

# Mobile User Experience 6MMCS001W

## Coursework 2 Report - Mobile App Implementation

Author:  
Student ID:

Group Number: 5

Links to WebSite: <https://icraft.cf>  
Youtube Demo: [https://youtu.be/yS\\_15rDQeal](https://youtu.be/yS_15rDQeal)

## Content

Content.....	2
2. Design and implementation of mobile web app features.....	3
2.1 Eye catching introduction.....	3
2.2 Incorporation of a short video .....	4
2.3 POI.....	5
2.4 Integration of google maps .....	7
2.5 List creation & sorting and filtering widgets .....	9
2.6 Booking form and Order .....	10
2.7 Favourites list .....	15
2.8 Gamification (might need to be combined with 2.9) .....	17
2.9 System feedback .....	22
2.10 Scores & points gained (might need to be combined with 2.7).....	25
2.11 Comments.....	25
2.12 Rating.....	29
2.13 AR.....	33
2.14 Differentiation between iPhone and iPad implementation .....	34
References.....	36

## 2. Design and implementation of mobile web app features

In this section below, the proposed design and implementation of the main features of the mobile website user interface is provided and explained.

### 2.1 Eye catching introduction

In this application, the introductory information is used and integrated right before the user accesses the homepage for the first time. This is simply 3 popups which gives the user a basic idea about the main process of the mobile site. When the application loads or the first time, the introductory information is displayed for the user to get a rough idea about the application usage and instructions. The user is given the option to either view the introductory information or to skip the section to directly interact with the system. The introductory information is only shown once when the user opens the application for the first time. When the user visits the application for the next time the application keeps track of the data whether the user was presented with the introduction popup.

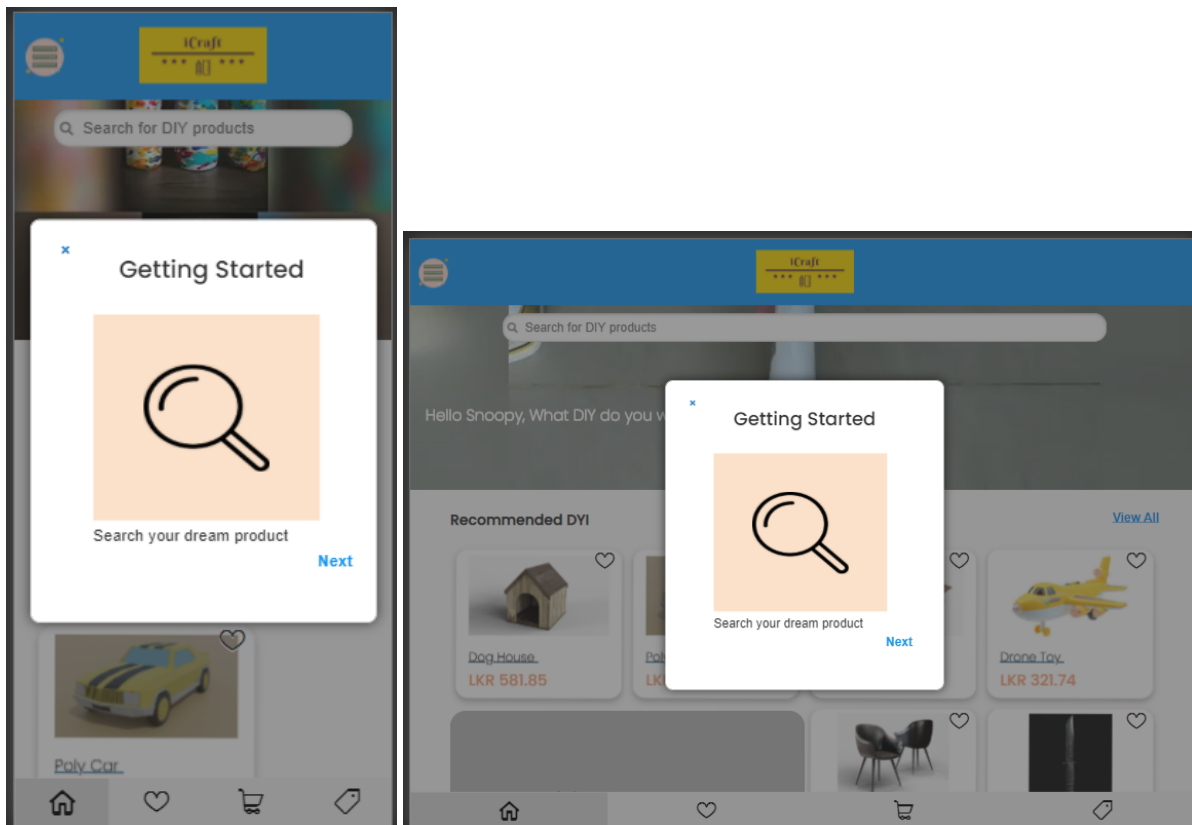
The mark-up which was used for the introductory section is as below.

```
<!--intro_popup1-->
<div data-role="popup" id="popupIntroductory" data-theme="a" class="ui-
corner-all" style="height: 400px">
  <form>
    <div style="padding: 10px 30px; height: 450px; width: 300px;">
      <a data-ajax="false" href="#home" data-transition="pop"
        style="position: absolute; margin: 10px 10px 0 0; text-
decoration: none">&times</a>
      <div class="header-title" style="margin-bottom: 17px; overflow:
auto">
        <h3>Getting Started</h3>
      </div>
      <div style="background-color: #f0862640; height: 205px; width:
225px; margin-left: 32px;">
        
        <span id="introText">Search your dream product</span>
      <br/>
      </div>
      <br/>
      <a id="nxtBtn" onclick="next(this)" data-transition="pop"
        style="float: right; margin: 10px 10px 0 0; text-decoration:
none">Next</a>
    </div>
  </form>
</div>
```

```

<script>
  let page = 1;
  const next = (element) => {
    if (page === 1) {
      $('#imageIntro').attr('src',
"../../common/assets/images/introductory/intro_info2.png");
      page++;
    } else if (page === 2) {
      $('#imageIntro').attr('src',
"../../common/assets/images/introductory/intro_info3.png");
      //element.innerHTML = "Finish";
      $('#nxtBtn').replaceWith('<a href="#home" data-
transition="pop" style="float: right; margin: 10px 10px 0 0; text-
decoration: none">Close</a>');
      //element.style.visibility = "hidden";
    }
  }
}
</script>

```



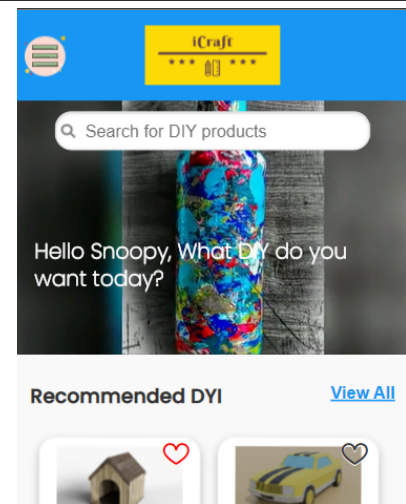
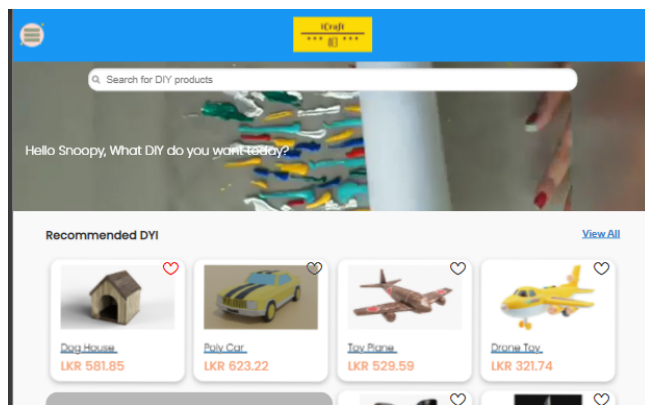
## 2.2 Incorporation of a short video

A short introductory video of the products that are available in the application and a basic shopping experience of the site which is delivered to the user via a looped video and cannot be scrubbed through and it automatically loads when the user comes to the home page. This video consists of attractive animations and transitions which easily distract the user's attention to the video from the promotional content which will be featured as

the main highlight of the page. This video container is of flexible size which will resize according to the device which the site will be used on.

The initial look of the iPhone portrait and landscape is adjusted according to the screen width and the application constantly determines the device type in order to redirect the user to the appropriate site to give the optimum experience. The application does not have to check the orientations all the time because the application's CSS is written in order to fit all the orientations reducing the extra work.

```
<video style="object-fit: cover;" width="100%" height="240" autoplay>
  <source src="../../../videos/intro.mp4" type="video/mp4">
  Your browser does not support the video tag.
</video>
```



## 2.3 POI

The POI part of the application draws attention from the user to display products and promotions etc. Therefore, the products are given a spacing between them for the user to focus on the products more easily and divided with a background colour for the product listing and slash-it listings.

Although the content is the same, the placing of the items in iPhone and iPad are different. iPhone layout has optimised for the user to see the most important items at a single place of view while the iPad view has been optimised for the user to get a clear view of many products at the same time since the screen space is much bigger than that of the iPhone.

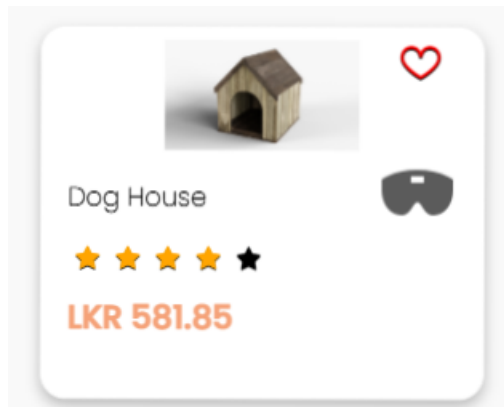
The product card contains a favourite icon which is indicated with a heart icon on the top right corner for the user add the products to wish List. And also, that icon can be used to

remove a product from the wish list as well maintaining the consistency between the user's actions enabling the user to get an idea how to revert his/her actions once an action is done. In the top centre of the product card, a high-resolution image scaled to fit the card is displayed in order to provide the user with an iconic preview of the product when browsing through the product lists. The photo is relatively larger than the other elements since the importance of the real preview is needed in a category like DIY items. The product title is displayed after the image of the product followed by the price of the product with a different colour than that of the usual colours to highlight the importance of the price of the product. We have also added paddings as needed to isolate the products from the other products.

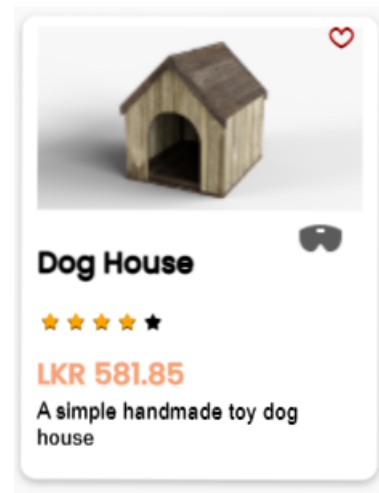
Clicking a product card will take the user in to the particular product which the user clicked on. The product data is dynamically fetched and loaded from a pre-saved JSON file which mimics the behaviour of a database. All products are formatted, and the descriptions are set with a common set of styles maintaining a bit of a consistency.

The code for the generation of the POI cards is as below where the products are loaded dynamically.

```
<div class="ui-block-<?php echo $column ?> card ui-corner-all custom-
corners">
  <span class="flaticon-122-heart fav-icon-card"
onclick="favourite(this)" id="<?php echo $key ?>"></span>
  <a data-ajax="false" href="<?php echo $page ?? 'product.php'
??id=<?php echo $key ?>">
    <div class="ui-bar ui-bar-a ui-card">
      </br>
      <p class="card-title"><?php echo $name ?> </p> </br> <span
        class="card-price">LKR <?php echo $price ?? ' '
?></span>
    </div>
  </a>
</div>
```



## 2.4 Integration google maps



of

is

Google map is integrated in contact us page to show multiple store locations with markers. Here, we have added 3 locations separately. The user is able to view all the shop branch locations by expanding the specific collapsible button and view the google map marker to the relevant shop branch and easily identify the location. This is integrated in both the iPhone and iPad views in the same manner regardless the orientation despite the fact that the collapsible component is widened as per the screen size of the device.

```
<script>
function initialize() {
    var latlng = new google.maps.LatLng(6.866673, 79.89347);
    var latlng2 = new google.maps.LatLng(7.487436, 80.365025);
    var latlng3 = new google.maps.LatLng(7.268108, 80.598766);

    var nugegoda = {
        zoom: 15,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    var kurunegala = {
        zoom: 15,
        center: latlng2,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    var kandy = {
        zoom: 15,
        center: latlng3,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    var map = new google.maps.Map(document.getElementById("map"),
nugegoda);

    var map2 = new google.maps.Map(document.getElementById("map-2"),
kurunegala);
```

```

    var map3 = new google.maps.Map(document.getElementById("map-3"),
kandy);

    var myMarker = new google.maps.Marker(
    {
        position: latlng,
        map: map,
        title: "Nugegoda"
    });

    var myMarker2 = new google.maps.Marker(
    {
        position: latlng2,
        map: map2,
        title: "Kurunegala"
    });

    var myMarker3 = new google.maps.Marker(
    {
        position: latlng3,
        map: map3,
        title: "Kandy"
    });
}

google.maps.event.addDomListener(window, 'load', initialize);
</script>

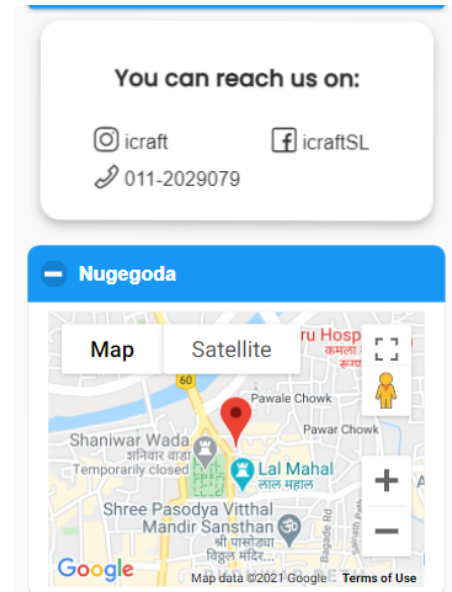
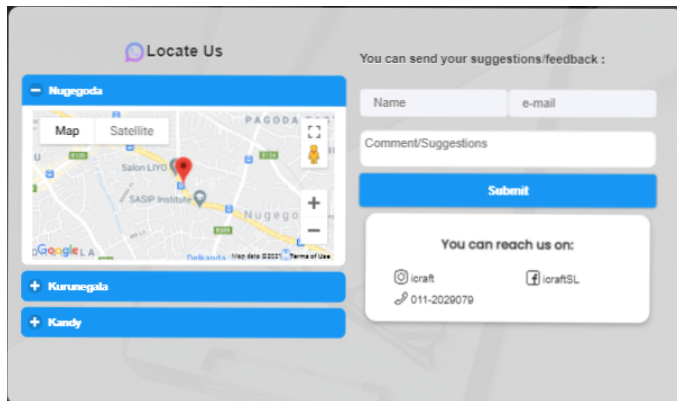
```

```

<div onload="initialize()" style="margin-bottom: 70px">
  <div data-role="collapsible" data-collapsed="false">
    <h4>Nugegoda</h4>
    <div>
      <div id="map" style="width:100%; height:220px"></div>
    </div>
  </div>
  <div data-role="collapsible">
    <h4>Kurunegala</h4>
    <div>
      <div id="map-2" style="width:100%; height:220px"></div>
    </div>
  </div>
  <div data-role="collapsible">
    <h4>Kandy</h4>
    <div>
      <div id="map-3" style="width:100%; height:220px"></div>
    </div>
  </div>
</div>

```



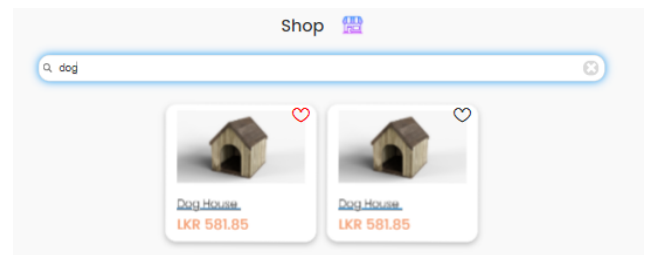
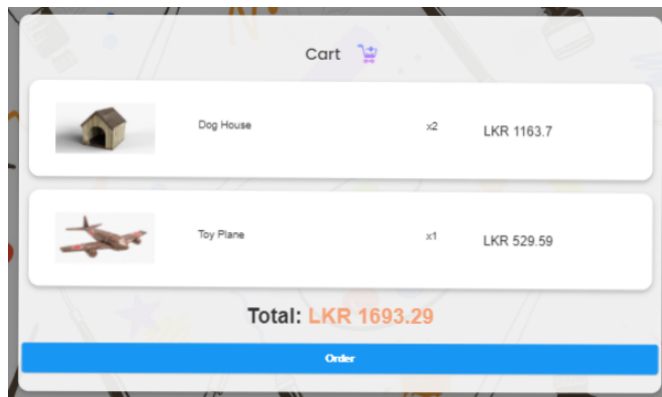


## 2.5 List creation & sorting and filtering widgets

There are features in the mobile web site which allows the user to sort and filter the lists of products which has been created accordingly to maximize the user friendliness and ease of access. In the pages which the list of products is shown i.e. home page, shop page, there is a search bar where the user can type a product name or a basic product keyword, then the list of products get sorted accordingly and provide the users only the most suitable product cards.

The alignment of the list of products in both iPhone and iPad are more likely the same but the number of columns in the iPad grid view is more than the iPhone grid view to maximize the usage of space, but in the pages where the vertical list view has been used i.e., Cart page and Order History page, there is no difference between the iPad view and iPhone view.

```
<div class="ui-grid-d search" data-filter="true" data-filter-
placeholder="Search for DIY products"
style="width: fit-content; margin: 200px auto 0 auto">
```



## 2.6 Booking form and Order

The booking form is designed such that a customer could book a private class with a DIY creator where the booking form will take all the needed values as an input for the form. A user is able to navigate to the booking form page from the side menu panel. The booking system just the way it is designed in the interactive prototype and following the rules of the style guide that was submitted as a part A of this coursework.

The payment for a booking is processed through Stripe payment gateway where a slot can be chosen with the relevant data entered to the forms. The stripe payment will redirect to its own payment UI by using the data we supplied and will redirect into the site once a payment is successful or failed to ensure the security. The user is given with feedbacks about the payment statuses.

Firstly, the user is provided with a form which the details can be inserted according to user's preference. There are input fields, and these contain validation as needed. The booking form is designed according to the style guide and the prototype which was designed previously and has a 'Place Booking' button. By clicking on that button, the user is directed to the payment page.

```

window.addEventListener("DOMContentLoaded", function () {
  var form = document.getElementById("booking-form");

  form.addEventListener("submit", function (ev) {
    ev.preventDefault();
    // Create an instance of the Stripe object with API key
    const stripe = Stripe("<?php echo $GLOBALS['stripe_key'] ?>");

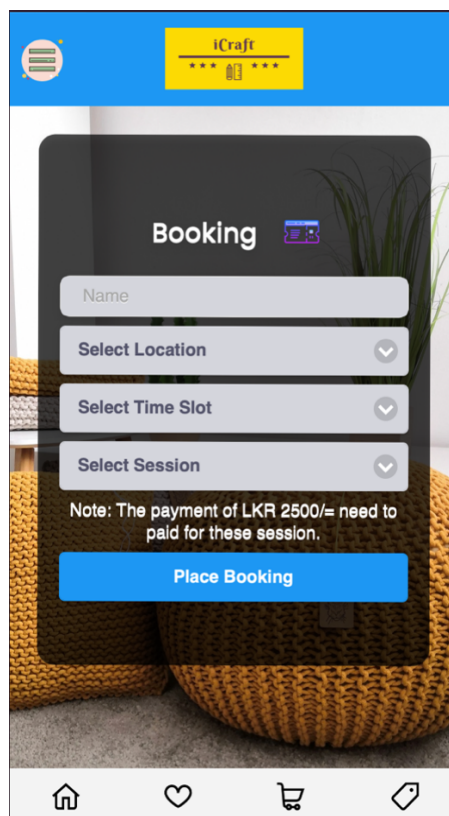
    let location = $('#location').find(":selected").text();
    let timeSlot = $('#timeSlot').find(":selected").text();
    let session = $('#session').find(":selected").text();
  });
});

```

```

fetch("../../common/functions/checkout.php?cat=booking&origin=<?php
echo $origin ?? 'iPhone' ?>", {
  method: "POST",
  body: JSON.stringify({name: session + " " + timeSlot + " " +
location}),
})
  .then(function (response) {
    return response.json();
  })
  .then(function (session) {
    return stripe.redirectToCheckout({sessionId: session.id});
  })
  .then(function (result) {
    // If redirectToCheckout fails due to a browser or network
    // error, you should display the localized error message to
your
    // customer using error.message.
    if (result.error) {
      alert(result.error.message);
    }
  })
  .catch(function (error) {
    console.error("Error:", error);
  });
});
})

```



After Completion stripe sends a notification to the backend.

```
<?php
session_start();
```

```

$orders = json_decode(file_get_contents("../orders.json"), true);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = json_decode(file_get_contents('php://input'), true);
    $order_id = $data['data']['object']['id'];
    foreach ($orders as $key => $order) {
        if ($key == $order_id) {
            $orders['status'][$key] = "PENDING";
            $json = json_encode($orders);
            file_put_contents("../orders.json", $json);
            echo "Success";
            return;
        }
    }
}
echo "Error";

```

The online payment was implemented as such the relevant product ids were sent to the backend and then the backend processes the data and passes the processed data to Stripe and redirects. Then, the client is allowed to perform the transaction with the Stripe service without any connection to the client app. After the completing or aborting the transaction, the user will be redirected to the application along with the process success/failure message. Then the data will be added to as such the order history is updated.

```

<?php
session_start();
require_once('../stripe-php/init.php');
include "../common/vars.php";

use Stripe\Checkout\Session;
use Stripe\Stripe;

header('Content-Type: application/json');

Stripe::setApiKey('sk_test_51Hs4vICZjSyoKagrFIBYkgON3TKZ9TST4xp8sJ5t99IQpr0EYUOYVFj1u8hIxx9GVr8BlhYKsrq20UfWWjVtMs4900xuY2ithd');

// post data
$data = json_decode(file_get_contents('php://input'), true);

// get data from db
// read file
$json = file_get_contents('../db.json');
$orders = json_decode(file_get_contents("../orders.json"), true);

// Converts it into a PHP object
$products = json_decode($json);
$line_items = array();

if (!isset($_GET['cat'])) {
    foreach ($data as $product_id) {

```

```

        $product = $products[$product_id['id']];
        $price_data = [
            'price_data' => [
                'currency' => 'lkr',
                'unit_amount' => floatval($product->price) * 100,
                'product_data' => [
                    'name' => $product->name,
                    'images' => [$product->image],
                ],
            ],
            'quantity' => $product_id['qty'],
        ];
        array_push($line_items, $price_data);
    }
} else if ($_GET['cat'] == 'booking') {
    $price_data = [
        'price_data' => [
            'currency' => 'lkr',
            'unit_amount' => 250000,
            'product_data' => [
                'name' => $data['name'],
                'images' =>
                ['https://c1.sfdcstatic.com/content/dam/blogs/ca/Blog%20Posts/shake-up-
                sales-meeting-og.jpg'],
            ],
            'quantity' => 1,
        ];
        array_push($line_items, $price_data);
    }

    $origin = $_GET['origin'];
    $checkout_session = Session::create(
        [
            // 'customer_email' => 'customer@example.com',
            'payment_method_types' => ['card'],
            'shipping_address_collection' => [
                'allowed_countries' => ['LK'],
            ],
            'line_items' => $line_items,
            'mode' => 'payment',
            'success_url' => $GLOBALS['domain'] . '/' . $origin .
            '/pages/orderHistory.php?status=success',
            'cancel_url' => $GLOBALS['domain'] . '/' . $origin .
            '/pages/orderHistory.php?status=failed',
        ]
    );
    $session_id = $checkout_session->id;
    $orders[$session_id] = $data;
    file_put_contents("../orders.json", json_encode($orders));
    echo json_encode(['id' => $session_id]);
}

```

Then as the user completes the payment, user gets a confirmation alert in the end. These booking can be later viewed by the user by checking the Order History page as all the orders and bookings are saved in the order history.

```

<?php
session_start();

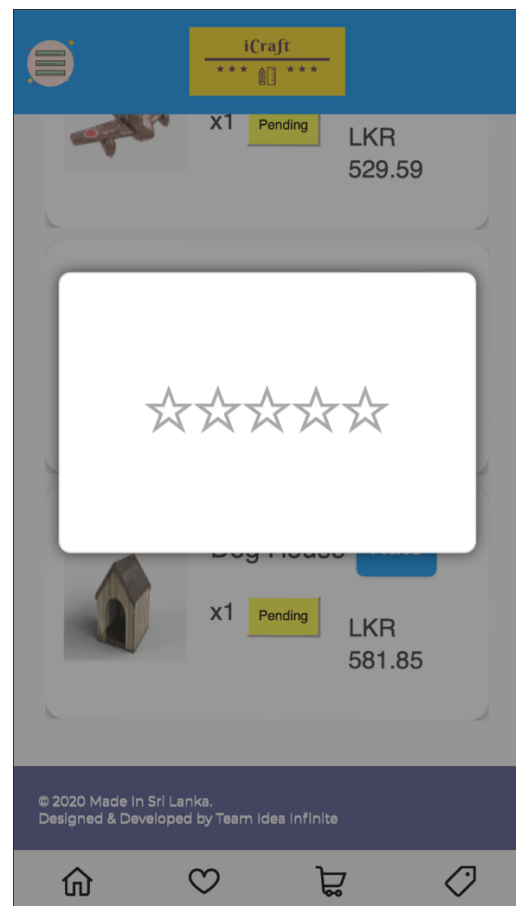
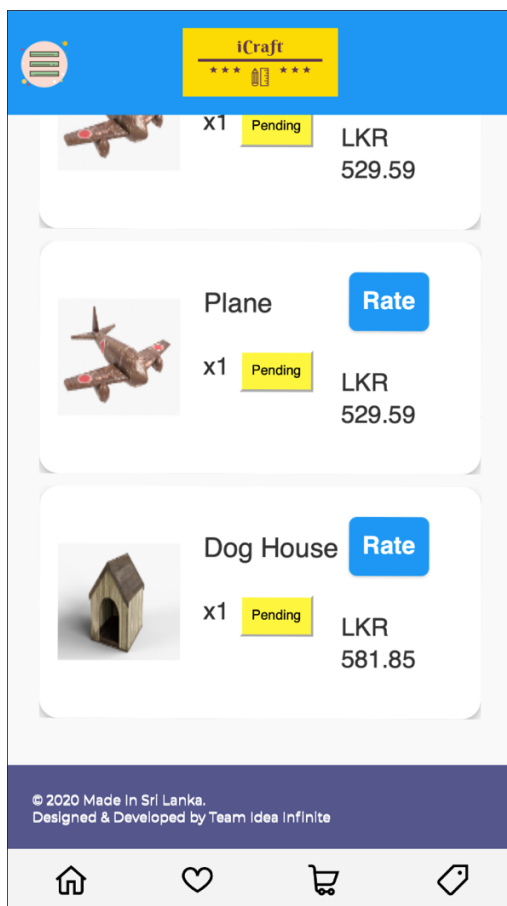
```

```

$orders = json_decode(file_get_contents("../orders.json"), true);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = json_decode(file_get_contents('php://input'), true);
    $order_id = $data['data']['object']['id'];
    foreach ($orders as $key => $order) {
        if ($key == $order_id) {
            $orders['status'][$key] = "PENDING";
            $json = json_encode($orders);
            file_put_contents("../orders.json", $json);
            echo "Success";
            return;
        }
    }
}
echo "Error";

```



The major difference in the favourites list on device orientation change is the change in the number of card columns. Due to the increase of screen resolution, the iPad has more space to accommodate content and therefore, two columns of cards have been implemented to allow the user access to more items without the need for scrolling.

## 2.7 Favourites list

The user is allowed to 'like' the products in the Home page, Shop page by simply clicking on the heart icon which is on the top right corner of the POI card. As the user clicks on that, the product gets added to the list of favourite items in the Wish List page. Afterwards the user can view them whenever needed and purchase them, view the full details of the product.

As there are items added to Wish List, the user is able to send the item list to an email address that can be entered by the user himself. Here, the system checks whether it is a valid email address or not. This way the user can keep track of his/her wish list and view them later. For this purpose, the interface shows a popup with a text field to enter the user's email address and by clicking on the 'send email' button, the email gets sent to that relevant email account.

The orientation of the wish list page and the Email the Wish List popup are more alike in both the iPhone and iPad view, yet the sizes of the UI components are slightly different according to the screen size of the devices. Since the iPad device's screen is wider than the iPhone's screen, the components show a more widened view in the iPad. Yet the functionality and the styling of the components are same and has followed the style guide accordingly.

```
<div role="main" class="ui-content">
  <div class="header-title" style="margin-bottom: 10px;">
    <h3>WishList</h3>
    
  </div>
  <!-- POI Card-->
  <div id="content" class="ui-grid-a search" data-filter="true"
    data-filter-placeholder="Search for your wish list"
    style="margin-top: 10px;">

    <div id="items"></div>
    <a data-rel="popup" data-position-to="window"
      data-transition="pop" href="#wishListEmail">
      <button>Email My List</button>
    </a></div>
  </div>
  <?php include '../popups/wishListEmail.php' ?>
  <?php include '../parts/bottomNavbar.php' ?>
  <?php include '../parts/footer.php' ?>
</div>
</body>
<script>
  const update = async () => {
```

```

    let data;
    if ($.cookie('wishList') != null) {
        data = JSON.parse($.cookie('wishList'));
        for (let j = 0; j < data.length; j++) {
            await addCard(data[j]);
        }
    }

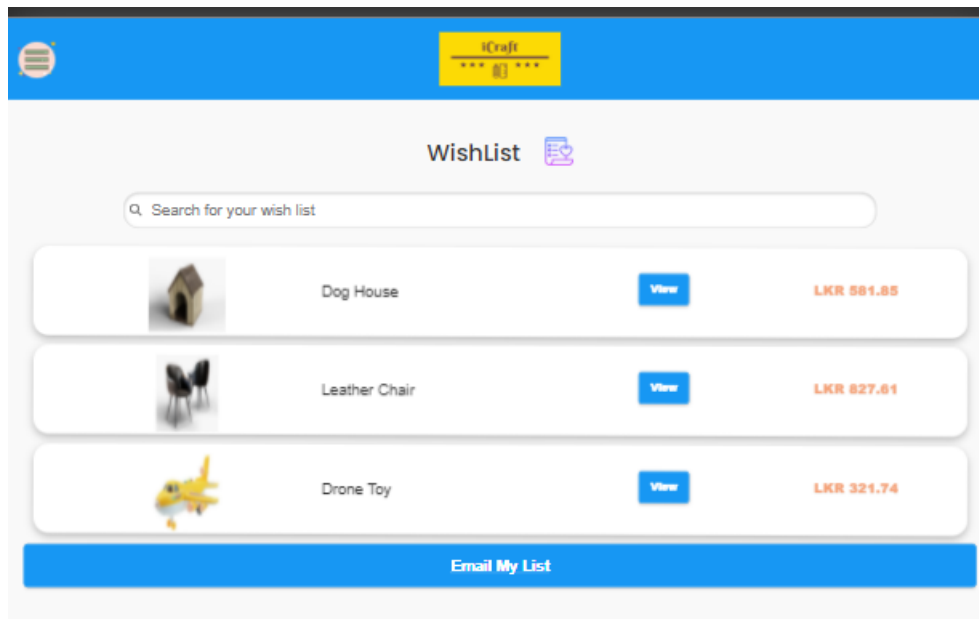
    const addCard = async (id) =>
        $.getJSON("../common/functions/getProduct.php?id=" + id,
function (data, status) {

            $('#items').append(
'<div class="ui-grid-c back-box" style="padding: 20px; margin-bottom:
10px">\n' +
                '
                    <div class="ui-block-a " style="width:
30%; height: 50px">\n' +
                        '
                            \n' +
                                        '\n' +
                                            '
                                                </div>\n' +
                                                    '
                                                        <div class="ui-block-b" style="width:
35%;">\n' +
                                                            '
                                                                <p>' + data["name"] + '</p>\n' +
                                                                    </div>\n' +
                                                                        '
                                                                            <div class="ui-block-c" style="width:
20%; font-size: 12px; ">\n' +
                                                                                '
                                                                                    <a data-ajax="false"
href="product.php?id=' + id + '" \n' +
                                                                                        '
                                                                                            class="ui-btn ui-corner-all ui-
shadow ui-btn-inline ui-btn-a"\n' +
                                                                                                '
                                                                                                    data-
transition="pop">View</a>\n' +
                                                                                                        '
                                                                                                            </div>\n' +
                                                                                                                '
                                                                                                                    <div class="ui-block-d" style="width:
15%; height: 50px">\n' +
                                                                                                                        '
                                                                                                                            <p style=" font-size: 14px; color:
#f5a278; font-weight: 800;">LKR ' + data["price"] + '</p>\n' +
                                                                                                                                </div>\n' +
                                                                                                                                    '
                                                                                                                                        </div>\n');
                });

            setTimeout(function () {
                update();
            }, 200);
        }
    </script>

```

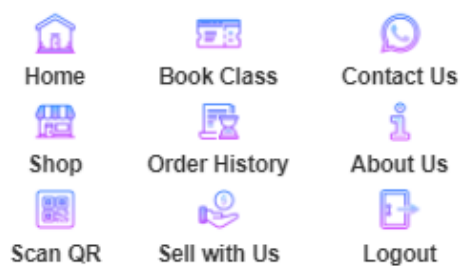
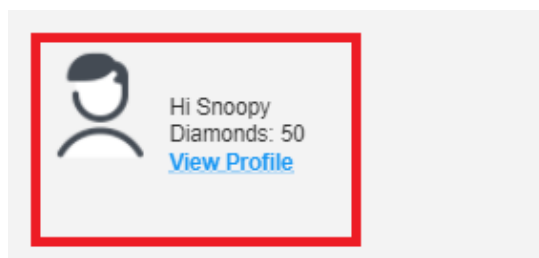




2.8

### Gamification (might need to be combined with 2.9)

The gamification throughout the application is highly related with the user's contribution towards the promotion and the purchases in the application. Therefore, the implementation of the gamification is designed as such that it is motivating the customer to make purchases in the application which is the core ideology behind the gamification process.



Here the diamonds earned by a customer is always displayed in the menu panel where it will be highlighted to the user to motivate the user from increasing the diamonds as a self-rewarding mechanism.

There are few ways of increasing the number of diamonds. One of the ways is scanning a QR code which is pated on to a delivered package after placing an order. The next method of gaining diamonds is purchasing of items with a value more than 1000/= LKR.

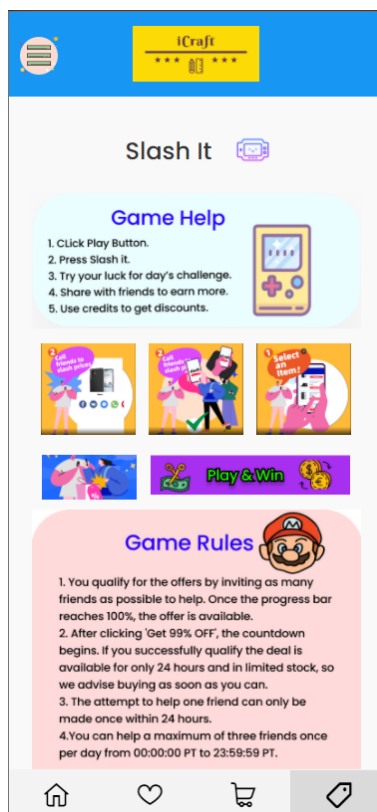
```
<!-- panel content goes here -->
<div style="height: 150px; background-color:#f3f3f3; padding: 20px;
display: flex">
  
  <div class="side-menu-info">
```

```

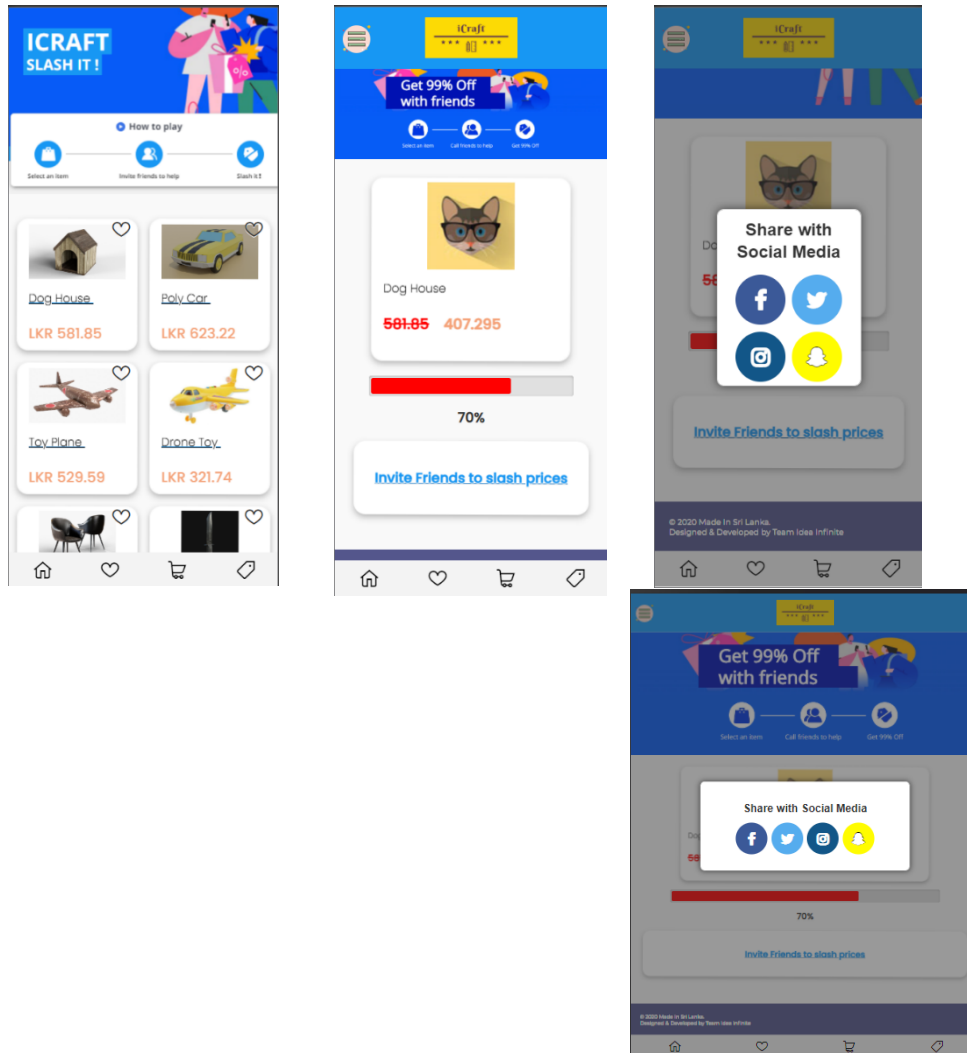
<p>Hi <span id="userName"></span></p>
<p>Diamonds: <span id="diamondCount"></span></p>
<p><a data-ajax="false" href="../pages/profile.php">View
Profile</a></p>
</div>
</div>
<script>
function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return
c.substring(nameEQ.length,c.length);
    }
    return null;
}
function displayDiamondCount() {
    // handle on success condition with the decoded message
    var diamondCount=readCookie('diamondCount');
    $("#diamondCount").html(diamondCount);
}
</script>

```

Also, the diamonds are added when a user frequently visits the site in a continuous number of days which ensures the user is faced with the promotional content each time the user visits the application.



Another gamification technique that we have used is the SlashIt feature where a user can apply for discounts by sharing the application to customer's friends thereby increasing the user base of the application which can drastically increase revenue for the business by promoting the application while giving a benefit for the user.



Much more like the shopping experience the customers will be able to shop products and apply for slashing the prices.

The Slashing process will be done by initially choosing a product. Then, the user will be directed to the SlashIt Product Page where the product will be displayed along with the progress of the SlashIt process with a colourful progress bar that motivates the user to reach the maximum out of the progress bar to achieve a greater discount.

```

<div data-role="page" id="slashit">
  <?php include '../parts/header.php' ?>
  <?php
    $data = $GLOBALS['db'];
    $json = json_decode($data, true);
    $id = $_GET['id'];
  ?>
  <div role="main" class="ui-content">
    <center>
      
    </center>
    <div class="ui-bar ui-bar-a ui-card" style="height: unset; max-
width: 70%; margin: auto;">

      <div class="ui-grid-a">
        <div class="ui-block-a" style="width: 80%">
          <p style="font-weight: 200; font-family:
'Poppins';"><?php echo $json[$id]['name'] ?? ' ' ?> </p>
          <div style="display: flex;">
            <p class="card-price"
              style="position: relative; color: red; text-
decoration: line-through;"><?php echo $json[$id]['price'] ?? ' ' ?></p>
            <p class="card-price"
              style="margin-left: 20px; position:
relative;"><?php echo $json[$id]['price'] * 0.7 ?? ' ' ?></p>
          </div>
        </div>
      </div>
      <div class="progress-wrapper">
        <div class="progress-bar">
          <span class="progress-bar-fill" style="width: 70%;"></span>
        </div>
        <center><h3>70%</h3></center>
      </div>
      <div class="back-box" style="margin-bottom: 20px">
        <center>
          <a href="#popupBasic" data-rel="popup" data-position-
to="window"
            data-transition="pop"><h3
              style="color: #1797F3;">Invite Friends
              to slash prices</h3></a>
        </center>
      </div>
      <div style="width: 400px; padding: 30px; text-align: center" data-
role="popup" id="popupBasic">
        <h2>Share with Social Media</h2>
        <div class="share-social"><a style="color: white;" href="#"
class="fa fa-facebook"></a>
          <a style="color: white;" href="#" class="fa fa-
twitter"></a>
          <a style="color: white;" href="#" class="fa fa-
instagram"></a>
          <a style="color: white;" href="#" class="fa fa-snapchat-
ghost"></a></div>
      </div>
    </div>
  </div>
  <script>

```

```

        $(document).on("popupafteropen", function() {
            $('#popupBasic').popup('reposition', 'positionTo: window');
        });
    </script>
    <?php include '../parts/bottomNavbar.php' ?>
    <?php include '../parts/footer.php' ?>
</div>

```

The views of the pages have been aligned so that the optimum spaces will be used in iPhone while the application will take use of the large screen in iPad.

```

<div data-role="page" id="slashit">
    <?php include '../parts/header.php' ?>
    <?php
        $data = $GLOBALS['db'];
        $json = json_decode($data, true);
        $id = $_GET['id'];
    ?>
    <div role="main" class="ui-content">
        <center>
            
            <div class="ui-bar ui-bar-a ui-card" style="height: unset; max-
width: 70%; margin: auto;">

                <div class="ui-grid-a">
                    <div class="ui-block-a" style="width: 80%">
                        <p style="font-weight: 200; font-family:
'Poppins';"><?php echo $json[$id]['name'] ?? ' ' ?> </p>
                        <div style="display: flex;">
                            <p class="card-price"
                                style="position: relative; color: red; text-
decoration: line-through;"><?php echo $json[$id]['price'] ?? ' ' ?></p>
                            <p class="card-price"
                                style="margin-left: 20px; position:
relative"><?php echo $json[$id]['price'] * 0.5 ?? ' ' ?></p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="progress-wrapper">
                <div class="progress-bar">
                    <span class="progress-bar-fill" style="width: 70%;"></span>
                </div>
                <center><h3>70%</h3></center>
            </div>
            <div class="back-box" style="margin-bottom: 20px">
                <center>
                    <a href="#popupBasic" data-rel="popup" data-position-
to="window"
                        data-transition="pop"><h3
                            style="color: #1797F3;">Invite Friends
                                to slash prices</h3></a>
                </center>

```

```

    </div>
    <div style="width: 200px; text-align: center" data-role="popup"
id="popupBasic">
        <h2>Share with Social Media</h2>
        <div class="share-social"><a style="color: white;" href="#"
class="fa fa-facebook"></a>
            <a style="color: white;" href="#" class="fa fa-
twitter"></a>
            <a style="color: white;" href="#" class="fa fa-
instagram"></a>
            <a style="color: white;" href="#" class="fa fa-snapchat-
ghost"></a></div>
    </div>

```

## 2.9 System feedback

The mobile web app has several methods of giving system feedback to the user to make it more interactive. Whenever the user is supposed to give any sort of an input to the system, such as making a payment, adding to cart, booking a class, when trying to get registered to the web app, the system confirms the user whether the given input is valid or whether the action that the user made has been done correctly. For this purpose, the system mainly uses popup alerts, dialog boxes.

As the whole mobile web application shares a very simple language that any type of a user can understand, for giving system feedback also it is used a very understandable and a simple language. In most occurrences, the feedback is given in a very short sentence by using only the most important keywords which explains what is needed to be understood by the user.

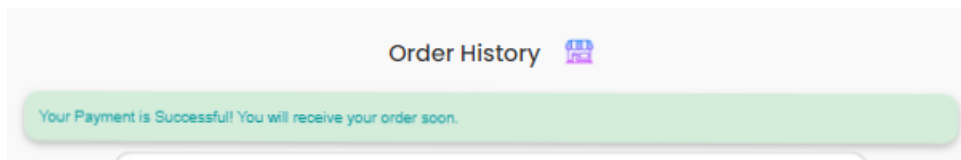
The system feedback is mostly a confirmation message that is provided to the user to inform that the action he/she performed is done successfully or not. To make it more clear to the user, the system uses specific colours and icons. When the user input or action is done successfully, the success message shows in a green alert box or uses a green tick mark therefore, the user is able to understand the meaning of it in one glance even without reading the whole message, and when the action or input has not done successfully, the message is shown in red colour or shown a cross mark in the dialog box.

These system feedback elements are designed and implemented along with the rule of consistency and applying the styles which have mentioned in the style guide. The whole mobile app shares a style of UI components with curved edges and to provide system

feedback also the alert boxes are made with curved edges. Therefore, it adds more consistency and uniformity among the UI elements and components.

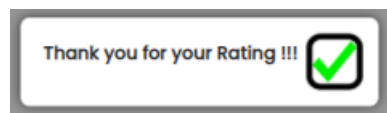
### #Order Status

```
<?php
if (isset($_GET['status']) && $_GET['status'] == 'success') {
    echo '<div class="ui-grid-c back-box" ' .
        'style="padding: 15px; margin-bottom: 10px; background-color: #D4EDDA;
color: #009999">' .
        'Your Payment is Successful! You will receive your order soon.</div>';
    setcookie('cart', null, time() - 3600, '/');
} else if (isset($_GET['status']) && $_GET['status'] == 'failed') {
    echo '<div class="ui-grid-c back-box" ' .
        'style="padding: 15px; margin-bottom: 10px; background-color: #F8D7DA;
color: #721C24">' .
        'Your Payment is not successful! Please check your card detail.</div>';
}
?>
```



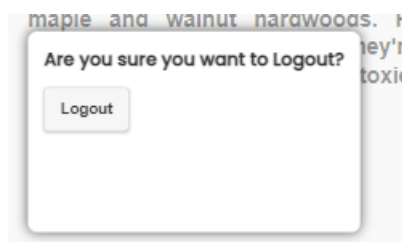
### # Rating

```
<div style="display: flex" data-role="popup" id="rate-feedback">
    <h3>Thank you for your Rating !!! </h3> 
</div>
```



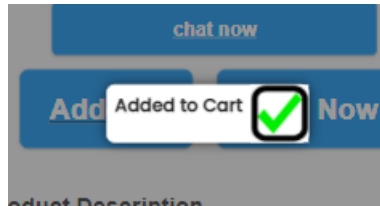
### #logout

```
<div role="main" class="ui-content">
    <h3 class="ui-title">Are you sure you want to Logout?</h3>
    <a data-ajax="false" href="../../../pages/logIn.php" class="ui-btn ui-corner-
all ui-shadow ui-btn-inline ui-btn-b"
        data-transition="flow">Logout</a>
</div>
```



### #add to cart

```
<div style="display: flex" data-role="popup" id="cart-feedback">
  <h3>Added to Cart</h3> 
</div>
```

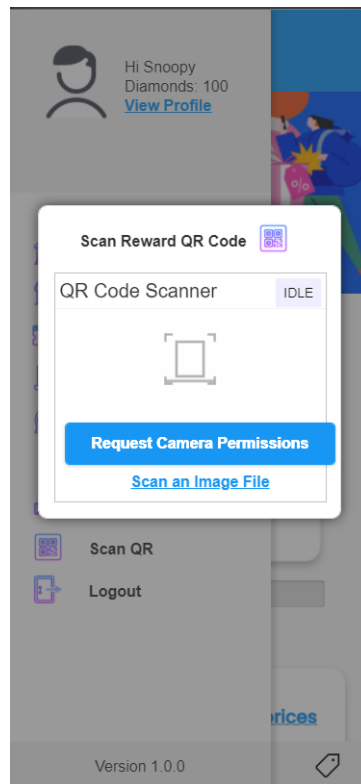
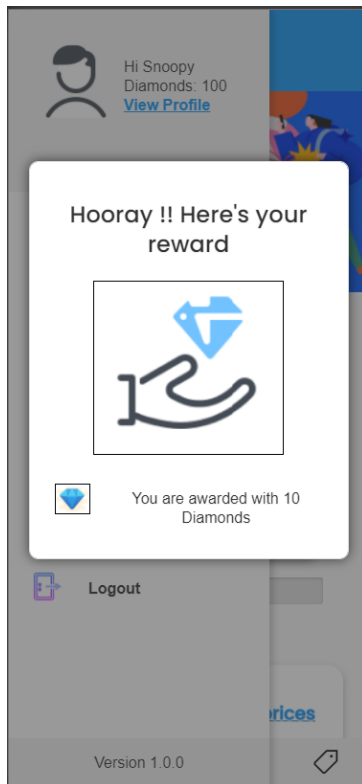


## #Game Reward

```
<div style="max-width: none; min-width: 300px;" data-role="popup" id="qrRewards"
data-theme="a"
  class="ui-content">
  <div class="header-title" style="margin-top: -10px; margin: 10px; display:
flex">
    <h3>You are awarded with</h3>
  </div>
  <div style="display: flex; justify-content: center;">
    
  </div>
  <br>
  <div style="display: inline-flex">
    
    <p>You are awarded with <span id="qrId"></span></p>
  </div>
</div>
```



## 2.10 Scores & points gained (might need to be combined with 2.7)



Here the user is given the chance to earn diamonds by scanning a package once it is delivered to door.

This motivates the user to continue ordering items.

```
<script>
  function onScanSuccess(qrCodeMessage) {
    // handle on success condition with the decoded message
    $("#viewQR").popup("close");
    $("#qrId").html(qrCodeMessage);
    $("#qrRewards").popup("open");
  }

  var html5QrcodeScanner = new Html5QrcodeScanner(
    "reader", {fps: 10, qrbox: 250});
  html5QrcodeScanner.render(onScanSuccess);
</script>
```

## 2.11 Comments

In this mobile web application, the users can leave comments regarding the POIs. In each product page, the user is able to inquire, review and tell whatsoever that he/she has to mention regarding the product. The commenting section can be seen in the very bottom of each product page. This includes a text field in which the user can input a new main comment. Underneath that text field, the user can see the list of comments which has been added to the product page by other users and user can reply to these comments using the 'Reply' button, as user clicks on that, a smaller yet similar to the main input comment text

field appear and using that, the user can input a reply to the main comment. At the same time, there is a 'thumbs-up' toggle button which user can click to mark it as liked.

The appearance of the comment section as per the device orientation changes since the width of the screen changes. The provided comments are shown as a list in both orientations of iPhone and iPad. A comment card contains the profile image of the user, the name of the user, the comment itself, reply and like buttons and the list of replies that the main comment has received if any. All the comment text is shown in black the buttons and the name of the user is shown in a larger font size to add more emphasis towards it.

```
<div class="ui-grid-a">
  <label><b>Comments</b></label>
  <table>
    <tr>
      <td>
        <div style="max-width: 250px; padding-left: 12px;">
          <input type="text" data-clear-btn="true" name="comment"
id="inputComment" value="">
        </div>
      </td>
      <td>
        <button id="postBtn" style="opacity: 1; max-height:40px;
max-width: inherit"
          onclick="getInputValue(this);">Post
        </button>
      </td>
    </tr>
  </table>

  <ul id="ul" style="list-style: none; margin-left: -40px;">
    <li>
      <div class="back-box" style="padding: 5px 10px;margin: 10px
10px 10px 10px;">
        <table>
          <tr>
            <td>
              
            </td>
            <td>
              <div class="product-comment"></div>
              <h3>Goofy</h3>
              <p>What are the colours available?</p>
            </td>
          </tr>
        </table>
      </div>
    </li>
  </ul>
  <div>
    <a style="margin: 10px"
```

```

onclick="openReply()">reply</a><i onclick="myFunction(this)" class="fa fa-
thumbs-up"></i>

        </div>
      </td>
    </tr>
  </tr>

      <td id="repliersPhoto">

      </td>
      <td id="displayReply">

      </td>
    </tr>
    <tr id="replyTextField">
    </tr>
  </table>
</div>

</li>
</ul>
</div>

<script>
  function getInputValue(e) {
    let comment = $('#inputComment').val();
    $("#ul").append('' +
      ' <li>' +
      '<div class="back-box" style="padding: 5px 10px;">' +
      '<table>' +
      '<tr>' +
      '<td>' +
      '' +
      '</td>' +
      '<td>' +
      '<div class="product-comment"></div>' +
      '<h3>Snoopy</h3>' +
      '<p id="snoopyComment">' + comment + '</p>' +
      '<div>' +
      '<a style="margin: 10px" onclick="openReply1()">reply</a>' +
      '<i onclick="myFunction(this)" class="fa fa-thumbs-up"></i>' +
      '</div>' +
      '</div>' +
      '</td>' +
      '</td>' +
      '</tr>' +
      '<tr>' +
      '<td id="repliersPhoto1">' +
      '</td>' +
      '<td id="displayReply1">' +
      '</td>' +
      '</tr>' +
      '<tr id="replyTextField1">' +
      '</tr>' +
      '</div>' +
      '</table>' +
      '</li>'
    )
  }

```

```

;

}

function openReply() {
  console.log('type a reply');
  $("#replyTextField").append('' +
    '<td></td>' +
    '<td>' +
    '<input type="text" data-clear-btn="true" name="reply"
id="inputReply" value="" style=" margin-right:2px; max-width: 100px;
border: none">' +
    '<button id="replyPostBtn" style="opacity: 1; background-color:
#1797F3; color: white; border: none"
onclick="displayReply();">Post</button>' +
    '</td>');
}

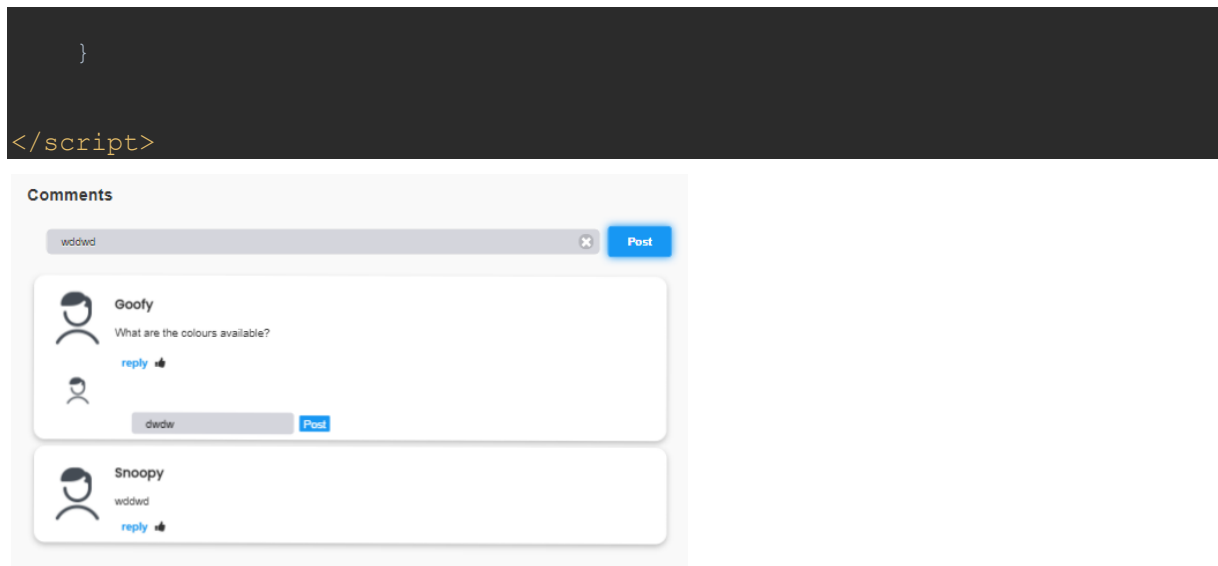
function displayReply() {
  let replyText = $('#inputReply').val();
  console.log('reply is: ' + replyText);
  $("#displayReply").append('' +
    '<h3>Snoopy</h3>' +
    '<p id="snoopyReply">' + replyText + '</p>'
  );
  $("#repliersPhoto").append('');
}

function myFunction(x) {
  x.classList.toggle("fa-thumbs-down");
}

function openReply1() {
  console.log('type a reply');
  $("#replyTextField1").append('' +
    '<td></td>' +
    '<td>' +
    '<input type="text" data-clear-btn="true" name="reply"
id="inputReply1" value="" style=" margin-right:2px; max-width: 100px;
border: none">' +
    '<button id="replyPostBtn1" style="opacity: 1; background-
color: #1797F3; color: white; border: none"
onclick="displayReply1();">Post</button>' +
    '</td>');
}

function displayReply1() {
  let replyText = $('#inputReply1').val();
  console.log('reply is: ' + replyText);
  $("#displayReply1").append('' +
    '<h3>Snoopy</h3>' +
    '<p id="snoopyReply1">' + replyText + '</p>'
  );
  $("#repliersPhoto1").append('');
}

```



## 2.12 Rating

The rating functionality of the application is implemented in the Order History page. As the user purchases the products and after receiving and experiencing the usage of the products, he/she is able to rate the product as needed. In both orientations of iPhone and iPad, rating needs to be done inside the feedback dialog box.

We used stars to indicate the rating as it a universal icon used to show rating; thus, the user will be familiar with the icon. When a user clicks on one of the five stars, all starts up to the one that was clicked is made to change to its filled state. An unfilled star indicates that it is not clicked, or a rating has not been given yet. We used blue for the rating icon fill state to keep in with the application's colour theme.

```
<div data-role="popup" id="popuprating" data-theme="a" class="ui-content">
  <link rel="stylesheet"
href="../../../common/js/star/SimpleStarRating.css">
  <style>
    body {
      background-color: #999;
      font-family: sans-serif;
      margin: 0;
    }

    main {
      background-color: white;
      width: auto;
      margin: 0 auto;
      padding: 50px;
      text-align: center;
    }

    table {
      display: inline-block;
    }
  </style>
</div>
```

```

    td {
        padding: 1em;
    }

    .golden {
        color: #ee0;
        background-color: #444;
    }

    .big-red {
        color: #f11;
        font-size: 50px;
    }
</style>

<main>
    <p>
        <table>
            <tr>
                <td><span id="rating" class="rating" data-default-
rating="0"></span>
                    <br/>
                    <span class='your-choice-was' style='display: none;
font-size: 20px; text-align: center;'>Your rating was <span style="font-
weight: bolder" class='choice'></span></span>
                </td>
            </tr>
        </table>

    </p>

</main>

<div style="display: flex" data-role="popup" id="rate-feedback">
    <h3>Thank you for your Rating !!! </h3> 
</div>
<script src="../../../common/js/star/SimpleStarRating.js"></script>
</div>

```

```

var SimpleStarRating = (function () {
    function SimpleStarRating(target) {
        function attr(name, d) {
            var a = target.getAttribute(name);
            return (a ? a : d);
        }

        var max = parseInt(attr('data-stars', 5)),
            disabled = typeof target.getAttribute('disabled') !=
'undefined',
            defaultRating = parseFloat(attr('data-default-rating', 0)),
            currentRating = -1,
            stars = [];

        target.style.display = 'inline-block';
    }

```

```

for (var s = 0; s < max; s++) {
    var n = document.createElement('span');
    n.className = 'star';
    n.addEventListener('click', starClick);
    if (s > 0)
        stars[s - 1].appendChild(n);
    else
        target.appendChild(n);

    stars.push(n);
}

function disable() {
    target.setAttribute('disabled', '');
    disabled = true;
}
this.disable = disable;

function enable() {
    target.removeAttribute('disabled');
    disabled = false;
}
this.enable = enable;

function setCurrentRating(rating) {
    currentRating = rating;
    target.setAttribute('data-rating', currentRating);
    showCurrentRating();
}
this.setCurrentRating = setCurrentRating;

function setDefaultRating(rating) {
    defaultRating = rating;
    target.setAttribute('data-default-rating', defaultRating);
    showDefaultRating();
}
this.setDefaultRating = setDefaultRating;

this.onrate = function (rating) {};

target.addEventListener('mouseout', function () {
    disabled = target.getAttribute('disabled') !== null;
    if (!disabled)
        showCurrentRating();
});

target.addEventListener('mouseover', function () {
    disabled = target.getAttribute('disabled') !== null;
    if (!disabled)
        clearRating();
});

showDefaultRating();

function showRating(r) {
    clearRating();
    for (var i = 0; i < stars.length; i++) {
        if (i >= r)
            break;
        if (i === Math.floor(r) && i !== r)
            stars[i].classList.add('half');
    }
}

```

```

        stars[i].classList.add('active');
    }
}

function showCurrentRating() {
    var ratingAttr = parseFloat(attr('data-rating', 0));
    if (ratingAttr) {
        currentRating = ratingAttr;
        showRating(currentRating);
    } else {
        showDefaultRating();
    }
}

function showDefaultRating() {
    defaultRating = parseFloat(attr('data-default-rating', 0));
    showRating(defaultRating);
}

function clearRating() {
    for (var i = 0; i < stars.length; i++) {
        stars[i].classList.remove('active');
        stars[i].classList.remove('half');
    }
}

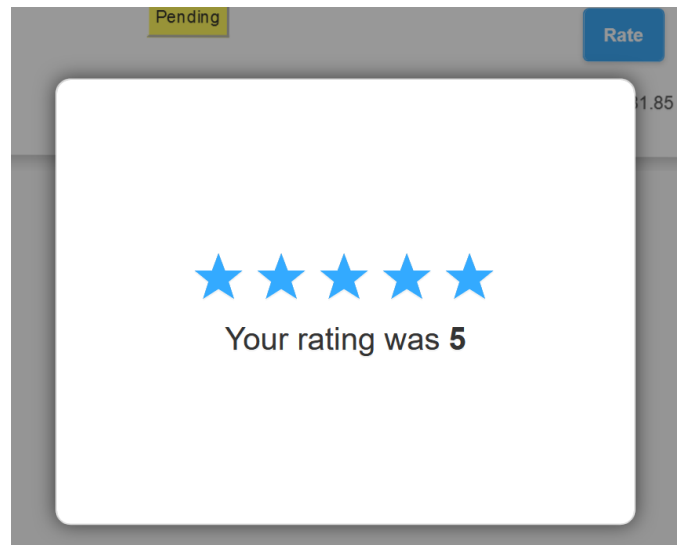
function starClick(e) {
    if (disabled) return;

    if (this === e.target) {
        var starClicked = stars.indexOf(e.target);
        if (starClicked !== -1) {
            var starRating = starClicked + 1;
            setCurrentRating(starRating);
            if (typeof this.onrate === 'function')
                this.onrate(currentRating);
            var evt = new CustomEvent('rate', {
                detail: starRating,
            });
            target.dispatchEvent(evt);
        }
    }
}

return SimpleStarRating;
})();

```





### 2.13 AR

There is a feature in the mobile web application where the user is able to see the AR view of the products. For that user has to go to the Product page of a particular product and there he/she can find the AR View button next to the product image. When clicking on that button, a popup appears with the AR model of the product. The user can check the model by swiping the image to different directions. Therefore, the user can see the full outer image of the product.

As to implement this feature in the application, an AR library called aframe.io was used. By saving different .glTF models to the database and was using them appropriately was included in the procedure of implementation of the feature.

```
<div style="max-width: none; min-width: 300px; top: 140px; height: 500px;"
data-role="popup" id="viewAR" data-theme="a"
class="ui-content">

    <meta name="description" content="Model Viewer (VR / AR) • A-Frame">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no"/>
    <script src="../../../common/js/aframe.min.js"></script>
    <script src="../../../common/js/aframe-extras.min.js"></script>

    <script src="../../../common/js/hidden-on-enter-ar.js"></script>
    <script src="../../../common/js/ar-shadows.js"></script>
    <script src="../../../common/js/model-viewer.js"></script>
    <script src="../../../common/js/background-gradient.js"></script>

    <body>
    <a-scene
        embedded
        renderer="colorManagement: true;"
        model-viewer="glTFModel: #man; title: <?php echo
$json[$id]['name'] ?>">
```

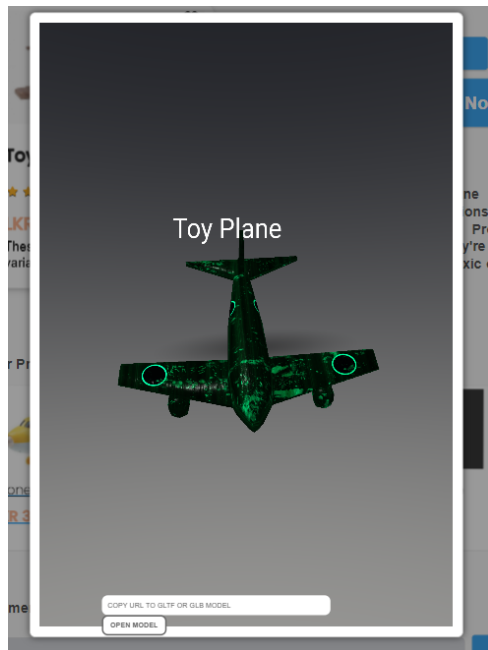
```

<a-assets>

  <a-asset-item id="man" src=<?php echo $json[$id]['ar_view'] ?>
    response-type="arraybuffer" crossorigin="anonymous"></a-asset-
item>

  </img>
</a-assets>
</a-scene>
</body>
</div>

```



## 2.14 Differentiation between iPhone and iPad implementation

When the user opens a page in either iPhone or iPad, it determines the user-agent of the device and then it renders the appropriate view for the user. Therefore, the user can access the application using the same domain when browsing from different types of devices (iPhone or iPad).

```

// Initial redirect rules for the site
var device = "iPad"; // default should be iPad as that is closest to PC
if (new RegExp("iPhone").test(navigator.userAgent) ||
    new RegExp("Android").test(navigator.userAgent)) {
    device = "iPhone";
}
if (new RegExp("iPad").test(navigator.userAgent)) {
    device = "iPad";
}
console.log("You are using an " + device);
$(document).ready(function () {
    $("#device").append(device);
    $("#device").css('color', 'red');
    $("#device").css('font-weight', 'bold');
});

```

```
setTimeout(function () {  
    if (device == "iPhone") location.href = "../iphone/pages/home.php";  
    if (device == "iPad") location.href = "../ipad/pages/home.php";  
}, 1500);  
});
```

## References

Anon., n.d. *AFRAME.IO Introduction*. [Online]

Available at: <https://aframe.io/docs/1.1.0/introduction/#features>

Anon., n.d. *jQuery Documentation*. [Online]

Available at: <https://demos.jquerymobile.com/1.1.2/docs/pages/index.html>

Anon., n.d. *PHP Tutorial*. [Online]

Available at: <https://www.w3schools.com/php/>