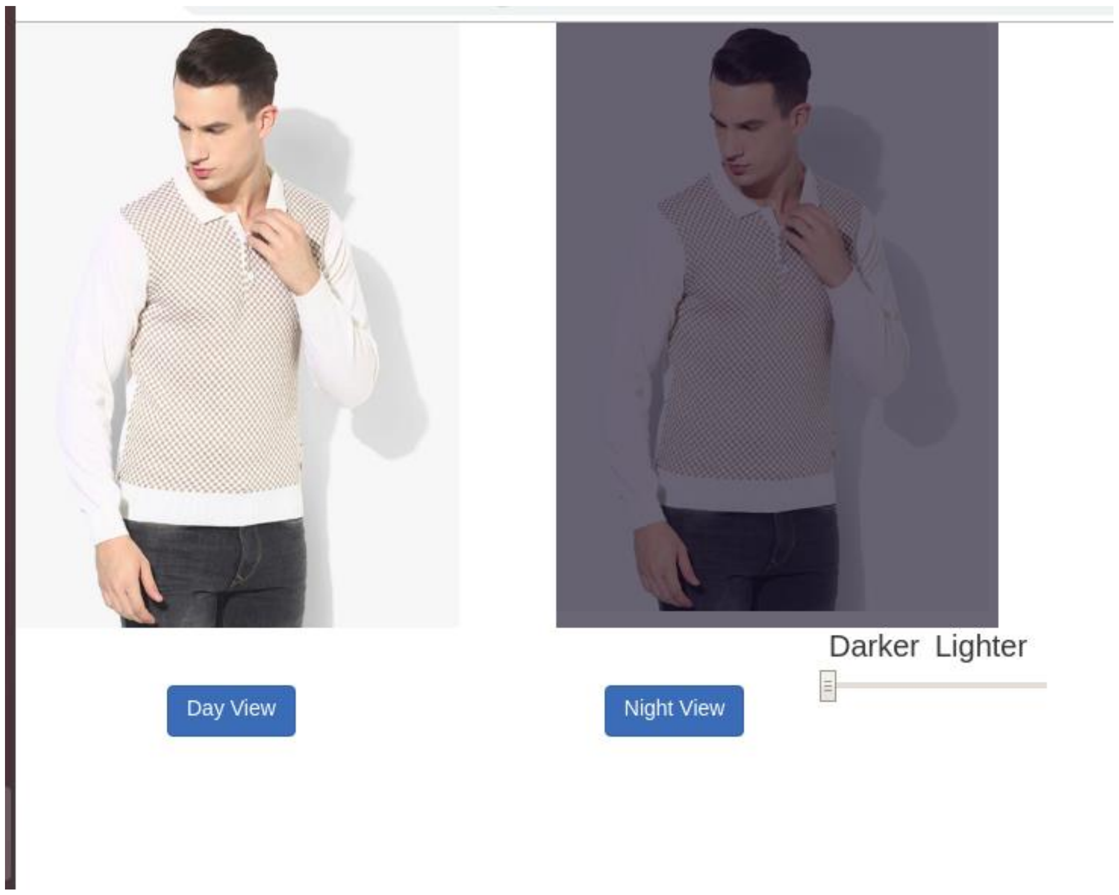
//HTML for a product view

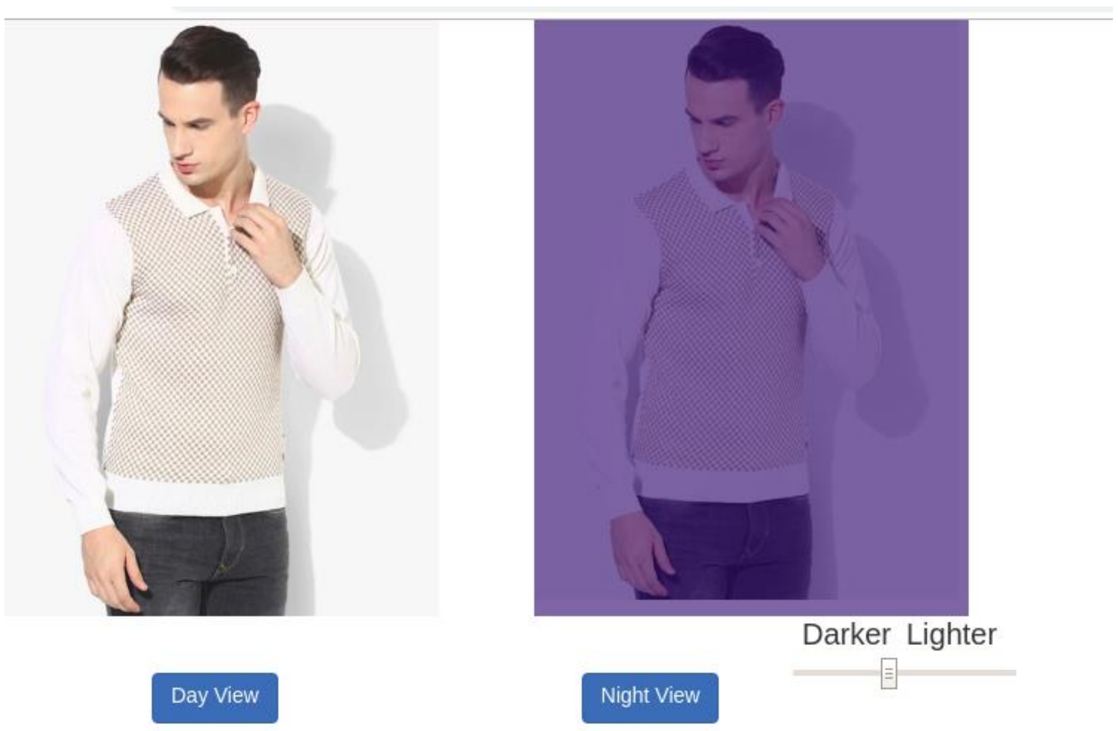
```
<div class='product' >
       <img src='product-1.jpg' style='float:left;height:400px;'>
       <div class='container'>
         <img src='product-1.jpg' class='image'>
         <div class='night' id='night'>
         </div>
       </div>
<p class='btn btn-primary btn1'>Day View</p>
<p class='btn btn-primary btn1 btn2'>Night View
       <fieldset class='field' style='display:inline-block;font-
       size:20px;'>
         <label for='b'></label>Darker&ensp;Lighter
         //User inputs blue colour value in RBG
         <input type='range' min='0' max='255' id='b' step='1'
       value='0'>
         <output for='b' style='font-size:20px;'></output>
       </fieldset>
</p>
<br><br>
</div>
```

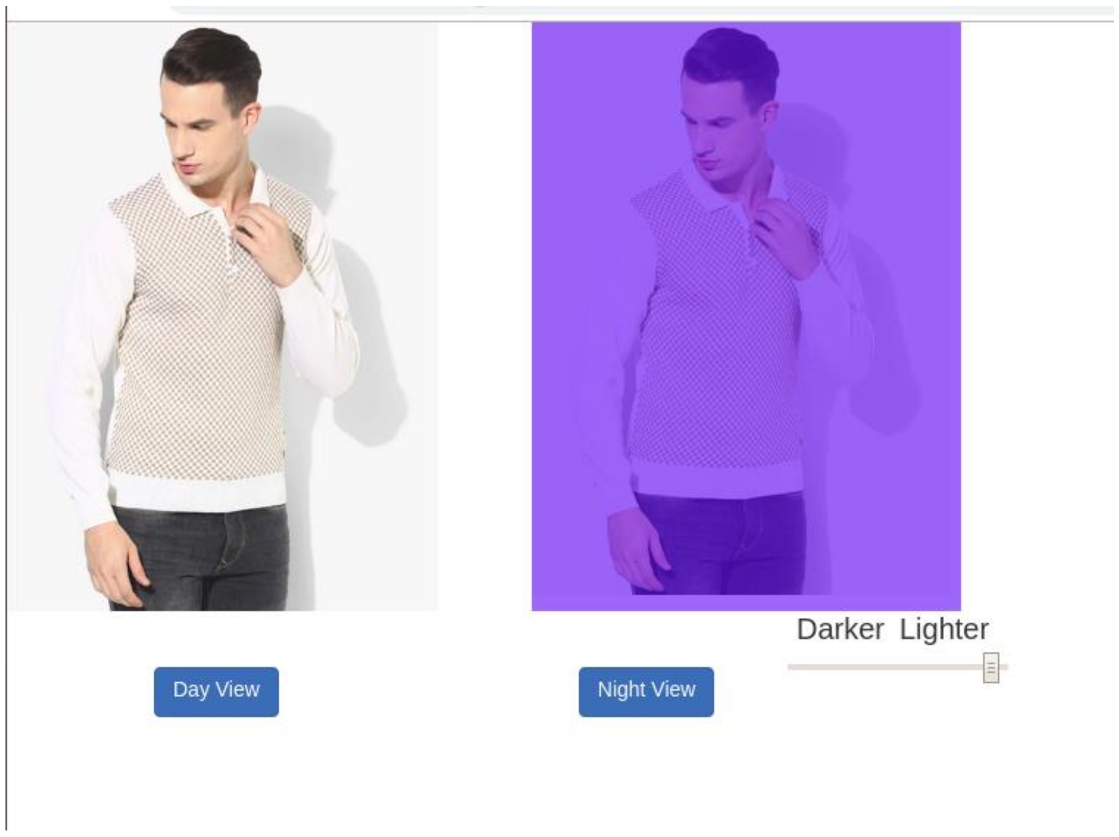
//CSS for a product view

```
.container {
         position: relative;}
.image {
         display: inline;
         height: auto;}
.night {
         position: absolute;
         height: 100%;
         opacity: 0.6;
         transition: 0.5s ease;}
```

5.3. Sample Screenshot







6. Live Map for customer during product delivery

6.1. Problem description and its Solution

Problem Description:

It is crucial to the customers to track where the shipment of the package is present, to judge whether they will be there to collect it or not, or delegate the job to a neighbour. Textual updates that are in current ecommerce websites don't do the job well enough, only sending text updates.

Solution:

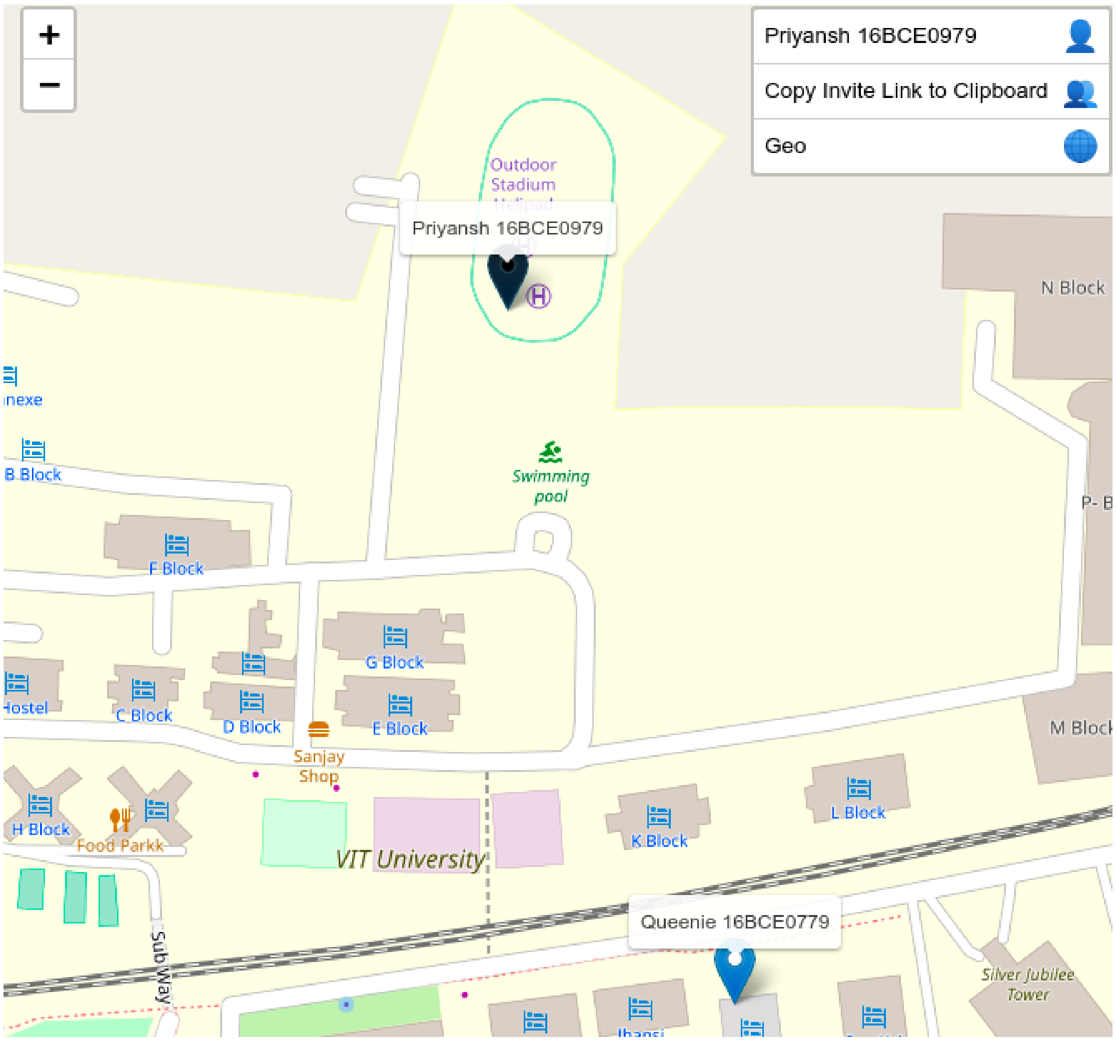
We developed a seperate real-time location sharing platform, geoshare, that helps the delivery/shipping agent provide location of the products being shipped in realtime, in the browser without the need for an app. It uses LeafletJS, OpenStreetMaps, and Agora Signalling API for the real time communication. You can either be the host, or simply a guest of any other person that shared a unique URL with you.

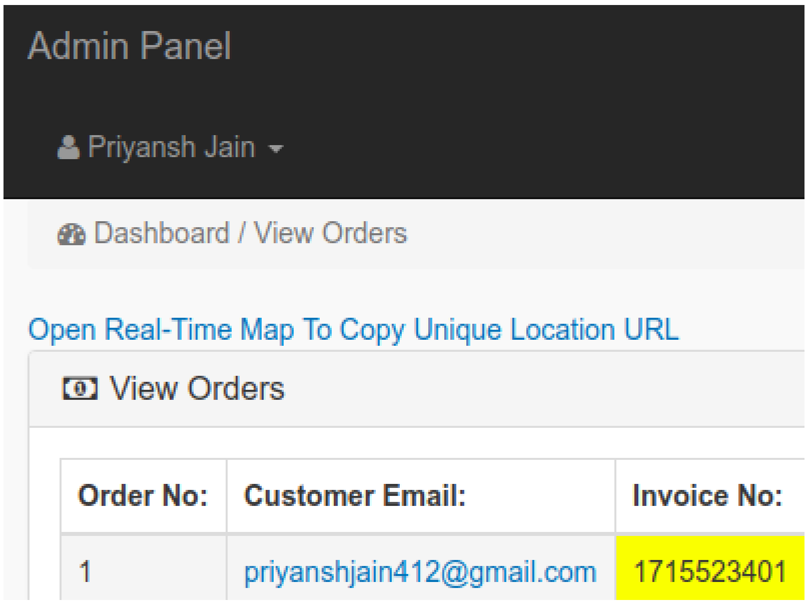
6.2. Sample code

```
document.addEventListener(
  "DOMContentLoaded",
  event => {
    // add the div map to the body
    hyperHTML(document.body)`<div id="map"></div>`;

    // both leaflet and pusher might have
    // some asynchronous initialization
    // wait for both of them, then kickstart the App
    Promise.all([leaflet.init("map"), agora.init()]).then(
      ([map, agoraInitResult]) => {
        // settings and messaging
        // are two Leflet plugins
        const channel = agoraInitResult[0];

        const session = agoraInitResult[1];
        let splugin, mplugin;
        const me = {
          id: USER_ACCOUNT
        };
        // the list of users on the map
        const users = Object.create(null);

        const triggerMessage = (message, info) => {
          channel.messageChannelSend(
            JSON.stringify({ message: message, info: info
})
          );
        };

        channel.onMessageChannelReceive = function(account,
uid, msg) {
          const { message, info } = JSON.parse(msg);
          if (message && info) {
```

```
        if (message.indexOf("client-update-") !== -1) {
          info.forEach(client => updateClient(client));
        } else if (message === "client-show-tooltip") {
          showTooltip(info);
        } else if (message === "client-update") {
          updateClient(info);
        } else if (message === "client-message") {
          mplugin.showMessage(info);
        } else if (message === "client-drop-tooltip") {
          dropTooltip(info);
        } else {
          console.log("Invalid");
        }
      }
      console.log(message, info);
    };

    // each user/member has an id, a name,
    // a marker on the map, an isHost info,
    // and a reference to the channel.
    const addMember = member => {
      users[member.id] = {
        id: member.id,
        name:
          member.id === me.id
            ? storage.get("name", IS_GUEST ? "guest" :
"host")
            : "guest",
        marker: null,
        isHost: !IS_GUEST && member.id === me.id,
        tooltip: me.tooltip && me.tooltip.info,
        triggerMessage: triggerMessage
      };
```

6.3. Sample Screenshot





Admin Panel

Priyansh Jain

Dashboard / View Orders

Open Real-Time Map To Copy Unique Location URL

View Orders

Order No:	Customer Email:	Invoice No:
1	priyanshjain412@gmail.com	1715523401

RECENT TECHNOLOGIES

1. Bootstrap

Bootstrap is a free front-end framework for faster and easier web development. Bootstrap includes HTML and CSS based design templates for forms, buttons, navigation, image carousels and many other. It gives the ability to easily create responsive designs.

1.1. Importance of the technology

Responsive Web Design is about creating web sites which automatically adjust themselves to look good on all devices, from small phones to large desktops.

Bootstrap's **grid system** is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small screen it would be better if the content items were stacked on top of each other. Bootstrap uses the grid system which allows up to 12 columns across the page. If we do not want to use 12 columns individually, we can group the columns together to create wider windows. The Bootstrap 4 grid system has **five classes**: .col-xs-, .col-sm-, .col-md-, .col-lg-, .col-xl-

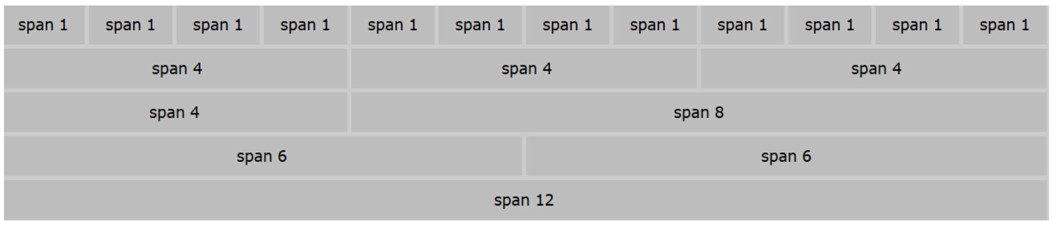


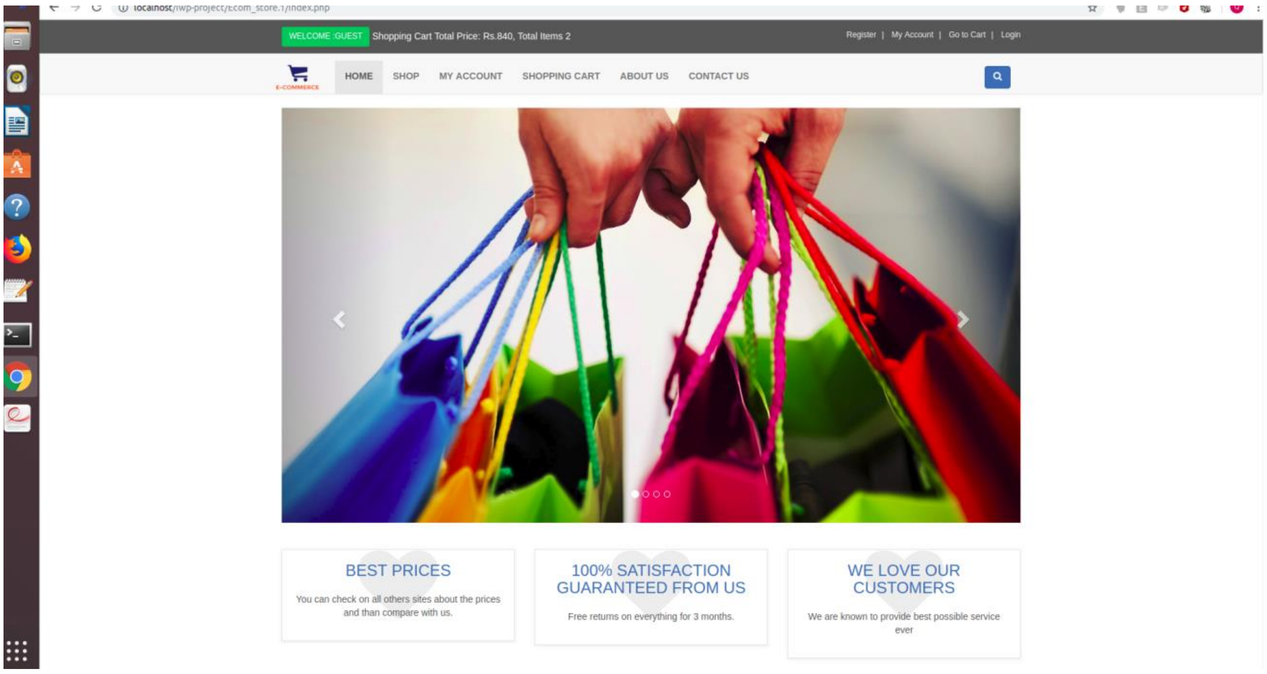
Figure : Grid System of Bootstrap

Bootstrap is developed *mobile first*. To ensure proper rendering and touch zooming for all devices, **add the responsive viewport meta tag** to <head> tag.

1.2. Sample code

```
<div id="content" >
<div class="container" >
<div class="col-md-12" >
<ul class="breadcrumb" >
<li>
<a href="index.php">Home</a>
</li>
<li>Contact Us</li>
</ul>
</div>
<div class="col-md-12" >
<div class="box animated zoomIn" >
<div class="box-header" >
<center>
<?php
```

1.3. Sample Screenshot


localhost/rwp-project/Ecom_store.1/index.php
WELCOME :GUEST
Register | My Account | Go to Cart | Login
E-COMMERCE
HOME
SHOP
MY ACCOUNT
SHOPPING CART
ABOUT US
CONTACT US

Shopping Cart Total Price: Rs.840, Total Items 2

BEST PRICES

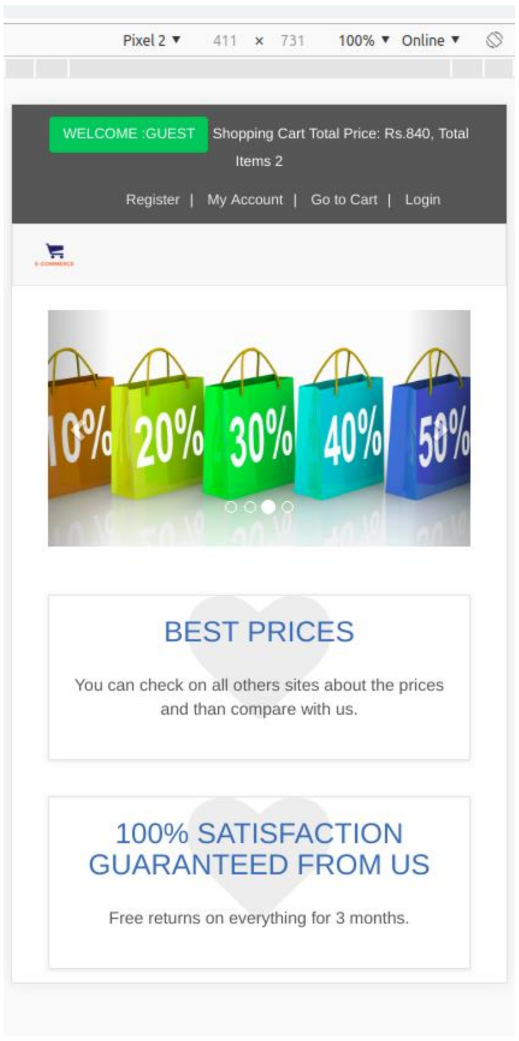
You can check on all others sites about the prices and than compare with us.

100% SATISFACTION GUARANTEED FROM US

Free returns on everything for 3 months.

WE LOVE OUR CUSTOMERS

We are known to provide best possible service ever



WELCOME :GUEST Shopping Cart Total Price: Rs.840, Total Items 2

Register | My Account | Go to Cart | Login

BEST PRICES

You can check on all others sites about the prices and than compare with us.

100% SATISFACTION GUARANTEED FROM US

Free returns on everything for 3 months.

2. JQuery

jQuery is a lightweight, "write less, do more", JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

2.1. Importance of the technology

The purpose of jQuery is to make it much easier to use JavaScript on the website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that one can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

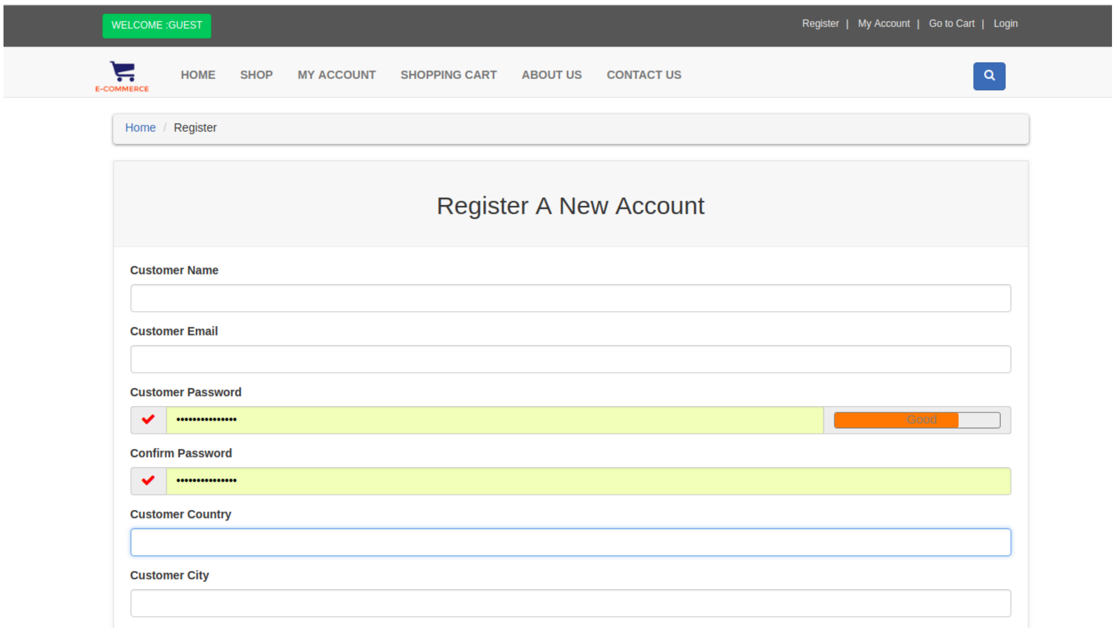
- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

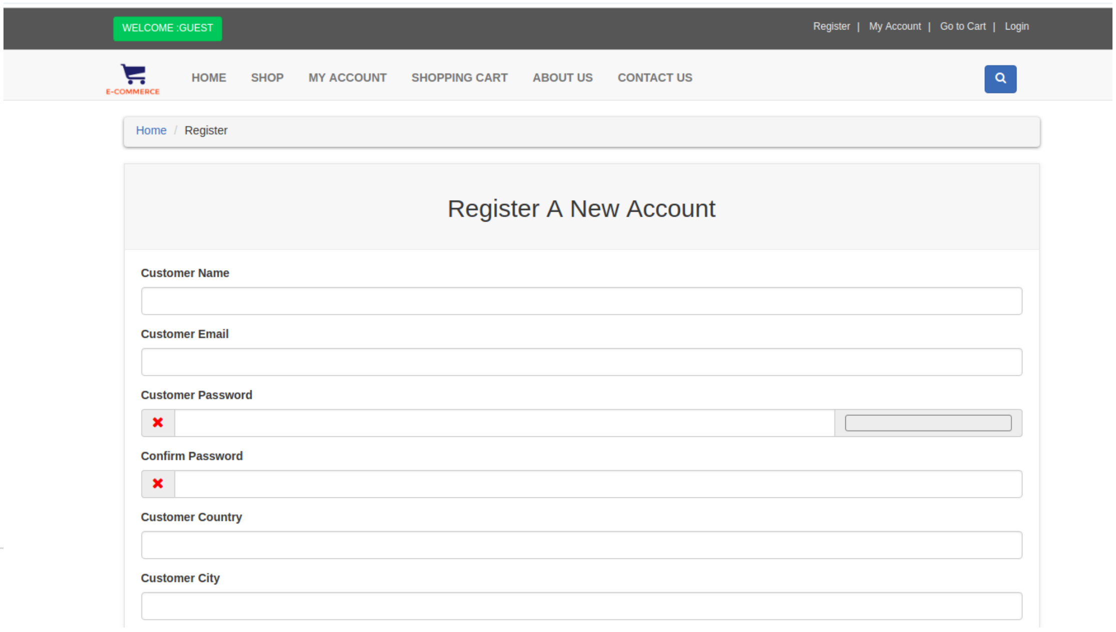
In addition, jQuery has plugins for almost any task out there

2.2. Sample code

```
$(document).ready(function(){
$('.tick1').hide();

$('.cross1').hide();

$('.tick2').hide();

$('.cross2').hide();

$('.confirm').focusout(function
(){
var password =
$('#pass').val();
var confirmPassword =
$('#con_pass').val();
if(password ==
confirmPassword){
$('.tick1').show();

$('.cross1').hide();

$('.tick2').show();

$('.cross2').hide();

}

else{

$('.tick1').hide();

$('.cross1').show();

$('.tick2').hide();

$('.cross2').show();
}
});
});
</script>
<script>
$(document).ready(function(){
$("#pass").keyup(function(){
check_pass();
});
});
```

2.3. Sample Screenshot





3. AngularJS

AngularJS is a JavaScript framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It extends HTML attributes with Directives, and binds data to HTML with Expressions. The resulting environment is extraordinarily expressive, readable, and quick to develop.

3.1. Importance of the technology

The main aspect of AngularJS in the project was Form Validation. AngularJS offers **client-side form validation**. It monitors the state of the form and input fields (input, textarea, select), and lets us notify the user about the current state. AngularJS also holds information about whether they have been touched, or modified, or not.

In AngularJS forms, input controls provides data-binding by using the **ng-model** directive.

```
<input type="text" ng-model="username">
```

We can use standard HTML5 attributes to validate input, or we can make our own validation functions. Angular provides **properties** on forms that help us validate them. They give us various information about a form or its inputs and are **applied to a form and inputs**.

Angular also provides classes on the form and its inputs so that you can style each state accordingly.

Property	Class	Description
$valid	ng-valid	*Boolean* Tells whether an item is currently valid based on the rules you placed.

$invalid	ng-invalid	*Boolean* Tells whether an item is currently invalid based on the rules you placed.
$pristine	ng-pristine	*Boolean* True if the form/input **has not** been used yet.
$dirty	ng-dirty	*Boolean* True if the form/input **has** been used.
$touched	ng-touched	*Boolean* True if the input **has** been blurred.

Table : Form Properties in AngularJS classes

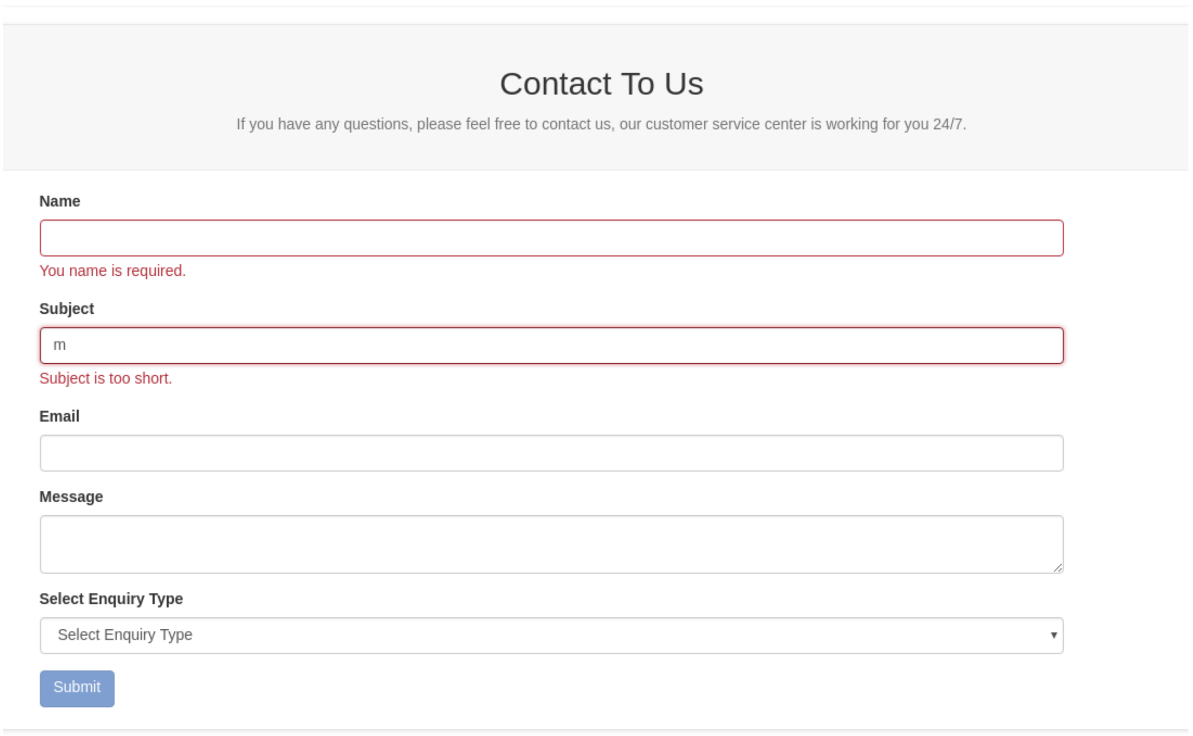
3.2. Sample code

```
//Using AngularJS
<head><script
         src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angu
       lar.min.js">
</script></head>

//HTML for Contact Us form using angularJS
<div ng-app="validationApp" ng-controller="mainController">
    <form name="userForm" ng-submit="submitForm()" novalidate>

        <!-- NAME -->
        <div class="form-group" ng-class="{ 'has-error' :
       userForm.name.$invalid && !userForm.name.$pristine }">
            <label>Name</label>
            <input type="text" name="name" class="form-control" ng-
       model="user.name" required>
            <p ng-show="userForm.name.$invalid &&
       !userForm.name.$pristine" class="help-block">You name is
       required.</p>
        </div>
              <!-- SUBJECT-->
        <div class="form-group" ng-class="{ 'has-error' :
       userForm.subject.$invalid && !userForm.subject.$pristine }">
            <label>Subject</label>
```

```
     <input type="text" name="subject" class="form-control" ng-
model="user.subject" ng-minlength="3" ng-maxlength="15">
     <p ng-show="userForm.subject.$error.minlength" class="help-
block">Subject is too short.</p>
     <p ng-show="userForm.subject.$error.maxlength" class="help-
block">Subject is too long.</p>
 </div>
 <!-- EMAIL -->
 <div class="form-group" ng-class="{ 'has-error' :
userForm.email.$invalid && !userForm.email.$pristine }">
     <label>Email</label>
     <input type="email" name="email" class="form-control" ng-
model="user.email">
     <p ng-show="userForm.email.$invalid &&
!userForm.email.$pristine" class="help-block">Enter a valid
email.</p>
 </div>
…..
<button type="submit" class="btn btn-primary" ng-
disabled="userForm.$invalid">Submit</button>
```

```
//JS for Contact Us form using AngularJS
<script src="js/jquery.min.js"> </script>
<script>
       var validationApp = angular.module('validationApp', []);// create
       angular app
       // create angular controller
       validationApp.controller('mainController', function($scope) {
            $scope.submitForm = function() {// submit the form after all
       validation
         if ($scope.userForm.$valid) {     // check form is completely
       valid
                alert('our form is submitted');
                }};});</script>
```

3.3. Sample Screenshot


Contact To Us

If you have any questions, please feel free to contact us, our customer service center is working for you 24/7.

Name

You name is required.

Subject

m

Subject is too short.

Email

Message

Select Enquiry Type

Select Enquiry Type

Submit

4. Node.js

4.1. Importance of the technology

Node.js is a server-side platform that is built using Chrome’s JavaScript V8 engine. It allows for asynchronous operations, and has support for HTTP servers. We use the technology to send requests with information such as the screen width and the image name, so the server can asynchronously perform the operation on the image and return the response. Jimp is the library being used for image alteration, and ExpressJS is used for making the server listen for HTTP requests.

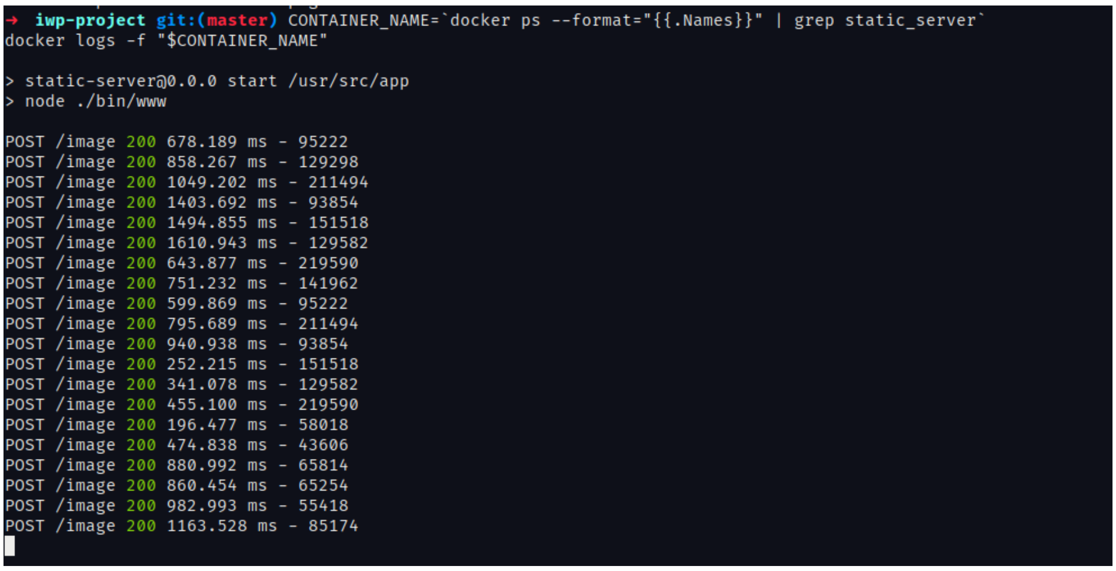
4.2. Sample code

```
router.post("/image", (req, res, next) => {
  const level = parseInt(req.body.level);
  const fpath = path.join(__dirname, "..", "images",
req.body.image_name);
  return Jimp.read(fpath, (err, image) => {
    if (err) res.json({ success: false, err: err });
    let scaleWidth = 0;
    if (level === 3) {
      scaleWidth = image.bitmap.width > 225 ? 225 :
image.bitmap.width;
```

```
  } else if (level === 2) {
   scaleWidth = image.bitmap.width > 425 ? 425 :
image.bitmap.width;
  } else if (level === 1) {
   scaleWidth = image.bitmap.width > 625 ? 625 :
image.bitmap.width;
  }
  image = image.resize(scaleWidth, Jimp.AUTO);
  return image.getBase64(Jimp.AUTO, (err, bitmap) => {
   return res.json({
    data: bitmap
   });
  });
 });
});
```

4.3. Sample Screenshot

This is a screenshot of the server logs, a POST request is being made for every image.



```
➜  iwp-project git:(master) CONTAINER_NAME=`docker ps --format="{{.Names}}" | grep static_server`
docker logs -f "$CONTAINER_NAME"

> static-server@0.0.0 start /usr/src/app
> node ./bin/www

POST /image 200 678.189 ms - 95222
POST /image 200 858.267 ms - 129298
POST /image 200 1049.202 ms - 211494
POST /image 200 1403.692 ms - 93854
POST /image 200 1494.855 ms - 151518
POST /image 200 1610.943 ms - 129582
POST /image 200 643.877 ms - 219590
POST /image 200 751.232 ms - 141962
POST /image 200 599.869 ms - 95222
POST /image 200 795.689 ms - 211494
POST /image 200 940.938 ms - 93854
POST /image 200 252.215 ms - 151518
POST /image 200 341.078 ms - 129582
POST /image 200 455.100 ms - 219590
POST /image 200 196.477 ms - 58018
POST /image 200 474.838 ms - 43606
POST /image 200 880.992 ms - 65814
POST /image 200 860.454 ms - 65254
POST /image 200 982.993 ms - 55418
POST /image 200 1163.528 ms - 85174
```

5. Ajax

5.1. Importance of the technology

AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

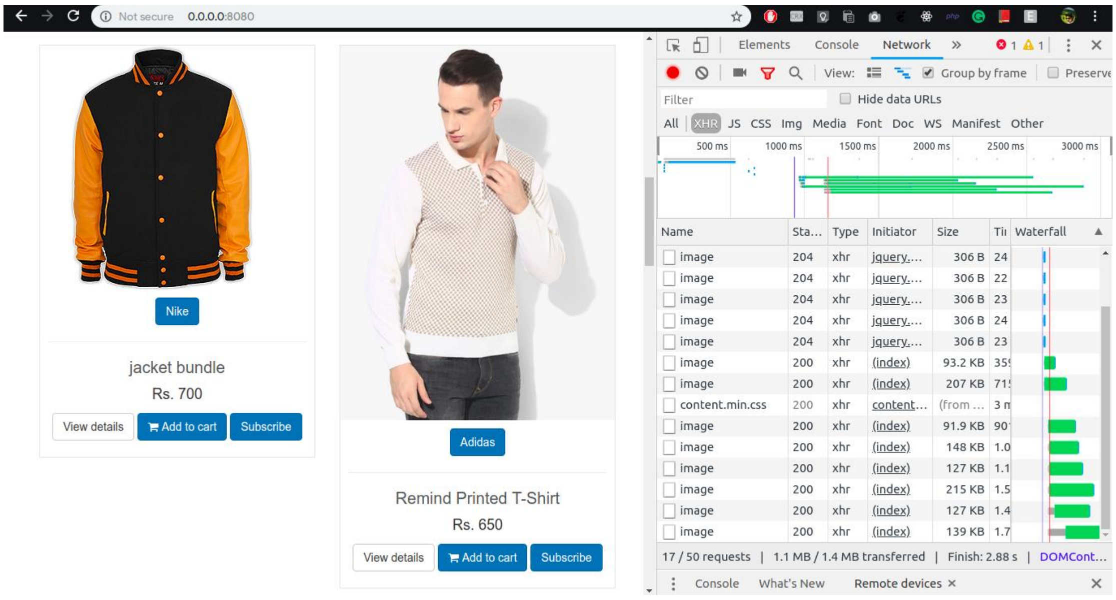
- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.

- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

5.2. Sample code

```
<script>
            var wid = screen.availWidth;
            .
            .
            $(document).ready(function () {
                $.ajax({
                    url: 'http://0.0.0.0:3000/image',
                    type: 'POST',
                    data: JSON.stringify({ image_name: '" . $pro_img1
. "', level: level }),
                    contentType: 'application/json; charset=utf-8',
                    success:       function       (response)       {
document.getElementById('imageId_$pro_id').src = response.data;
                    }
                });
            });
        </script>
```

5.3. Sample Screenshot



Name	Sta…	Type	Initiator	Size	Ti	Waterfall
image	204	xhr	jquery.…	306 B	24	
image	204	xhr	jquery.…	306 B	22	
image	204	xhr	jquery.…	306 B	23	
image	204	xhr	jquery.…	306 B	24	
image	204	xhr	jquery.…	306 B	23	
image	200	xhr	(index)	93.2 KB	35	
image	200	xhr	(index)	207 KB	71	
content.min.css	200	xhr	content…	(from …	3 m	
image	200	xhr	(index)	91.9 KB	90	
image	200	xhr	(index)	148 KB	1.0	
image	200	xhr	(index)	127 KB	1.1	
image	200	xhr	(index)	215 KB	1.5	
image	200	xhr	(index)	127 KB	1.4	
image	200	xhr	(index)	139 KB	1.7	

17 / 50 requests | 1.1 MB / 1.4 MB transferred | Finish: 2.88 s | DOMCont…

6. Docker and Docker Swarm

6.1. Importance of the technology

Docker is containerization software, where only the required dependencies for any software are bundled into a single image, and these images can be used further in any environment they are deployed to, given docker is installed. Docker swarm is a clustering tool that manages services created from these images, and is used to specify the deployment constraints for each service, like restart-on-failure policy, replicas, hostname and so on.

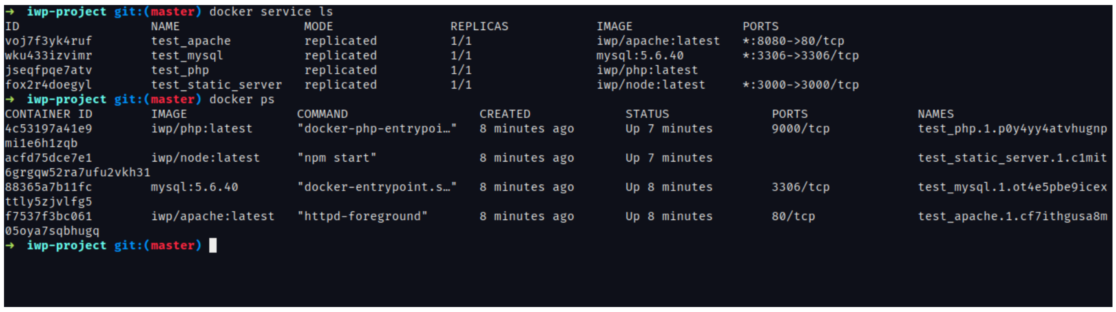
6.2. Sample code

```
version: "3.2"

networks:
  testnet:
    external:
      name: testnet

services:
  php:
    deploy:
      replicas: 1
      restart_policy:
        condition: on-failure
        delay: 5s
        max_attempts: 3
    image: iwp/php
    networks:
      testnet:
        aliases:
          - php
    hostname: php
    volumes:
      - ./public_html/:/var/www/html/

  apache:
    deploy:
      replicas: 1
      restart_policy:
        condition: on-failure
        delay: 5s
        max_attempts: 3
    image: iwp/apache
    depends_on:
      - php
      - mysql
    networks:
      testnet:
        aliases:
          - apache

    hostname: apache
    ports:
      - "8080:80"
    volumes:
      - ./public_html/:/var/www/html/
```

```
static_server:
  deploy:
    replicas: 1
    restart_policy:
      condition: on-failure
      delay: 5s
      max_attempts: 3
    placement:
      constraints:
        - node.hostname == priyansh-Inspiron-3543
  image: iwp/node
  command: npm start
  hostname: static_server
  volumes:
    - ./static_server:/usr/src/app
  ports:
    - "3000:3000"
  networks:
    testnet:
      aliases:
        - static_server
```

6.3. Sample Screenshot

Below is the screenshot of the services running in the swarm



```
➜  iwp-project git:(master) docker service ls
ID                  NAME                  MODE                REPLICAS            IMAGE               PORTS
voj7f3yk4ruf        test_apache           replicated          1/1                 iwp/apache:latest   *:8080->80/tcp
wku433izvimr        test_mysql            replicated          1/1                 mysql:5.6.40        *:3306->3306/tcp
jseqfpqe7atv        test_php              replicated          1/1                 iwp/php:latest
fox2r4doegyl        test_static_server    replicated          1/1                 iwp/node:latest     *:3000->3000/tcp
➜  iwp-project git:(master) docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
4c53197a41e9        iwp/php:latest      "docker-php-entrypoi…"   8 minutes ago       Up 7 minutes        9000/tcp            test_php.1.p0y4yy4atvhugnp
mi1e6h1zqb
acfd75dce7e1        iwp/node:latest     "npm start"              8 minutes ago       Up 7 minutes                            test_static_server.1.c1mit
6grgqw52ra7ufu2vkh31
88365a7b11fc        mysql:5.6.40        "docker-entrypoint.s…"   8 minutes ago       Up 8 minutes        3306/tcp            test_mysql.1.ot4e5pbe9icex
ttly5zjvlfg5
f7537f3bc061        iwp/apache:latest   "httpd-foreground"       8 minutes ago       Up 8 minutes        80/tcp              test_apache.1.cf7ithgusa8m
05oya7sqbhugq
➜  iwp-project git:(master) █
```

7. Agora Signalling API and LeafletJS

7.1. Importance of the technology

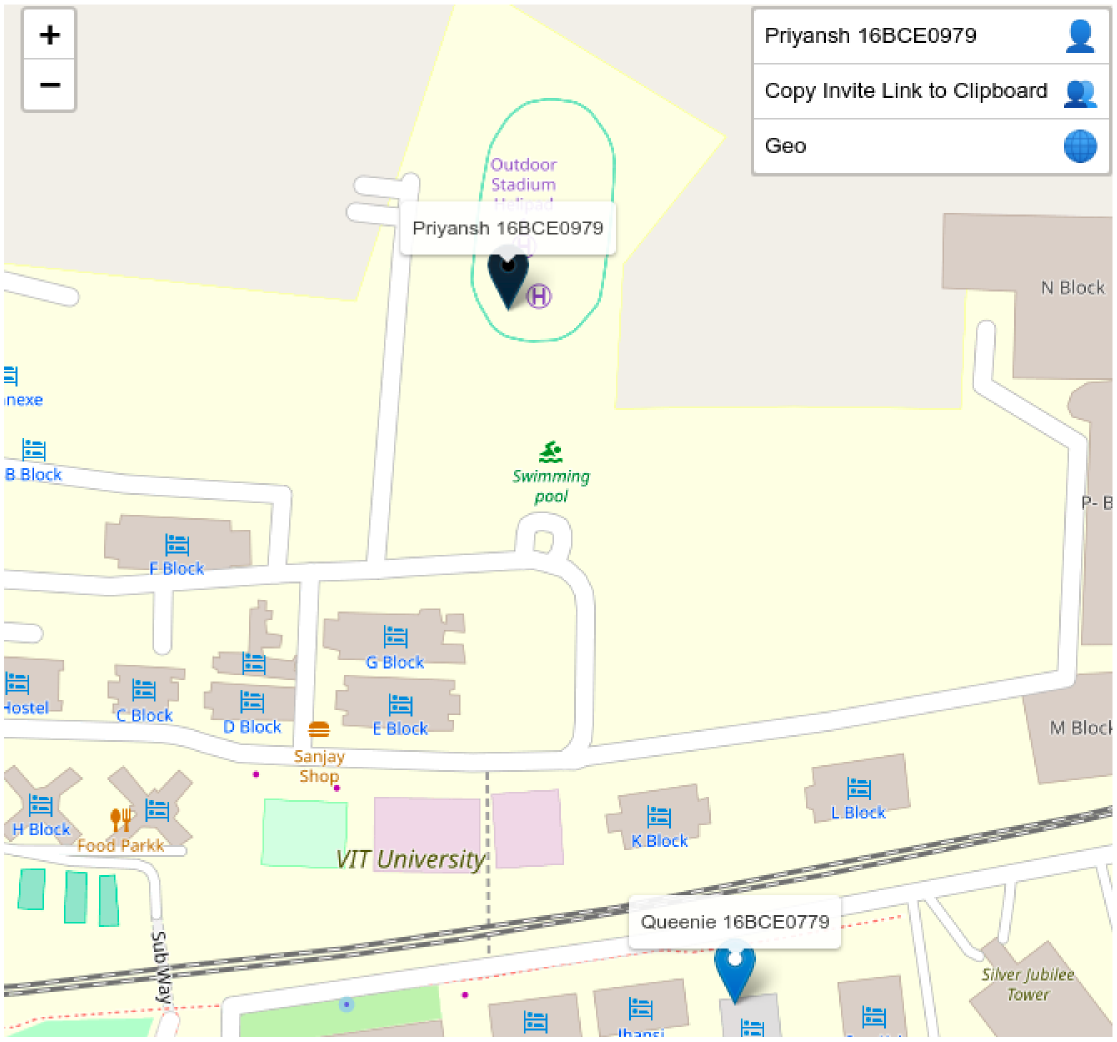
Agora Signalling API is used to create channels that users can subscribe to. When a URL is opened, a channel is created with a unique hash, and a share URL is generated based on that hash. When people with the share

URL open the website, they subscribe to the channel that the hash has. Thus all updates that include position changes, or messages, or people joining or leaving the channel, happen via the channel. Every event is pushed to the channel, and all the subscribers are notified immediately. Agora RTC uses the we sockets protocol. Leaflet JS is the mapping library used, with OpenStreetMap as the online map supplier. Thus the tooltips are plotted on the map according to the messages received from agora signalling channels.

7.2. Sample code

```
import { CHANNEL, AGORA, USER_ACCOUNT } from
"./constants.js";

let promise;

export default {
  init() {
    if (promise) return promise;

    promise = new Promise((res, rej) => {
      // debug all messages on localhost
      /* Login to the system. */
      const signal = Signal(AGORA.appId);
      var session = signal.login(USER_ACCOUNT,
"_no_need_token");
      session.onLoginSuccess = function(uid) {
        /* Join a channel and set the onChannelJoined and
onMessagChannelReceive callbacks. */
        var channel = session.channelJoin(CHANNEL);
        channel.onChannelJoined = function() {
          return res([channel, session]);
        };
      };
    });

    return promise;
  }
};
```

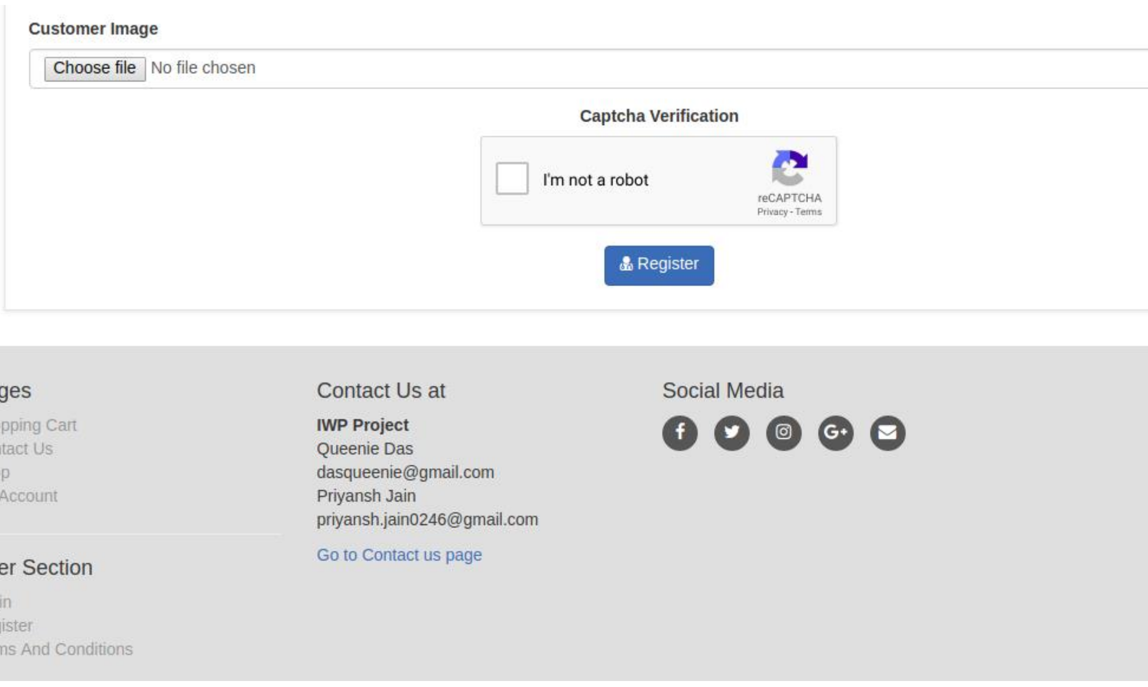
7.3. Sample Screenshot



8. Other plugins, libraries and technologies used:

i) ReCaptcha by Google

reCAPTCHA is a CAPTCHA-like system designed to establish that a computer user is human (normally in order to protect websites from bots) and, at the same time, assist in the digitization of books. he main purpose of a CAPTCHA system is to prevent automated access to a system by computer programs or "bots".



Pages
Shopping Cart
Contact Us
Shop
My Account

User Section
Login
Register
Terms And Conditions

Contact Us at
IWP Project
Queenie Das
dasqueenie@gmail.com
Priyansh Jain
priyansh.jain0246@gmail.com

Go to Contact us page

Social Media

ii) Font Awesome

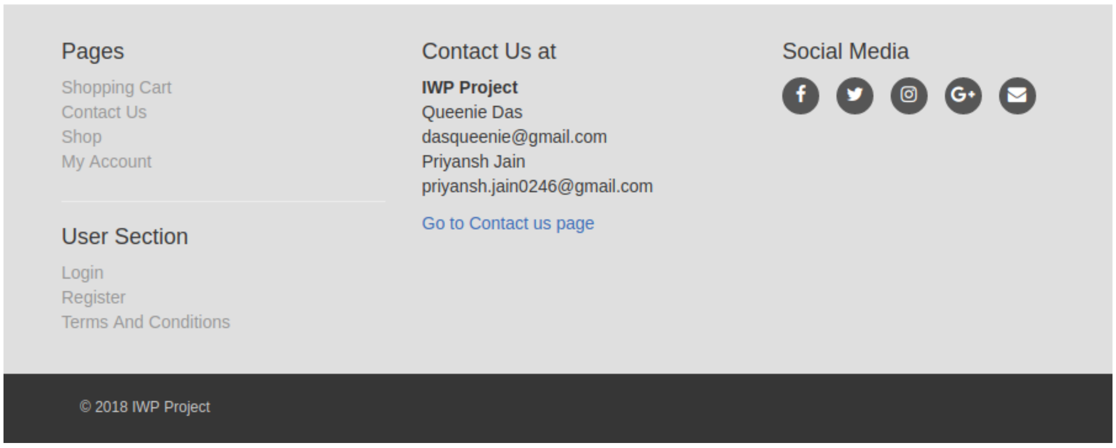
Font Awesome is a font and icon toolkit based on CSS and LESS.

Once you hook your site up to Font Awesome, you can access every icon via a simple HTML tag. For example, to get this cool icon:

…all I need to do is use a piece of code like this:

```
<i class="fa fa-car"></i>
```

Font Awesome will take it and convert it to a live icon. And you can really customize these icons, too. You can change the sizes, colors and shades — basically, you can do anything that's achievable through CSS. As far as you're aware, these icons behave just like any other font.


Pages
Shopping Cart
Contact Us
Shop
My Account

User Section
Login
Register
Terms And Conditions

Contact Us at
IWP Project
Queenie Das
dasqueenie@gmail.com
Priyansh Jain
priyansh.jain0246@gmail.com

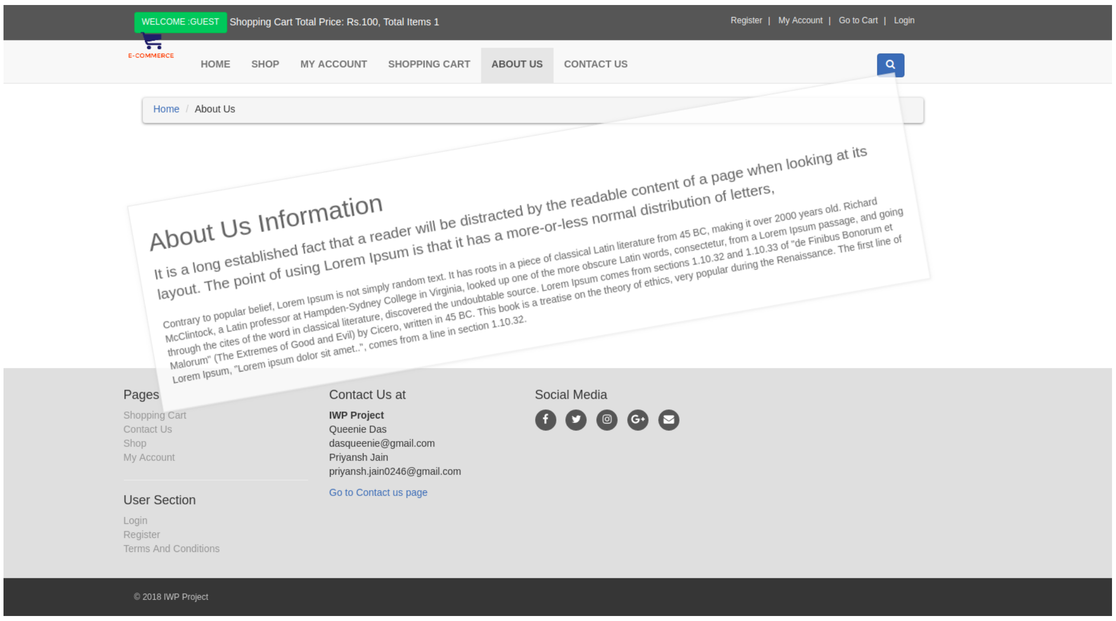
Go to Contact us page

Social Media

© 2018 IWP Project

iii) Animate.css

Animate.css is a bunch of fun, cross-browser animations for us to use in our projects. It is great for emphasis, home pages, sliders, and general just-add-water-animations. In order to use the animations, add the class animated to the element we want to animate. We may also want to include the class infinite for an infinite loop. The name of the class is the animation that we wish to implement for example, *bounce, flash, pulse, rubberBand, bounceIn, fadeout, zoomIn* etc.



Shopping Cart Total Price: Rs.100, Total Items 1

Register | My Account | Go to Cart | Login

Home / About Us

About Us Information

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters,

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

Pages
Shopping Cart
Contact Us
Shop
My Account

User Section
Login
Register
Terms And Conditions

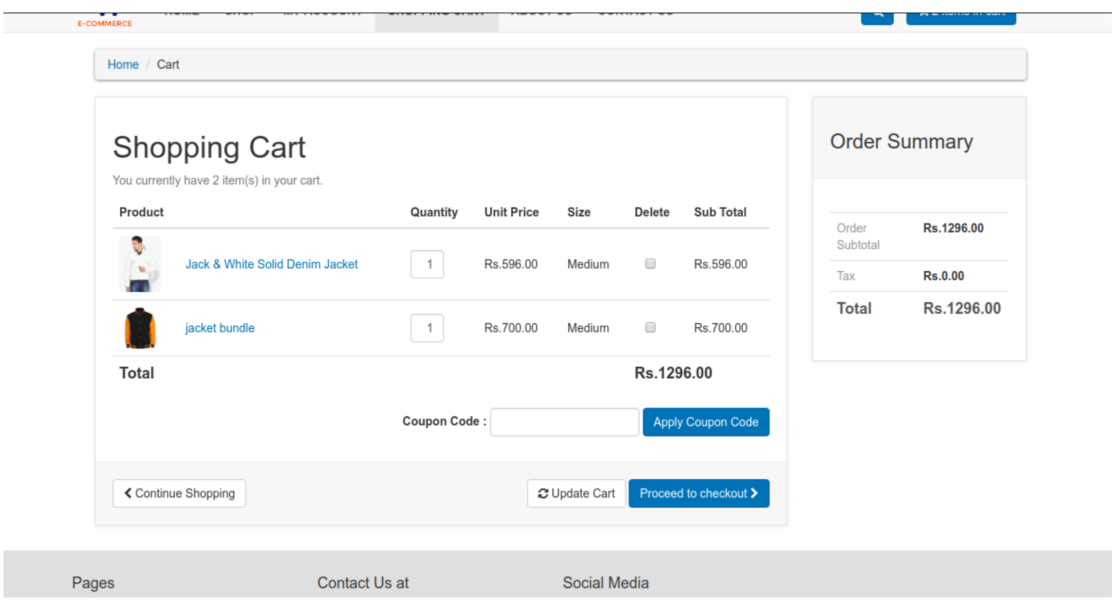
Contact Us at
IWP Project
Queenie Das
dasqueenie@gmail.com
Priyansh Jain
priyansh.jain0246@gmail.com

Go to Contact us page

Social Media

© 2018 IWP Project

SCREENSHOTS



Home / Cart

Shopping Cart

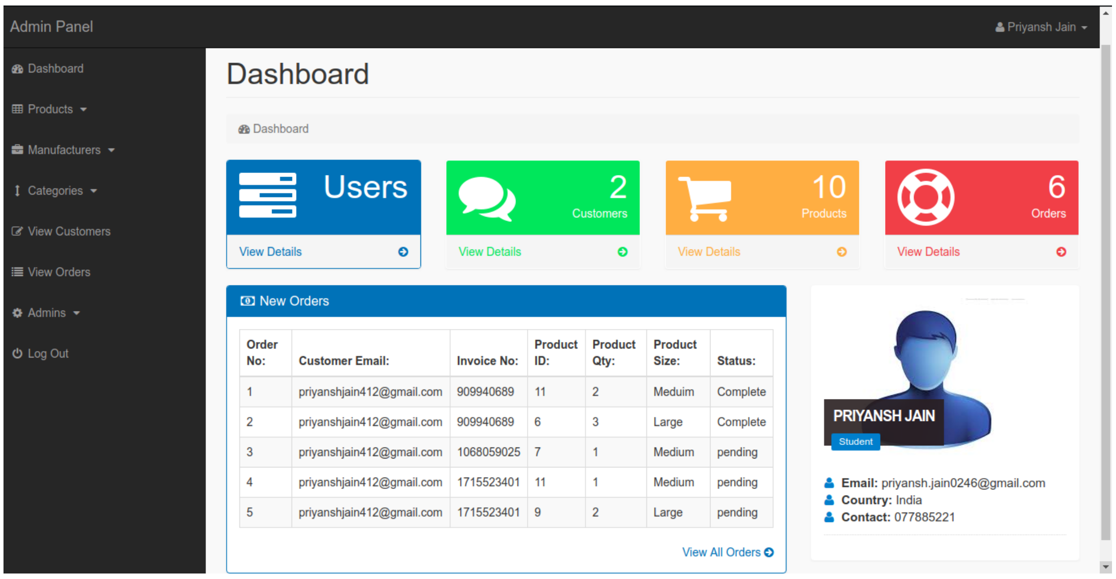
You currently have 2 item(s) in your cart.

Product		Quantity	Unit Price	Size	Delete	Sub Total
	Jack & White Solid Denim Jacket	1	Rs.596.00	Medium	☐	Rs.596.00
	jacket bundle	1	Rs.700.00	Medium	☐	Rs.700.00
Total						Rs.1296.00

Coupon Code :

Order Summary

Order Subtotal	Rs.1296.00
Tax	Rs.0.00
Total	Rs.1296.00



Dashboard

New Orders

Order No:	Customer Email:	Invoice No:	Product ID:	Product Qty:	Product Size:	Status:
1	priyanshjain412@gmail.com	909940689	11	2	Meduim	Complete
2	priyanshjain412@gmail.com	909940689	6	3	Large	Complete
3	priyanshjain412@gmail.com	1068059025	7	1	Medium	pending
4	priyanshjain412@gmail.com	1715523401	11	1	Medium	pending
5	priyanshjain412@gmail.com	1715523401	9	2	Large	pending

PRIYANSH JAIN
Student

Email: priyansh.jain0246@gmail.com
Country: India
Contact: 077885221

CONCLUSION

Thus we have developed a complete end-to-end E-Commerce website that features fixes to many problems that are currently faced, by making efficient use of recent technologies. The website is responsive, and efficient, with zero-downtime measures. The authentication system is done with respect to customers, sellers and admins.