

**UNIVERSITY OF  
WESTMINSTER**



**INFORMATICS  
INSTITUTE OF  
TECHNOLOGY**

**INFORMATICS INSTITUTE OF TECHNOLOGY**

**in collaboration with**

**the University of Westminster, UK**

**BEng/BSc (Hons) DEGREE PROGRAMME in  
Software Engineering**

**Advanced Server-side Web programming  
(6COSC011C)**

**Coursework 02 – Report Submission**

Name : M.A Samasha Nayananthi

IIT ID : 2017323

UOW ID : w1697763

## Table of Contents

1.	Answers to the three questions.....	1
1.1	Author's main conclusion concerning web frameworks.....	1
1.2	Main arguments made by the author to support his conclusion.....	2
1.3	My arguments on web frameworks.....	3
2.	WishIT .....	7
3.	Screenshots of the WishIT web site.....	7
3.1.	Home page .....	7
3.2.	Sign up page.....	8
3.3.	Login page .....	9
3.4.	Wish list .....	9
3.5.	Shared wish list .....	14
3.6.	Logout.....	15
4.	Views .....	15
4.1.	Home page (homepage.php) .....	15
4.2.	Footer (footer.php) .....	17
4.3.	Sign up (signUp.php) .....	18
4.3.1.	Sign up page CSS (signup.css) .....	20
4.4.	Login (login.php) .....	22
4.4.1.	Login page CSS (login.css).....	24
4.5.	Wish list (wishList.php).....	25
4.5.1.	Wish list page CSS (wishList.css) .....	29
4.5.2.	Wish list user data handling JavaScript (wishlist-user.js).....	35
4.5.3.	Wish list dropdown data handling JavaScript (dropdown.js) .....	36
4.5.4.	Wish list basic data handling JavaScript (wishlist-list.js) .....	37

4.5.5. Wish list items data handling JavaScript (wishlist-items.js).....	40
4.6. Shared wish list (sharedWishList.php) .....	44
4.6.1. Share wish list page CSS (sharedWishList.css) .....	48
5. Controllers.....	51
5.1. HomePage .....	51
5.2. Login.....	51
5.3. Register .....	51
5.4. WishList.....	52
6. API Controllers .....	52
6.1. Authentication.....	52
6.2. User.....	53
6.3. Occasion.....	54
6.4. Priority .....	55
6.5. ListDetails .....	55
6.6. WishItem.....	58
7. Models.....	61
7.1. AuthenticateModel.....	61
7.2. UserModel.....	62
7.3. OccasionModel .....	62
7.4. PriorityModel.....	63
7.5. ListModel.....	64
7.6. WishItemModel .....	65
8. Other configurations .....	66
8.1. Set the base URL .....	66
8.2. Set the landing page .....	66
8.3. Logging.....	66

8.4. Session .....	66
8.5. Database.....	67
9. SQL file.....	67

# 1. Answers to the three questions

## 1.1 Author's main conclusion concerning web frameworks

The author's overall conclusion on web frameworks was, knowing the basics before using a web framework is more useful than just learning multiple frameworks without any basic knowledge. Learning basics are very important as frameworks that exist change from time to time and new frameworks also introducing to the world. But the underlying concepts and basic knowledge won't change. If we spent the time to learn the basics first, other than running behind multiple frameworks, it will help to develop our websites easier, as we have an understanding of how the abstracted layer works. Knowing the basics helps to evaluate frameworks and helps to pick the best framework according to the requirement. This way developers will be able to compare and contrast frameworks and choose the best framework that fits their requirements.

In the wish list application, I am using the MVC framework with CodeIgniter for server-side and backbone js which also provide the MVC framework for the client-side. As I had prior knowledge and a basic understanding of HTML, JavaScript, CSS, SQL, etc. it was easy for me to understand the frameworks efficiently and use them.

In backbone js data requests are handle through defining a model. In the following code fragment, I'm defining a model by setting the URL (API request URL) and setting the structure of the model. And then when I need to call a GET method I just have to call model.fecth(), like vice backbone js provide post and delete method to handle POST, PUT and DELETE requests.

```
var List = Backbone.Model.extend({
  url: base_url + "api/ListDetails/list",
  idAttribute: 'id',
  defaults: {"id": null, "name": "", "description": "", "occasionId": null,
    "occasion": "", "userId": null}
});

var list = new List();
```

This same request can be handled through ajax library without using any js framework as follows.

```
$.ajax({
  url: "<?php echo base_url()."api/ListDetails/list" ?>",
  method: "GET"
},
success: function() {
```

```
    //render the html  
    }  
  });
```

## 1.2 Main arguments made by the author to support his conclusion

Using a framework helps us to use things that already implemented without re-writing it from scratch as long as we know how the framework works, how our requirement get fulfilled with the framework and the framework we use will only provide what we want (not providing or forcing us to use things that we don't want to use). I agree on this argument with the author. Frameworks make our life easier. But if in order to use the framework I have to adapt my implementation and change my requirements accordingly, then I'm just following someone else's implementation rather than my own work. And also, if the framework provides things that I won't use ever in my application then it is just a waste. I can go with a library that only supports my requirement rather than implement a framework which comes with load of extra things that I don't want.

Even though, it is easy to use frameworks as a beginner, it is just the fast track. If you start using frameworks without knowing the concepts, you will be addicted to frameworks but still you don't know how everything works which is not good as a developer. I agreed on this with the author. Every developer should know the basic concepts and should have a fundamental understanding on a specific programming language. Learning any framework will be easy if we know the underlying concepts of it.

We can write complicated websites without a framework. GitHub is now built with query selectors, ajax calls, event handling and DOM elements without any help from frameworks. We just have to spend a little time on reading the MDN documents which provides the information about open web technologies. On my opinion, if we know the basic concepts and a using a framework makes the life easier, it is okay to go with a suitable framework that fit with our requirements.

For handling the API requests, we can simply use a library which provides an easy way to interpreting requests and responses over a socket. In here we use abstraction over HTTP. But if we use a framework in this, it abstracts away a whole mess. And we follow a someone else's idea on web programming.

I am agreeing with the author's argument that every developer should have the basic knowledge/foundation. A developer should be aware how to write a SQL query other than using inbuilt method to perform CRUD operations. In the coursework I am using CodeIgniter Active Record set to perform CRUD operations. Which is very easy and faster way to write a database transaction.

```
function getListDetails($userId) {
    $this->db->select('l.id, l.name, l.description, l.occasion_id, l.user_id,
o.name occasion');
    $this->db->join('occasions o', 'l.occasion_id = o.id', 'left');
    $query = $this->db->get_where('list_details l', array('user_id' =>
$userId));
    return $query->row();
}
```

But also, I have the understanding on what is really happens inside these select(), join(), get\_where() methods and how they create the SQL query. Following is the SQL string built from the above select active record.

```
$sql = "SELECT l.id, l.name, l.description, l.occasion_id, l.user_id, o.name
occasion
FROM list_details l
LEFT JOIN occasions o ON l.occasion_id = o.id
WHERE user_id = " + $id;
```

Likewise, when using a framework, we should know the underlying concepts of each components in the framework.

### 1.3 My arguments on web frameworks

- If we use a framework without knowing the concepts and language behind it, then we only learn the framework, not the language or theories. So, as a beginner it is best to learn the basics and the programming language first and use it other than learning a framework.

The way we use backbone js is quite different than vanilla JavaScript. In backbone js, we have models, collections, and views which provide the MVC architecture. A model helps to perform API requests by simply calling fetch(), save() and destroy() methods. By following this we will be adapted to the framework and get less JavaScript knowledge.

```
var List = Backbone.Model.extend({
    url: base_url + "api/ListDetails/list",
    idAttribute: 'id',
    defaults: {"id": null, "name": "", "description": "", "occasionId": null,
        "occasion": "", "userId": null}
});
var list = new List();
var ListView = Backbone.View.extend({
```

```

    el: '#listMain',
    initialize: function () {
        this.listenTo(this.model, 'sync', this.render);
        this.model.fetch();
    },
    render: function () {
        // html view
    },
    wait: true
  });
var listView = new ListView({model:list});

```

- Limited to the framework's behaviour and forced to respects to its limits. And also, frameworks control the structure of our codes.

In the coursework always I had to stick with the models and their views. With CodeIgniter and backbone js, we adapt to the MVC (model-view-controller) architecture and its code structure. As for an example, when we using backbone js, we create models bind collections to the model as for the requirement and bind those with views. The Underscore.js template acts as the view and the views, we create act as controllers. Inside the views we have to implement the rendering mechanism according to our requirements. Following is a code fragment including creating a model to map wish item, a collection bind with wish item model to load a list of wish items and the view of the wish item collection.

```

var WishItem = Backbone.Model.extend({
    url: base_url + "api/WishItem/wishItems",
    idAttribute: 'id',
    defaults: {"id": null, "title": "", "listId": null, "priorityId": null,
    "itemUrl": "", "price": null, "quantity": null, "priorityLvl": null,
    "priority": ""}});
var wish = new WishItem();

var WishItems = Backbone.Collection.extend({
    model: WishItem,
    comparator: 'priorityLvl',
    url: base_url + "api/WishItem/wishItems"
});

```



```

var wishItems = new WishItems();

var WishItemsView = Backbone.View.extend({
  el: '#wishItems',
  initialize : function () {
    this.model.fetch({async:false});
    this.listenTo(this.model, 'sync', this.render());
  },
  render : function () {
    //build the html template
  },
  wait: true
});

var wishItemsView = new WishItemsView({model:wishItems});

```

- Might affect the loading time of the web site.

Frameworks come with pre-defined code sets. We might not need all the code. But this bulk of code will load with the framework. These files might include large files that have an impact on the speed and the performance of the website.

On the other hand, frameworks can be very useful if we use them correctly.

- Frameworks are efficient. Task that needed hundreds of lines of code to implement can be done using inbuilt functions easily. For an example using CodeIgniter we can easily connect with the database and by using active record set we can easily interact with the database other than writing connections to the database and writing SQL queries. Following is an example from the coursework where active record set was used to perform CRUD operations.

```

function getListDetails($userId) {
  $this->db->select('l.id, l.name, l.description, l.occasion_id, l.user_id,
o.name occasion');
  $this->db->join('wi_occasions o', 'l.occasion_id = o.id', 'left');
  $query = $this->db->get_where('wi_list_details l', array('user_id' =>
$userId));
  return $query->row();
}

function saveList($name, $description, $occasionId, $userId) {
  $list = array('name' => $name, 'description' => $description,

```

```

'occasion_id' => $occasionId,
  'user_id' => $userId);
$this->db->insert('wi_list_details', $list);

return $this->db->insert_id();
}

function updateList($id, $name, $description, $occasionId) {
  $data = array('name' => $name, 'description' => $description,
'occasion_id' => $occasionId);
  $this->db->update('wi_list_details', $data, array('id' => $id));
}

function deleteListById($id) {
  $this->db->delete('wi_list_details', array('id' => $id));
}

```

- For widely used frameworks, help/support available. When coding, we might end up in problems. If the framework we use is a widely used framework, then the problem might be a common problem and the answer to our problem is already available. And also, standard templates and themes might be already available to use.

While coding with backbone js I encountered many problems. One problem that I encountered is, when I tried to deal with delete requests which were handled by the destroy() method in backbone js, it didn't send an id in the URL even though the id is set in the model. When I'm searching for a solution, I got an answer from a discussion on GitHub and it said to custom the URL inside my destroy() method. And this helps me to overcome my problem.

```

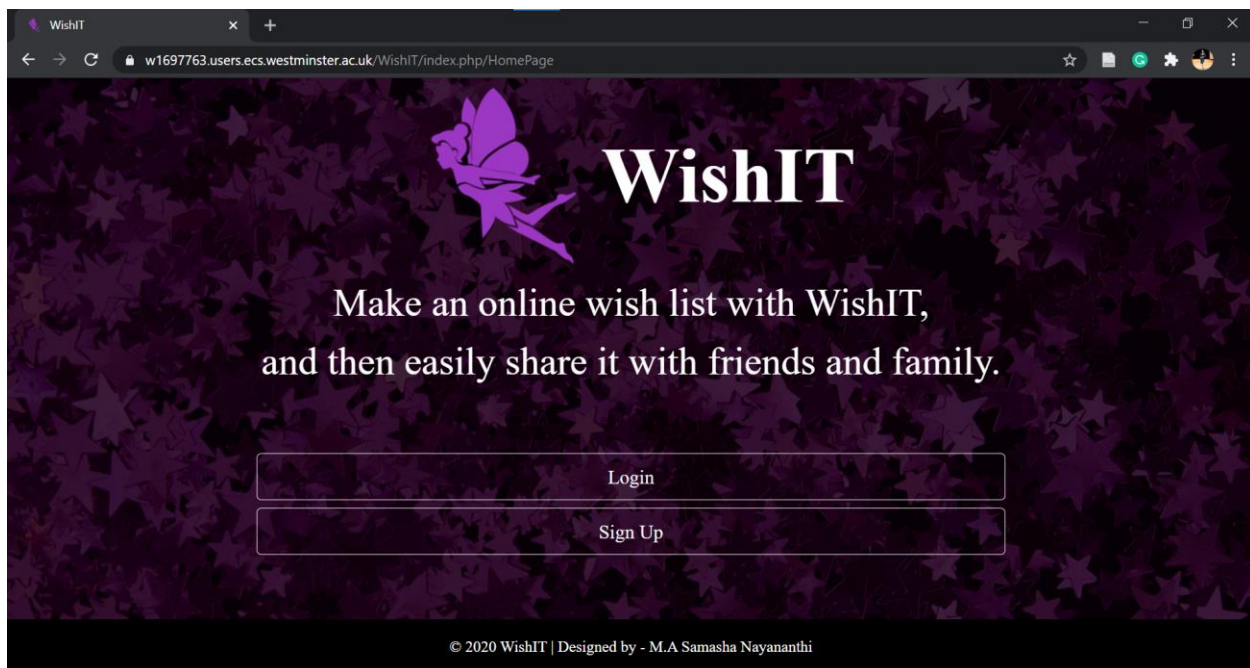
item.destroy({
  url: base_url + "api/WishItem/wishItems/id/" + item.get('id'),
  success: function () {
    wishItemsView.render();
  },
  error: function () {
    alert('Error occurred while deleting the item.');
```

## 2. WishIT

URL - <https://w1697763.users.ecs.westminster.ac.uk/WishIT/index.php/HomePage>

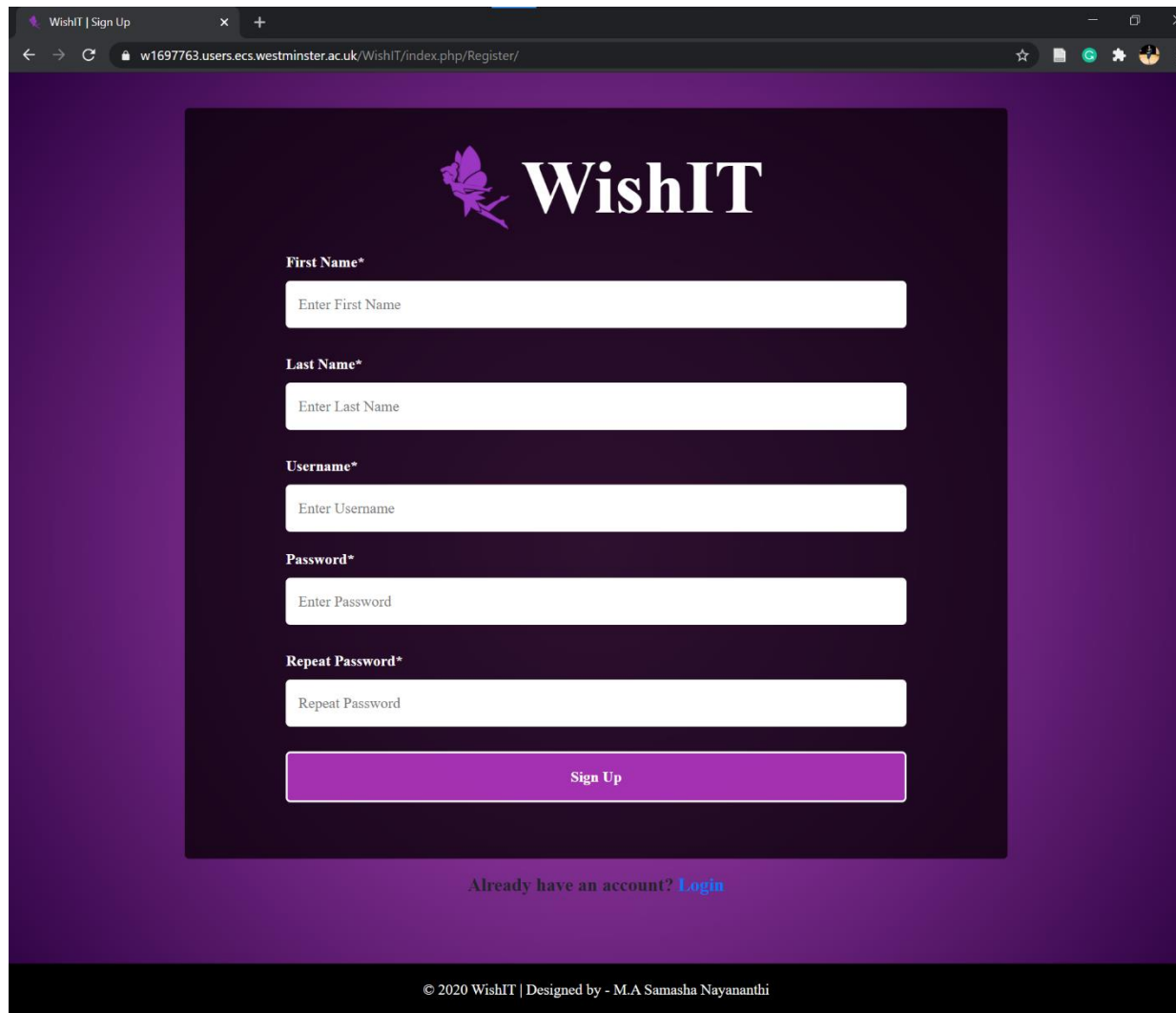
## 3. Screenshots of the WishIT web site

### 3.1. Home page



This is the landing page of the web site. A new user can sign up to the website. And if the user has a account created, user can simply login to the website.

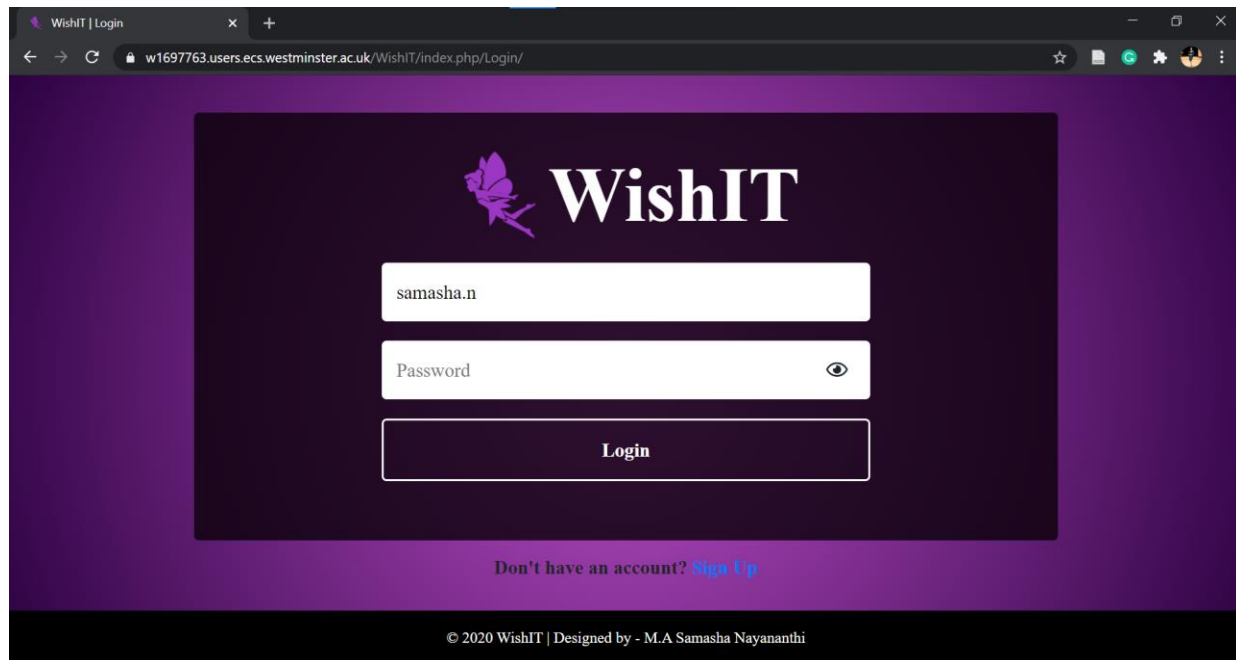
### 3.2. Sign up page



The screenshot shows a web browser window with the title 'WishIT | Sign Up'. The address bar displays the URL 'w1697763.users.ecs.westminster.ac.uk/WishIT/index.php/Register/'. The page has a dark purple background. In the center, there is a white rectangular form with the 'WishIT' logo at the top. The logo consists of a stylized purple butterfly icon followed by the text 'WishIT' in a white serif font. Below the logo, the form contains five input fields, each with a label and an asterisk indicating it is required: 'First Name\*', 'Last Name\*', 'Username\*', 'Password\*', and 'Repeat Password\*'. Each field has a placeholder text: 'Enter First Name', 'Enter Last Name', 'Enter Username', 'Enter Password', and 'Repeat Password'. At the bottom of the form is a large purple button with the text 'Sign Up' in white. Below the form, there is a link that says 'Already have an account? [Login](#)'. At the very bottom of the page, there is a small copyright notice: '© 2020 WishIT | Designed by - M.A Samasha Nayananthi'.

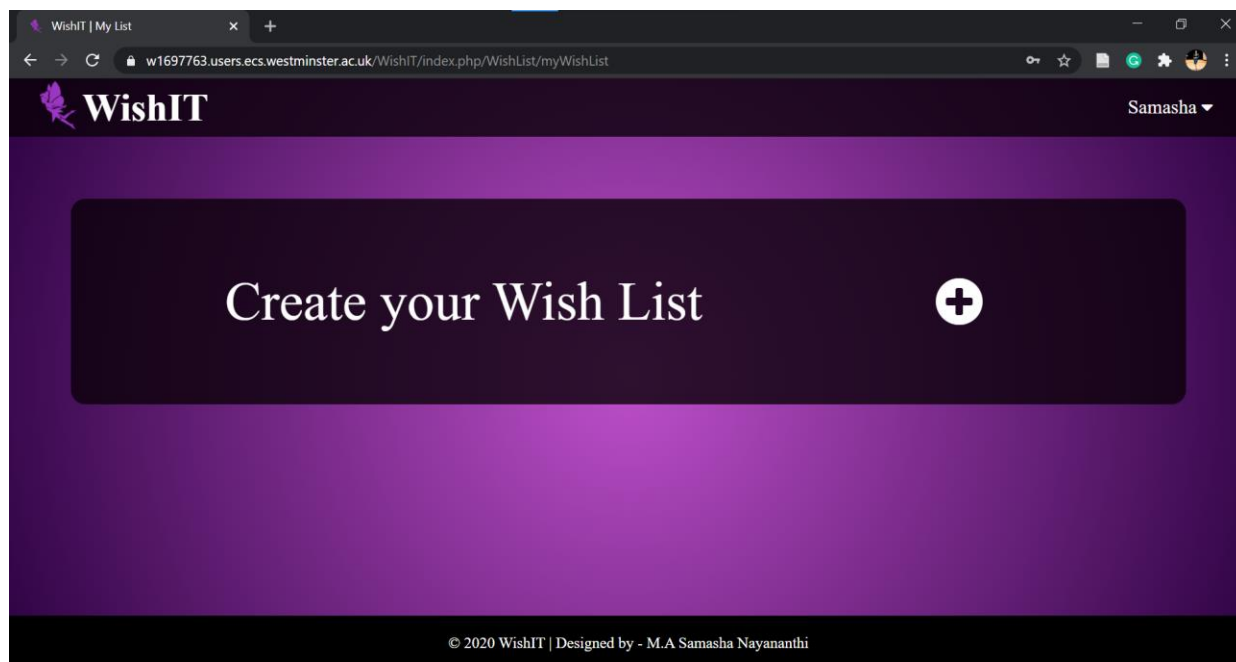
A new user can sign up to the website using above form. This form is validated to fill out all the fields and all the input validations have been implemented. Once the user clicks on sign up button system will first check whether there is a user with the same username that user provide (check whether username already taken or not). If the user account created successfully, user will be asked to login to the system.

### 3.3. Login page



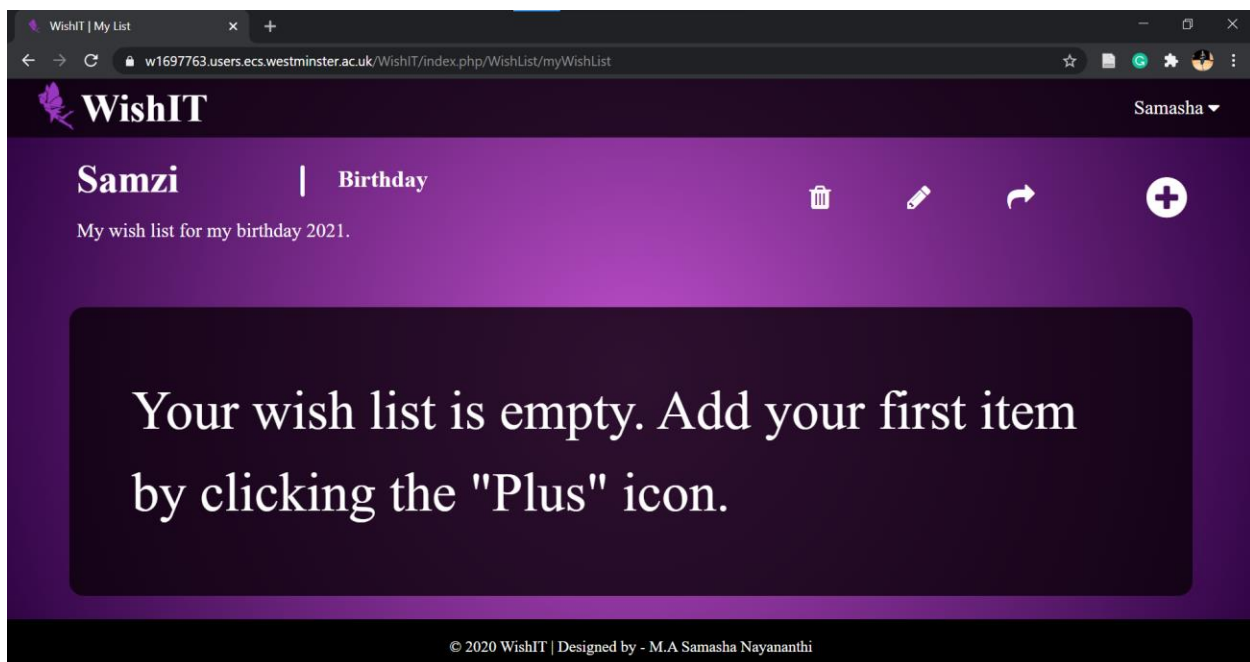
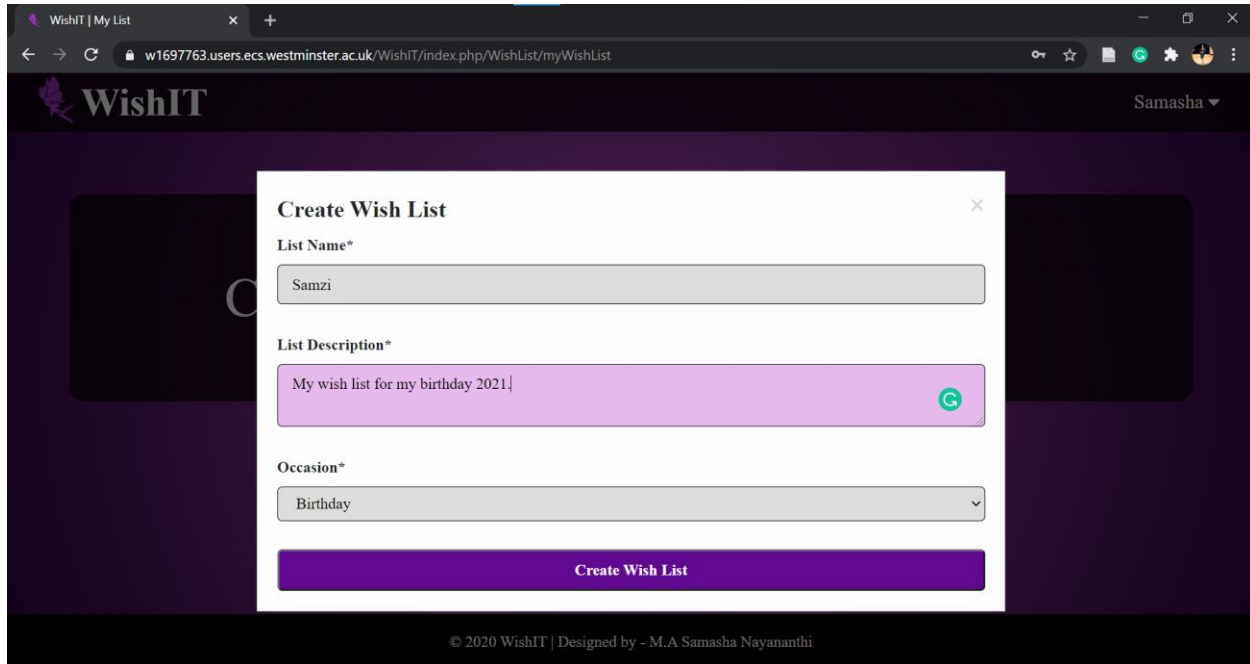
Once the user created an account, user can login to the system using above form. User authentication and authorisation is handled accordingly.

### 3.4. Wish list

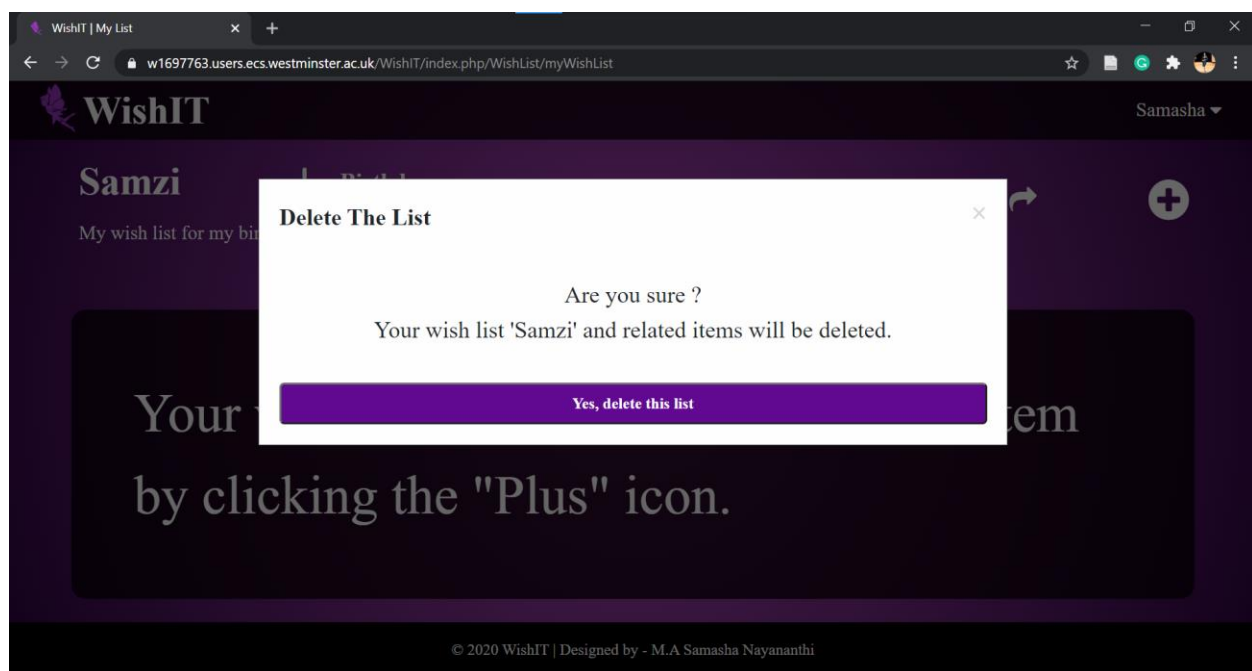
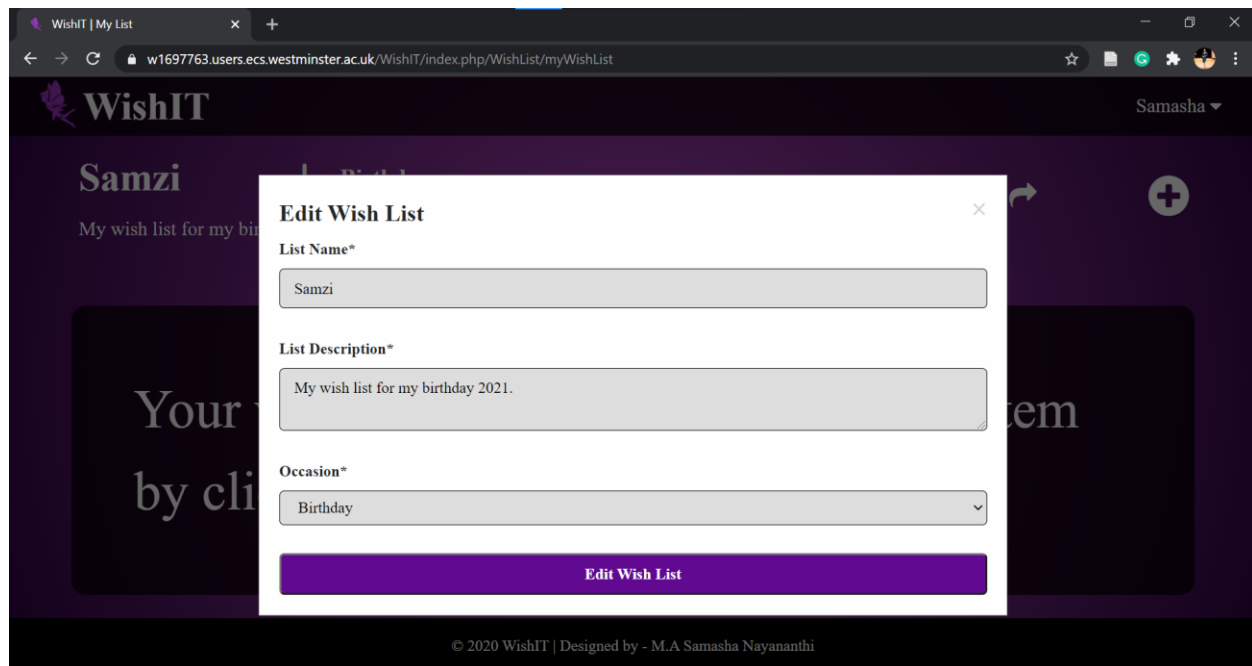


When the account is initially created, the wish list page will look like the above figure. User can create their wish list using the plus icon.

Following is the wish list creating pop up.



User also has the option to edit or delete the created wish list. Following are the edit and delete pop ups.



Once the user deletes the list, all the related items will also be deleted. So, the user is notified with a pop up when deleting the list.

Once the wish list is created user can add items to the list using plus icon. Items will be sorted according to their priority.

The screenshot shows the 'Add an Item' modal form in the WishIT application. The form is titled 'Add an Item' and contains the following fields:

- Title\***: A text input field containing 'Chocolate gift pack'.
- Item URL\***: A text input field containing 'https://chocolate.lk/product/chocolate-gift-box-1/'.
- Priority\***: A dropdown menu with 'I want it' selected.
- Price\***: A text input field containing '3990'.
- Quantity\***: A text input field containing '1'.

At the bottom of the form is a purple button labeled 'Add an Item'. The background shows the user's profile 'Samzi' and the title 'My wish list for my birthday 2021'.

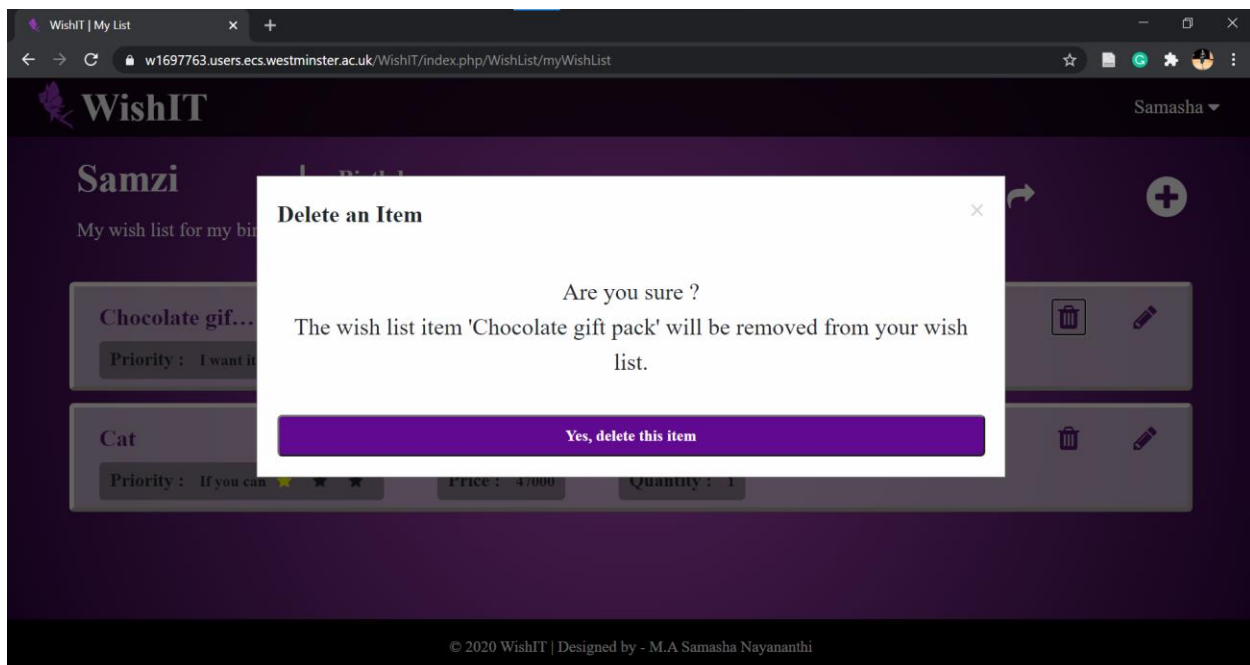
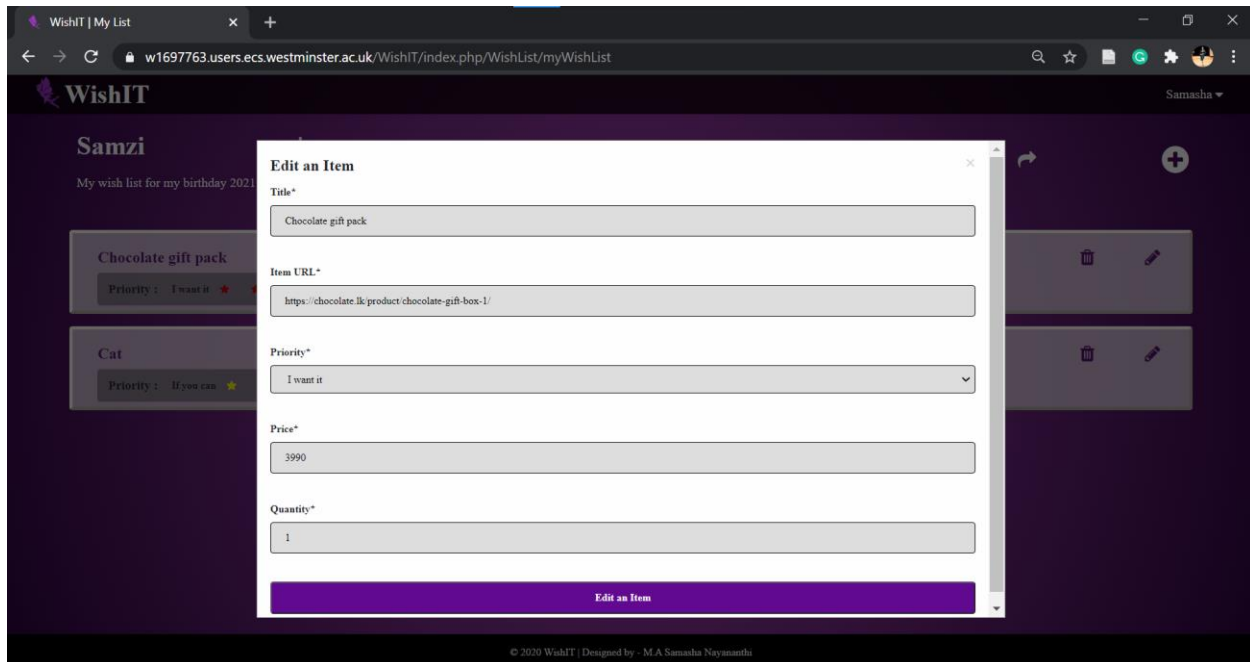
The screenshot shows the 'WishIT' application interface with the user's wish list. The header shows the user's name 'Samzi' and the occasion 'Birthday'. Below the header, there are icons for deleting, editing, and adding items. The wish list contains two items:

Item Name	Item URL	Priority	Price	Quantity	Actions
Chocolate gif...	<a href="https://chocolate.lk/product/chocolate-gift-box-1/">https://chocolate.lk/product/chocolate-gift-box-1/</a>	I want it (3 stars)	3990	1	Delete, Edit
Cat	<a href="https://ikman.lk/en/ad/persian-cat-for-sale-gampaha-224">https://ikman.lk/en/ad/persian-cat-for-sale-gampaha-224</a>	If you can (3 stars)	47000	1	Delete, Edit

The footer of the page reads '© 2020 WishIT | Designed by - M.A Samasha Nayananthi'.

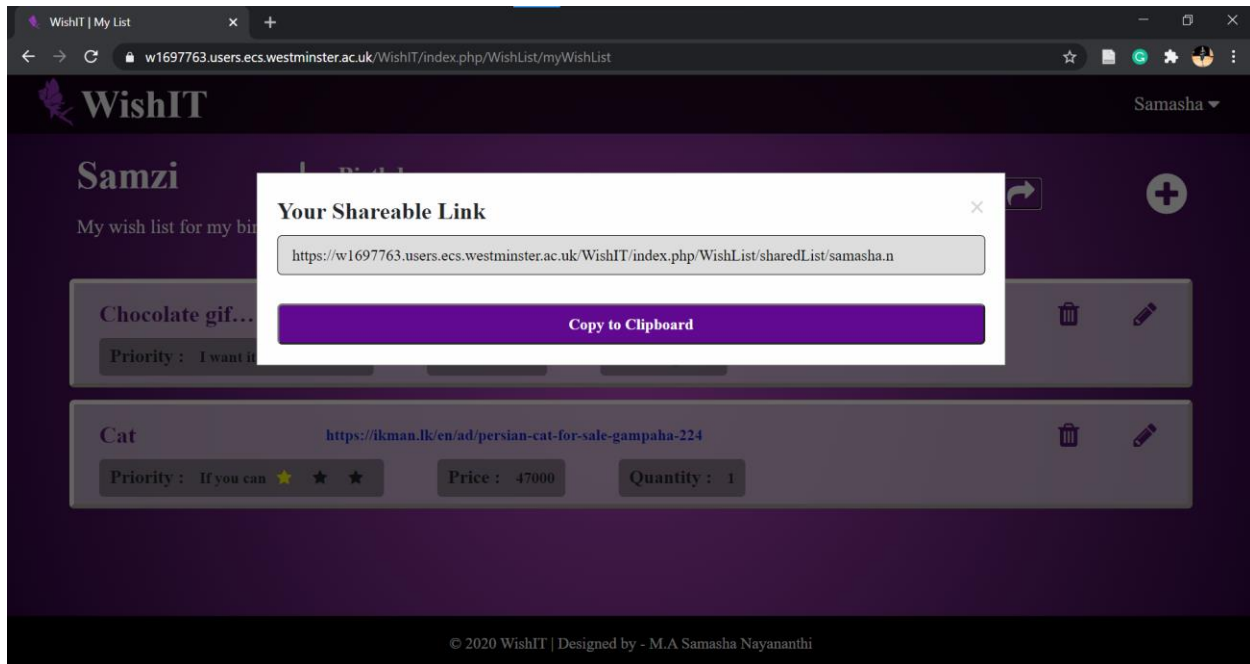


User has the opportunity to edit and delete the wish list items as well. Following are the wish list item edit and delete pop ups.

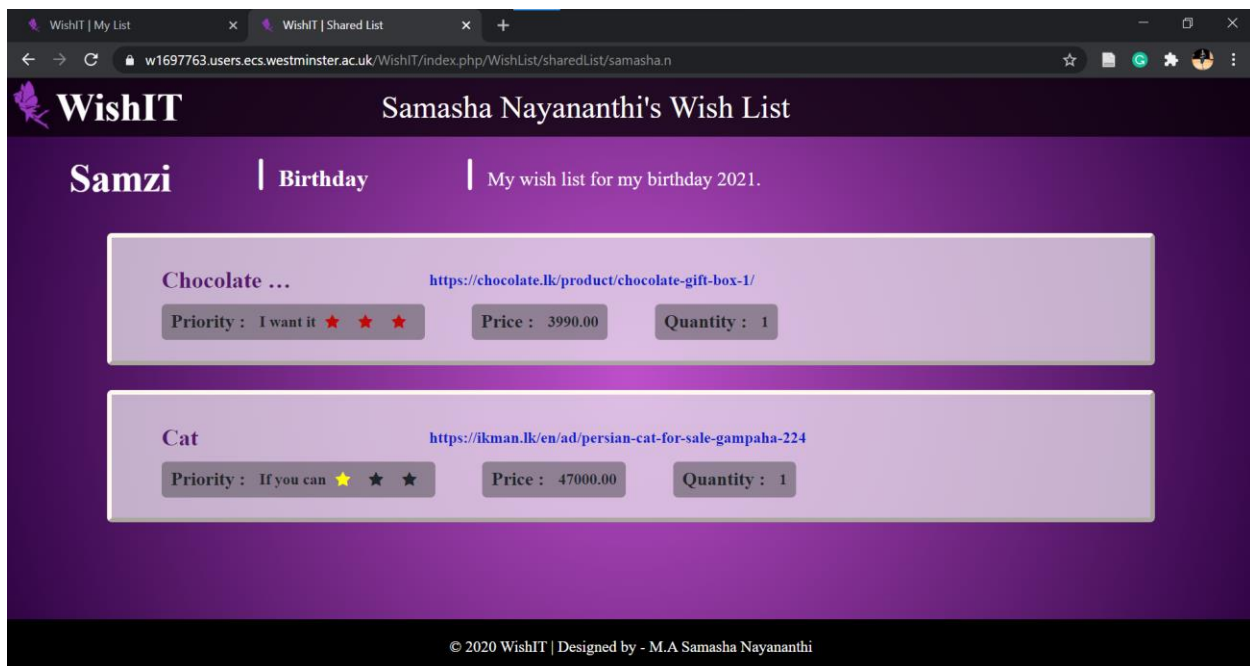


To perform all the add, edit, delete actions, user must be an authorised user.

User also has the option to get a shareable link of the wish list. User can copy the URL by using copy to clipboard button.

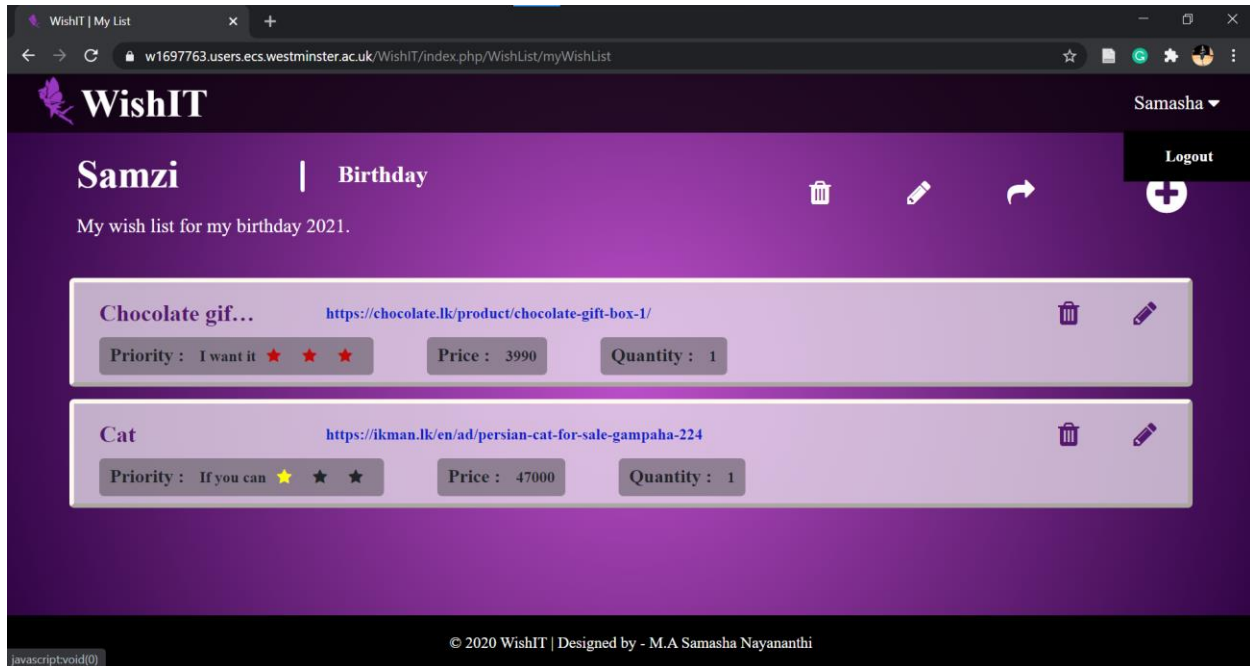


### 3.5. Shared wish list



Anyone with the provided URL can access this shared wish list page without any login process, but cannot edit anything in the list.

### 3.6. Logout



User can use the logout button in the top right corner to logout from the system. Once logout, user will be redirect to the home page.

## 4. Views

### 4.1. Home page (homepage.php)

```
<html lang="en">
<head>
  <title>WishIT</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="shortcut icon" type="image/x-icon"
href="/WishIT/images/favicon.ico" />
  <link rel="stylesheet" href="/WishIT/css/bootstrap.min.css">
  <script src="/WishIT/js/jquery-3.5.1.min.js"></script>
  <script src="/WishIT/js/popper.min.js"></script>
  <script src="/WishIT/js/bootstrap.min.js"></script>

  <style>
    body {
      height: 100%;
      margin: 0;
      background-image: url("/WishIT/images/bg.jpg");
      background-size: cover;
    }
  </style>
</head>
```

```
.main-container {
  width: 100vw;
  height: 100vh;
  background-color: #0a0009cc;
  font-family: serif;
}

.page-name {
  display: inline-flex;
  width: 100vw;
}

.logo {
  float: left;
  margin: auto 0 auto auto;
}

.name {
  color: white;
  float: right;
  margin: auto auto auto 0;
  font-size: 500%;
  font-weight: bold;
}

.desc {
  color: white;
  text-align: center;
  font-size: 250%;
}

.my-btn {
  width: 60vw;
  margin: 5vw 20vw 2vw 20vw;
}
</style>
</head>

<body>

<div class="main-container">
  <div class="page-name">
    
    <div class="name">WishIT</div>
  </div>

  <div class="desc">
    Make an online wish list with WishIT, <br>
    and then easily share it with friends and family.
  </div>

  <div class="my-btn">
    <button type="button" class="btn btn-outline-light btn-block btn-lg"
      onclick='Login()'>Login</button>
    <button type="button" class="btn btn-outline-light btn-block btn-lg"
      onclick='SignUp()'>Sign Up</button>
  </div>
</div>
```

```
</div>

<?php
include_once("footer.php");
?>

<script>
    function Login() {
        window.location = "<?php echo base_url()."Login/" ?>";
    }

    function SignUp() {
        window.location = "<?php echo base_url()."Register/" ?>";
    }
</script>

</body>
</html>
```

## 4.2. Footer (footer.php)

```
<html>

<head>
    <style>
        .footer {
            text-align: center;
            color: white;
            position: fixed;
            bottom: 0;
            width: 100%;
            background-color: #000000;
            font-family: serif;
        }

        .para {
            display: block;
            margin-block-start: 1em;
            margin-block-end: 1em;
            margin-inline-start: 0px;
            margin-inline-end: 0px;
        }

        .cv {
            color: white;
        }
    </style>
</head>

<body>

<div class="footer">
    <footer>
```

```

        <p class="para">&copy; 2020 WishIT | Designed by - <a href="<?php echo
base_url() . "/HomePage/myCV" ?>"
                                class="cv">M.A Samasha
Nayananthi</a></p>
    </footer>
</div>

</body>
</html>

```

### 4.3. Sign up (signUp.php)

```

<html lang="en">
<head>
    <title>WishIT | Sign Up</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="shortcut icon" type="image/x-icon"
href="/WishIT/images/favicon.ico" />
    <link rel="stylesheet" href="/WishIT/css/bootstrap.min.css">
    <link href="/WishIT/css/signup.css" rel="stylesheet">

    <script src="/WishIT/js/jquery-3.5.1.min.js"></script>
    <script src="/WishIT/js/popper.min.js"></script>
    <script src="/WishIT/js/bootstrap.min.js"></script>
    <script src="/WishIT/js/underscore.js" type="text/javascript"></script>
    <script src="/WishIT/js/backbone.js" type="text/javascript"></script>

</head>

<body>

<div class="container">
    <div class="page-name">
        
        <div class="name">WishIT</div>
    </div>

    <div class="sign-up-box">
        <label for="fname" class="lbl">First Name*</label>
        <input type="text" placeholder="Enter First Name" name="fname"
id="fname" required>

        <label for="lname" class="lbl">Last Name*</label>
        <input type="text" placeholder="Enter Last Name" name="lname"
id="lname" required>

        <label for="username" class="lbl">Username*</label>
        <input type="text" placeholder="Enter Username" name="username"
id="username" required>

        <label for="psw" class="lbl">Password*</label>
        <input type="password" placeholder="Enter Password" name="psw"
id="psw" required>

```

```

        <label for="pswRepeat" class="lbl">Repeat Password*</label>
        <input type="password" placeholder="Repeat Password" name="pswRepeat"
id="pswRepeat" required>

        <input type="button" value="Sign Up" class="btn"
onclick="checkForm()" >
    </div>

</div>

<div class="login-div">
    Already have an account? <a href="<?php echo base_url(). 'Login/' ?>"
class="login">Login</a>
</div>

<?php
include_once("footer.php");
?>

<script>

    var User = Backbone.Model.extend({
        url: "<?php echo base_url(). 'api/User/users' ?>",
        idAttribute: 'id',
        defaults: {"id": null, "fname": "", "lname": "", "username": "",
"password":""}
    });

    var user = new User();

    function checkForm() {
        let namePattern = /^[A-Za-z]+$//;
        let fname = document.getElementById('fname').value;
        let lname = document.getElementById('lname').value;
        let username = document.getElementById('username').value;
        let psw = document.getElementById('psw').value;
        let pswRepeat = document.getElementById('pswRepeat').value;

        if (fname === "" || lname === "" || lname === "" || username === ""
|| psw === "" || pswRepeat === "") {
            alert('Please fill all the fields.');
```

```

    } else if(psw !== pswRepeat) {
        alert('The password and repeat password do not match.');
```

```

    } else {
        registerUser(fname, lname, username, psw);
    }

}

function registerUser(fname, lname, username, psw) {
    user.set('fname', fname);
    user.set('lname', lname);
    user.set('username', username);
    user.set('password', psw);
    user.save(null, {async: false,
        success: function (data, statusText, xhr) {
            alert("User registered successfully. Please login.");
            window.location = "<?php echo base_url()."Login/" ?>";
        },
        error: function (data, statusText, xhr) {
            if (statusText.status === 409) {
                alert('Username already exists.');
```

```

            } else {
                alert('Error occurred while registering the user. Please
try again');
```

```

            }
        }
    });
}

</script>

</body>
</html>

```

#### 4.3.1. Sign up page CSS (signup.css)

```

body {
    background: radial-gradient(circle, #b537c4e0, #2c0241);
    font-family: serif;
}

.container {
    position: relative;
    border-radius: 5px;
    background-color: #0a0009cc;
    width: 70%;
    padding: 2%;
    margin: 3% 15% 1% 15%
}

.page-name {
    color: white;
}

```



```
        display: flex;
    }

    .logo {
        width: 100px;
        margin: auto 0 auto auto;
    }

    .name {
        font-size: 500%;
        font-weight: bold;
        margin: auto auto auto 0;
    }

    .sign-up-box {
        width: 80%;
        margin: 1% auto;
    }

    .lbl {
        color: white;
        font-weight: bold;
    }

    input[type=email], [type=text], input[type=password], .btn {
        width: 100%;
        padding: 2%;
        margin: 0 1% 4% 0;
        border: none;
        border-radius: 5px;
    }

    input[type=text]:focus, input[type=password]:focus {
        background-color: #dddddd;
        outline: none;
    }

    .btn {
        color: white;
        font-weight: bold;
        border: solid white;
    }

    .btn:hover {
        color: white;
        background-color: #b537c4e0;
    }

    .login-div {
        text-align: center;
        font-size: 20px;
        font-weight: bolder;
        margin-bottom: 10%;
    }

    .login-div a{
        text-decoration: none;
    }
```

```

        cursor: pointer;
    }

    .login-div a:hover{
        color: white;
    }

    .login {
        color: #007bff;
    }

```

#### 4.4. Login (login.php)

```

<html lang="en">
<head>
    <title>WishIT | Login</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="shortcut icon" type="image/x-icon"
href="/WishIT/images/favicon.ico" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.13.0/css/all.min.css">
    <link rel="stylesheet" href="/WishIT/css/bootstrap.min.css">
    <link href="/WishIT/css/login.css" rel="stylesheet">

    <script src="/WishIT/js/jquery-3.5.1.min.js"></script>
    <script src="/WishIT/js/popper.min.js"></script>
    <script src="/WishIT/js/bootstrap.min.js"></script>
    <script src="/WishIT/js/underscore.js" type="text/javascript"></script>
    <script src="/WishIT/js/backbone.js" type="text/javascript"></script>

</head>

<body>

<div class="login-box">
    <div class="page-name">
        
        <div class="name">WishIT</div>
    </div>

    <div class="form">
        <input type="text" name="username" placeholder="Username"
id="username" required>
        <input type="password" name="password" placeholder="Password"
id="password" required>
        <i class="far fa-eye" id="togglePassword"
onclick="passwordVisibility()"></i>

        <input type="button" value="Login" class="btn" onclick="login()">
    </div>
</div>

<div class="sign-up-div">

```

```

    Don't have an account? <a href="<?php echo base_url()."Register/" ?>"
class="sign-up">Sign Up</a>
</div>

<?php
include_once("footer.php");
?>

<script>

    function passwordVisibility() {
        const psw = document.getElementById("password");
        if (psw.type === "password") {
            psw.type = "text";
        } else {
            psw.type = "password";
        }
    }

    var Authentication = Backbone.Model.extend({
        url: "<?php echo base_url().'api/Authentication/authenticate' ?>",
        idAttribute: 'id',
        defaults: {"id": null, "username": "", "password":""}
    });

    var authentication = new Authentication();

    function login() {
        let username = document.getElementById('username').value;
        let password = document.getElementById('password').value;

        if (username === "" || password === "") {
            alert('Please fill both fields.');
```

```
</body>  
</html>
```

#### 4.4.1. Login page CSS (login.css)

```
body {  
  background: radial-gradient(circle, #b537c4e0, #2c0241);  
  font-family: serif;  
}  
  
.login-box {  
  position: relative;  
  border-radius: 5px;  
  background-color: #0a0009cc;  
  width: 70%;  
  padding: 2% 2% 4% 2%;  
  margin: 3% 15% 1% 15%;  
}  
  
.page-name {  
  color: white;  
  display: flex;  
}  
  
.logo {  
  width: 100px;  
  margin: auto 0 auto auto;  
}  
  
.name {  
  font-size: 500%;  
  font-weight: bold;  
  margin: auto auto auto 0;  
}  
  
.form {  
  width: 60%;  
  margin: auto;  
}  
  
input, .btn {  
  width: 100%;  
  padding: 3%;  
  border: none;  
  border-radius: 5px;  
  margin: 2% 0;  
  font-size: 20px;  
}  
  
.btn {  
  color: white;  
  font-weight: bold;  
  border: solid white;  
}
```

```
.btn:hover {
  color: white;
  background-color: #b537c4e0;
}

.form i {
  margin-left: -50px;
  cursor: pointer;
  display: inline;
  font-size: 20;
}

.sign-up-div {
  text-align: center;
  font-size: 20px;
  font-weight: bolder;
}

.sign-up-div a{
  text-decoration: none;
  cursor: pointer;
}

.sign-up-div a:hover{
  color: white;
}

.sign-up {
  color: #007bff;
}
```

#### 4.5. Wish list (wishList.php)

```
<html lang="en">
<head>
  <title>WishIT | My List</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="shortcut icon" type="image/x-icon"
href="/WishIT/images/favicon.ico" />
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link rel="stylesheet" href="/WishIT/css/bootstrap.min.css">
  <link href="/WishIT/css/wishList.css" rel="stylesheet">

  <script src="/WishIT/js/jquery-3.5.1.min.js"></script>
  <script src="/WishIT/js/popper.min.js"></script>
  <script src="/WishIT/js/bootstrap.min.js"></script>
  <script src="/WishIT/js/underscore.js" type="text/javascript"></script>
  <script src="/WishIT/js/backbone.js" type="text/javascript"></script>
</head>

<body>
```

```

<ul class="topnav">
  <li>
    <div class="page-name">
      <div class="name">WishIT</div>
    </div>
  </li>

  <li class="right">
    <a href="javascript:void(0)" class="dropbtn" id="loggedInUser"></a>
    <div class="dropdown-content">
      <a href="<?php echo base_url(). 'api/Authentication/logout' ?>"
id="logout">Logout</a>
    </div>
  </li>
</ul>

<input type="hidden" id="base" value="<?php echo base_url(); ?>">

<div class="no-list" id="noListView">
  <div class="no-list-msg">Create your Wish List</div>
  <button class='add-icon' onclick="openList('add')">
    <i class="fa fa-plus-circle" data-toggle="tooltip" data-
placement="bottom" title="Create Wish List"></i>
  </button>
</div>

<div class="list-details" id="listView">
  <div class="list-left" id="listMain"></div>

  <div class="list-right">
    <button class="edit-icon" onclick="openDeleteList()">
      <i class="fa fa-trash" data-toggle="tooltip" data-
placement="bottom" title="Delete List"></i>
    </button>
    <button class="edit-icon" onclick="openList('edit')">
      <i class="fa fa-pencil" data-toggle="tooltip" data-
placement="bottom" title="Edit List"></i>
    </button>
    <button class="share-icon" onclick="shareableLink()">
      <i class="fa fa-share" data-toggle="tooltip" data-
placement="bottom" title="Get Shareable Link"></i>
    </button>
    <button class='add-icon list-add' onclick="openItemPopup('add',
null)">
      <i class="fa fa-plus-circle" data-toggle="tooltip" data-
placement="bottom" title="Add an Item"></i>
    </button>
  </div>
</div>

<div class="popup" id="listPopup">
  <div class="popup-content">
    <span class="close" onclick="closeListPopup()">&times;</span>
    <div class="popup-heading" id="listPopupHeading"></div>

    <label for="name" class="lbl">List Name*</label>

```

```

        <input type="text" placeholder="List Name" name="name" id="name">

        <label for="desc" class="lbl">List Description*</label>
        <textarea name="desc" id="desc" placeholder="List
Description"></textarea>

        <label for="occasion" class="lbl">Occasion*</label>
        <select name="occasion" id="occasion"></select>

        <input type="button" value="Create Wish List" class="popupBtn"
id="addListBtn" onclick="onSubmitList('add')">
        <input type="button" value="Edit Wish List" class="popupBtn"
id="editListBtn" onclick="onSubmitList('edit')">
    </div>
</div>

<div class="popup" id="deleteListPopup">
    <div class="popup-content">
        <span class="close" onclick="closeDeleteListPopup()">&times;</span>
        <div class="popup-heading">Delete The List</div>

        <label for="name" class="del-msg" id="deleteListMsg"></label>

        <input type="button" value="Yes, delete this list" class="popupBtn"
onclick="onClickDeleteList()">
    </div>
</div>

<div class="no-list" id="noItemView">
    <div class="no-list-msg">Your wish list is empty. Add your first item by
clicking the "Plus" icon.</div>
</div>

<div class="items" id="wishItems"></div>

<div class="popup" id="itemPopup">
    <div class="popup-content">
        <span class="close" onclick="closeItemPopup()">&times;</span>
        <div class="popup-heading" id="popupHeading"></div>

        <input type="hidden" id="id" name="id" value="">

        <label for="title" class="lbl">Title*</label>
        <input type="text" placeholder="Title" name="title" id="title">

        <label for="itemUrl" class="lbl">Item URL*</label>
        <input type="text" placeholder="Item URL" name="itemUrl"
id="itemUrl">

        <label for="priority" class="lbl">Priority*</label>
        <select name="priority" id="priority"></select>

        <label for="price" class="lbl">Price*</label>
        <input type="number" placeholder="Price" name="price" id="price"
min="1" step="any"
oninput="validity.valid||(value='')">

```

```

        <label for="quantity" class="lbl">Quantity*</label>
        <input type="number" placeholder="Quantity" name="quantity"
id="quantity" min="1"
            oninput="validity.valid||(value='');">

        <input type="button" value="Add an Item" class="popupBtn"
id="addItemBtn" onclick="onSubmitItem('add')">
        <input type="button" value="Edit an Item" class="popupBtn"
id="editItemBtn" onclick="onSubmitItem('edit')">
    </div>
</div>

<div class="popup" id="deletePopup">
    <div class="popup-content">
        <span class="close" onclick="closeDeletePopup()">&times;</span>
        <div class="popup-heading">Delete an Item</div>

        <input type="hidden" id="deleteId" name="deleteId" value="">

        <label for="name" class="del-msg" id="deleteMsg"></label>

        <input type="button" value="Yes, delete this item" class="popupBtn"
onclick="onClickDeleteItem()">
    </div>
</div>

<div class="popup" id="shareablePopup">
    <div class="popup-content">
        <span class="close" onclick="closeShareablePopup()">&times;</span>
        <div class="popup-heading">Your Shareable Link</div>

        <input type="text" value="" id="shareableLink">

        <input type="button" value="Copy to Clipboard" class="popupBtn"
onclick="onClickCopyToClipboard()">
    </div>
</div>

<?php
include_once("footer.php");
?>

<script>
    $(document).ready(function() {
        $('[data-toggle="tooltip"]').tooltip();
    });

    $('#logout').click(function(event) {
        event.preventDefault();
        $.ajax({
            url: $(this).attr('href'),
            success: function(response) {
                window.location = "<?php echo base_url().\"HomePage/\" ?>";
            },
            error: function () {
                alert('Unable to logout. Try again.');
```



```

    });
    return false;
  });

  function divVisibilityChange() {
    if (list.get('id') == null) {
      document.getElementById("listView").style.display = "none";
      document.getElementById("noListView").style.display = "flex";
      document.getElementById("wishItems").style.display = "none";
      document.getElementById("noItemView").style.display = "none";
    } else {
      document.getElementById("listView").style.display = "flex";
      document.getElementById("noListView").style.display = "none";

      if (wishItems.length === 0) {
        document.getElementById("wishItems").style.display = "none";
        document.getElementById("noItemView").style.display = "flex";
      } else {
        document.getElementById("wishItems").style.display = "block";
        document.getElementById("noItemView").style.display = "none";
      }
    }
  }
}

</script>

<script src="/WishIT/js/wishlist-user.js" type="text/javascript"></script>
<script src="/WishIT/js/dropdown.js" type="text/javascript"></script>
<script src="/WishIT/js/wishlist-list.js" type="text/javascript"></script>
<script src="/WishIT/js/wishlist-items.js" type="text/javascript"></script>

</body>
</html>

```

#### 4.5.1. Wish list page CSS (wishlist.css)

```

body {
  background: radial-gradient(circle, #b537c4e0, #2c0241);
  font-family: serif;
  margin: 0;
}

ul.topnav {
  list-style-type: none;
  margin: 0;
  padding: 0 2%;
  overflow: hidden;
  background-color: #0a0009cc;
}

ul.topnav li { float: left; }

ul.topnav li a {
  display: block;

```

```
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

ul.topnav li.right { float: right; }

li a, .dropbtn {
    display: inline-block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

li.dropdown { display: inline-block; }

.dropdown-content {
    display: none;
    position: absolute;
    background-color: black;
    min-width: 135px;
    z-index: 1;
    right: 0;
}

.dropdown-content a {
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
    text-align: left;
    font-weight: bold;
}

.right:hover .dropdown-content { display: block; }

.dropdown-content a:hover {
    background-color: #b537c4e0;
    color: black;
}

@media screen and (max-width: 600px) {
    ul.topnav li.right,
    ul.topnav li {float: none;}
}

.page-name {
    color: white;
    display: flex;
}

.page-name img {
    width: 50px;
    height: 55px;
}
```

```
.name {
  font-size: 250%;
  font-weight: bold;
  margin-top: auto;
  margin-bottom: auto;
}

.user-name {
  font-size: 125%;
  display: flex;
  margin: auto;
}

.icon {
  color: white;
  padding: 2% 6%;
}

.no-list {
  margin: 5% auto;
  width: 90%;
  display: flex;
  color: white;
  font-size: 350%;
  background-color: #0a0009cc;
  padding: 5%;
  border-radius: 15px;
}

.no-list-msg {
  margin: auto;
}

.no-list .add-icon {
  font-size: 100%;
  padding: 0 5%;
  margin: auto;
}

.popup {
  font-family: inherit;
  display: none;
  position: fixed;
  z-index: 1;
  padding-top: 100px;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  overflow: auto;
  background-color: rgb(0, 0, 0);
  background-color: rgba(0, 0, 0, 0.6);
}

.popup-content {
  background-color: #fefefe;
```

```
padding: 20px;
border: 1px solid #888;
width: 60%;
margin: auto;
max-height: 90%;
overflow: auto;
}

.popup-heading {
  font-weight: bold;
  font-size: 150%;
  margin: 0 0 1% 0;
}

.lbl {
  font-weight: bold;
}

.del-msg {
  font-size: x-large;
  text-align: center;
  margin: 5% auto;
  width: 100%;
}

input[type=text], input[type=number], textarea, .btn, select, option {
  width: 100%;
  padding: 1% 2%;
  margin: 0 0 4% 0;
  border: 1px solid;
  border-radius: 5px;
  background-color: #dddddd;
}

input[type=text]:focus, input[type=number]:focus, textarea:focus,
select:focus, option:focus {
  background-color: #e6baed;
  outline: none;
}

.popupBtn {
  color: white;
  font-weight: bold;
  padding: 1%;
  background-color: #620992;
  border-radius: 5px;
  width: 100%;
}

.popupBtn:hover {
  color: white;
  background-color: #b537c4e0;
}

.close {
  color: #aaaaaa;
  float: right;
}
```

```
    font-size: 28px;
    font-weight: bold;
}

.close:hover,
.close:focus {
    color: #000;
    text-decoration: none;
    cursor: pointer;
}

.list-details {
    color: white;
    display: flex;
    margin: 1% 5%;
}

.list-left {
    float: left;
    width: 65%;
}

.list-heading {
    display: flex;
    width: 100%;
}

.list-name, .vl, .list-note, .edit-icon, .share-icon { margin: auto 4% auto 1%; }

.list-name {
    font-size: 250%;
    font-weight: bold;
    width: 25%;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
}

.vl {
    font-size: 200%;
    font-weight: bold;
    font-family: cursive;
}

.occasion {
    font-size: 150%;
    width: 25%;
    margin: auto 0;
    font-weight: bold;
}

.list-note {
    display: block;
    font-size: 125%;
    width: 100%;
    white-space: nowrap;
}
```

```
        overflow: hidden;
        text-overflow: ellipsis;
        margin-top: 1%;
    }

    .edit-icon, .add-icon, .item-icon, .share-icon {
        font-size: 175%;
        background: none;
        border: none;
        color: white;
    }

    .list-add {
        font-size: 350%;
        margin-left: 10%;
    }

    .list-right {
        display: flex;
        float: right;
        width: 35%;
        justify-content: space-between;
    }

    .add-icon {
        font-size: 300%;
    }

    .items {
        margin: 3% auto 10% auto;
    }

    .item-card {
        background-color: #e8dfec4;
        margin: 1% 5%;
        padding: 1% 2%;
        border-radius: 5px;
        font-weight: bold;
        border-color: floralwhite;
        border-style: outset;
        border-width: thick;
    }

    .card-top {
        margin: 0 0 1% 0;
        display: flex;
    }

    .item-name {
        font-size: 150%;
        margin-right: 2%;
        color: #55176f;
        width: 15%;
        white-space: nowrap;
        overflow: hidden;
        text-overflow: ellipsis;
    }
```

```

.item-url {
  margin: auto;
  width: 60%;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}

.item-url a {
  color: #0e1dcc;
}

.item-detail {
  margin: auto 5% auto auto;
  background-color: #4f4b5085;
  padding: 1%;
  border-radius: 5px;
}

.item-lbl {
  font-size: 120%;
  margin-right: 1%;
}

.checked {
  margin: auto 1%;
}

.item-icon {
  float: right;
  color: #55176f;
  margin: 0 0 0 4%;
}

```

#### 4.5.2. Wish list user data handling JavaScript (wishlist-user.js)

```

var base_url = $('#base').val();

var User = Backbone.Model.extend({
  url: base_url + "api/User/users",
  idAttribute: 'id',
  defaults: {"id": null, "fname": "", "lname": ""}
});

var user = new User();

var UserView = Backbone.View.extend({
  el: '#loggedInUser',
  initialize : function () {
    this.listenTo(this.model, 'sync', this.render);
    this.model.fetch({async:false});
  },
  render : function () {

```

```

        var html = '<div class="user-name">' + this.model.get('fname') +
            '<i class="fa fa-sort-down icon"></i></div>';
        this.$el.html(html);
    }
});

var userView = new UserView({model:user});

```

#### 4.5.3. Wish list dropdown data handling JavaScript (dropdown.js)

```

var base_url = $('#base').val();

var Occasion = Backbone.Model.extend({
    url: base_url + "api/Occasion/occasion",
    idAttribute: 'id',
    defaults: {"id": null, "name": ""}
});

var Occasions = Backbone.Collection.extend({
    model: Occasion,
    url: base_url + "api/Occasion/occasion"
});

var occasionList = new Occasions();

var OccasionView = Backbone.View.extend({
    el: '#occasion',
    initialize : function () {
        this.listenTo(this.model, 'sync', this.render);
        this.model.fetch({async:false});
    },
    render : function () {
        var html = '';
        this.model.each(function (occasion) {
            html += '<option value="' + occasion.get('id') + '">' +
                occasion.get('name') + '</option>';
        });
        this.$el.html(html);
    }
});

var occasionView = new OccasionView({model:occasionList});

var Priority = Backbone.Model.extend({
    url: base_url + "api/Priority/priority",
    idAttribute: 'id',
    defaults: {"id": null, "priority": null, "name": ""}
});

var Priorities = Backbone.Collection.extend({
    model: Priority,
    url: base_url + "api/Priority/priority"
});

```



```

var priorityList = new Priorities();

var PriorityView = Backbone.View.extend({
  el: '#priority',
  initialize : function () {
    this.listenTo(this.model, 'sync', this.render);
    this.model.fetch({async:false});
  },
  render : function () {
    var html = '';
    this.model.each(function (priority) {
      html += '<option value="' + priority.get('id') + '>' +
priority.get('name') + '</option>';
    });
    this.$el.html(html);
  }
});

var priorityView = new PriorityView({model:priorityList});

```

#### 4.5.4. Wish list basic data handling JavaScript (wishlist-list.js)

```

var base_url = $('#base').val();

var List = Backbone.Model.extend({
  url: base_url + "api/ListDetails/list",
  idAttribute: 'id',
  defaults: {"id": null, "name": "", "description": "", "occasionId": null,
    "occasion":"","userId": null}
});

var list = new List();

var ListView = Backbone.View.extend({
  el: '#listMain',
  initialize: function () {
    this.listenTo(this.model, 'sync remove', this.render);
    this.model.fetch({async:false});
    divVisibilityChange();
  },
  render: function () {
    divVisibilityChange();

    var html = '<div class="list-heading">' +
      '<div class="list-name" data-toggle="tooltip" data-
placement="bottom" title="' +
      this.model.get('name') + '>' + this.model.get('name') + '</div>'
+
      '<div class="vl">|</div>' +
      '<div class="occasion">' + this.model.get('occasion') +
'</div></div>' +
      '<div class="list-note" data-toggle="tooltip" data-
placement="bottom" title="' +
      this.model.get('description') + '>' +

```

```

this.model.get('description') + '</div>';

    this.$el.html(html);
    return this;
},
wait: true
});

var listView = new ListView({model:list});

var listPopup = document.getElementById("listPopup");

function clearListPopUp() {
    document.getElementById("name").value = '';
    document.getElementById("desc").value = '';
    document.getElementById("occasion").value = 1;
}

function closeListPopup() {
    clearListPopUp();
    listPopup.style.display = "none";
}

function openList(action) {
    if (action === 'add') {
        document.getElementById("listPopupHeading").innerHTML = 'Create Wish
List';
        document.getElementById("addListBtn").style.display = 'block';
        document.getElementById("editListBtn").style.display = 'none';
        listPopup.style.display = "block";
    } else {
        document.getElementById("listPopupHeading").innerHTML = 'Edit Wish
List';
        document.getElementById("addListBtn").style.display = 'none';
        document.getElementById("editListBtn").style.display = 'block';
        document.getElementById("name").value = list.get('name');
        document.getElementById("desc").value = list.get('description');
        document.getElementById("occasion").value = list.get('occasionId');
        listPopup.style.display = "block";
    }
}

function onSubmitList(action) {
    var name = document.getElementById("name").value;
    var desc = document.getElementById("desc").value;
    var occasion = document.getElementById("occasion").value;

    if (name !== '' && desc !== '') {
        if (action === 'add') {
            list.set('id', null);
        }
        list.set('name', name);
        list.set('description', desc);
        list.set('occasionId', occasion);
        list.set('userId', user.get('id'));
        list.save(null, { async: false,
            success: function () {

```

```

        listPopup.style.display = "none";
        clearListPopUp();
    },
    error: function () {
        if (action === 'add') {
            alert('Error occurred while creating the list.');
```

```
        } else {
            alert('Error occurred while editing the list.');
```

```
        }
    }
});

} else {
    alert('Please fill all the required fields.');
```

```

}

var listDeletePopup = document.getElementById("deleteListPopup");
```

```

function closeDeleteListPopup() {
    listDeletePopup.style.display = "none";
}

```

```

function openDeleteList() {
    document.getElementById("deleteListMsg").innerHTML = "Are you sure ? <br>
Your wish list '" +
        list.get('name') + "' and related items will be deleted. ";
    listDeletePopup.style.display = "block";
}

```

```

function onClickDeleteList() {
    list.destroy({ url: base_url + "api/ListDetails/list/id/" +
list.get('id'),
    success: function () {
        list.clear().set(list.defaults);
        wishItems.reset();
        divVisibilityChange();
        listDeletePopup.style.display = "none";
    },
    error: function () {
        alert('Error occurred while deleting the list.');
```

```

    }
});

var shareablePopup = document.getElementById("shareablePopup");
```

```

function closeShareablePopup() {
    shareablePopup.style.display = "none";
}

```

```

function shareableLink() {
    $.ajax({
        url: base_url + "api/ListDetails/shareableLink",
        method: "GET",
        success: function(response) {
            document.getElementById("shareableLink").value = response;
```

```

        shareablePopup.style.display = "block";
    },
    error: function () {
        alert('Error occurred while getting shareable link.');
```

```

    }
});
}

function onClickCopyToClipboard() {
    var copiedText = document.getElementById("shareableLink");
    copiedText.select();
    copiedText.setSelectionRange(0, 99999)
    document.execCommand("copy");
}

```

#### 4.5.5. Wish list items data handling JavaScript (wishlist-items.js)

```

var base_url = $('#base').val();

var WishItem = Backbone.Model.extend({
    url: base_url + "api/WishItem/wishItems",
    idAttribute: 'id',
    defaults: {"id": null, "title": "", "listId": null, "priorityId": null,
    "itemUrl": "",
        "price": null, "quantity": null, "priorityLvl": null, "priority": ""}
});

var wish = new WishItem();

var WishItems = Backbone.Collection.extend({
    model: WishItem,
    comparator: 'priorityLvl',
    url: base_url + "api/WishItem/wishItems"
});

var wishItems = new WishItems();

var WishItemsView = Backbone.View.extend({
    el: '#wishItems',
    initialize : function () {
        this.listenTo(this.model, 'sync remove', this.render);
        this.model.fetch({async:false});
        divVisibilityChange();
    },
    render : function () {
        divVisibilityChange();

        var html = '';
        this.model.each(function (item) {
            html += '<div class="item-card">' +
                '<div class="card-top">' +
                '<div class="item-name" data-toggle="tooltip" data-
placement="bottom" title="' +
item.get('title') + '>' + item.get('title') + '</div>' +

```

```

        '<div class="item-url"><a href="' + item.get('itemUrl') + '"
target="_blank">' +
        item.get('itemUrl') + '</a></div>' +
        '<button class="item-icon" onclick="openDeleteItemPopup(' +
item.get('id') + ')">' +
        '<i class="fa fa-trash" data-toggle="tooltip" data-
placement="bottom" title="Delete Item"></i>' +
        '</button>' +
        '<button class="item-icon" onclick="openItemPopup(\''edit\'', '
+ item.get('id') + ')">' +
        '<i class="fa fa-pencil" data-toggle="tooltip" data-
placement="bottom" title="Edit Item"></i>' +
        '</button>' +
        '</div>' +
        '<div class="card-bottom">' +
        '<span class="item-detail"><span class="item-lbl">Priority :
</span>' +
        '<span class="item-lbl-detail">' + item.get('priority') +
        '</span>';
        if (item.get('priorityLvl') == 1) {
            html += '<span class="fa fa-star checked" style="color:
#bf0a0a"></span>' +
                '<span class="fa fa-star checked" style="color:
#bf0a0a"></span>' +
                '<span class="fa fa-star checked" style="color:
#bf0a0a"></span>';
        } else if (item.get('priorityLvl') == 2) {
            html += '<span class="fa fa-star checked" style="color:
orange"></span>' +
                '<span class="fa fa-star checked" style="color:
orange"></span>' +
                '<span class="fa fa-star checked"></span>';
        } else {
            html += '<span class="fa fa-star checked" style="color:
yellow"></span>' +
                '<span class="fa fa-star checked"></span>' +
                '<span class="fa fa-star checked"></span>';
        }
        html += '</span>' +
        '<span class="item-detail"><span class="item-lbl">Price :
</span>' +
        '<span class="item-lbl-detail">' + item.get('price') +
        '</span>' +
        '<span class="item-detail"><span class="item-lbl">Quantity :
</span>' +
        '<span class="item-lbl-detail">' + item.get('quantity') +
        '</span>' +
        '</div>' +
        '</div>';
    });
    $('#wishItems').html(html);
}
});

```

```

var wishItemsView = new WishItemsView({model:wishItems});

var itemPopup = document.getElementById("itemPopup");

function clearItemPopUp() {
    document.getElementById("title").value = '';
    document.getElementById("itemUrl").value = '';
    document.getElementById("price").value = '';
    document.getElementById("quantity").value = '';
    document.getElementById("id").value = '';
    document.getElementById("priority").value = 1;
}

function closeItemPopup() {
    clearItemPopUp();
    itemPopup.style.display = "none";
}

function openItemPopup(action, id) {
    if (action === 'add') {
        document.getElementById("popupHeading").innerHTML = 'Add an Item';
        document.getElementById("addItemBtn").style.display = 'block';
        document.getElementById("editItemBtn").style.display = 'none';
        itemPopup.style.display = "block";
    } else {
        document.getElementById("popupHeading").innerHTML = 'Edit an Item';
        document.getElementById("addItemBtn").style.display = 'none';
        document.getElementById("editItemBtn").style.display = 'block';
        document.getElementById("id").value = id;
        itemPopup.style.display = "block";

        var selectedItem = wishItems.get(id);
        document.getElementById("title").value = selectedItem.get('title');
        document.getElementById("itemUrl").value =
selectedItem.get('itemUrl');
        document.getElementById("priority").value =
selectedItem.get('priorityId');
        document.getElementById("price").value = selectedItem.get('price');
        document.getElementById("quantity").value =
selectedItem.get('quantity');
    }
}

function onSubmitItem(action) {
    var title = document.getElementById("title").value;
    var itemUrl = document.getElementById("itemUrl").value;
    var priority = document.getElementById("priority").value;
    var price = document.getElementById("price").value;
    var quantity = document.getElementById("quantity").value;

    var urlRegex = /^((http[s]?:\//){0,1}(www\.){0,1}[a-zA-Z0-9\.\-]+\.[a-zA-Z]{2,5}[\.\-]{0,1})$/;

    if (title !== '' && itemUrl !== '' && price !== '' && quantity !== '') {
        if (urlRegex.test(itemUrl)) {
            var item;
            if (action === 'add') {

```

```

        item = new WishItem();
    } else {
        item = wishItems.get(document.getElementById("id").value);
    }

    item.set('title', title);
    item.set('listId', list.get('id'));
    item.set('priorityId', priority);
    item.set('itemUrl', itemUrl);
    item.set('price', price);
    item.set('quantity', quantity);

    item.save(null, {async:false,
        success: function () {
            wishItems.add(item);
            wishItems.sort();
            itemPopup.style.display = "none";
            clearItemPopUp();
        },
        error: function () {
            if (action === "add") {
                alert('Error occurred while adding the new item.');
```

```

            } else {
                alert('Error occurred while editing the item.');
```

```

        }
    });
}
```

```

    } else {
        alert('Please provide a valid URL.');
```

```

    } else {
        alert('Please fill all the required fields.');
```

```

var deletePopup = document.getElementById("deletePopup");
```

```

function closeDeletePopup() {
    deletePopup.style.display = "none";
}
```

```

function openDeleteItemPopup(id) {
    var item = wishItems.get(id);
    document.getElementById("deleteMsg").innerHTML = "Are you sure ? <br> The  
wish list item '" +  
    item.get('title') + "' will be removed from your wish list.";
    document.getElementById("deleteId").value = id;
    deletePopup.style.display = "block";
}
```

```

function onClickDeleteItem() {
    var item = wishItems.get(document.getElementById("deleteId").value);

    item.destroy({ url: base_url + "api/WishItem/wishItems/id/" +  
item.get('id'),
```

```

    success: function () {
        wishItems.remove(item);
        deletePopup.style.display = "none";
    },
    error: function () {
        alert('Error occurred while deleting the item.');
```

#### 4.6. Shared wish list (sharedWishList.php)

```

<html lang="en">
<head>
    <title>WishIT | Shared List</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="shortcut icon" type="image/x-icon"
href="/WishIT/images/favicon.ico" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="/WishIT/css/bootstrap.min.css">
    <link href="/WishIT/css/sharedWishList.css" rel="stylesheet">

    <script src="/WishIT/js/jquery-3.5.1.min.js"></script>
    <script src="/WishIT/js/popper.min.js"></script>
    <script src="/WishIT/js/bootstrap.min.js"></script>
    <script src="/WishIT/js/underscore.js" type="text/javascript"></script>
    <script src="/WishIT/js/backbone.js" type="text/javascript"></script>
</head>

<body>

<div class="nav-bar">
    <div class="page-name">
        <div class="name">WishIT</div>
    </div>

    <div class="username" id="username"></div>
</div>

<div class="list-details">
    <div class="list-left" id="listDetails"></div>
</div>

<div id="wishItems"></div>

<?php
include_once("footer.php");
?>

<script>
    $(document).ready(function(){
        $('[data-toggle="tooltip"]').tooltip();
```



```

});

var User = Backbone.Model.extend({
  url: "<?php echo base_url().'api/User/users' ?>",
  idAttribute: 'id',
  defaults: {"id": null, "fname": "", "lname": "", "username": ""}
});

var user = new User();
var username = window.location.href.split("/").pop();

var UserView = Backbone.View.extend({
  el: '#username',
  initialize : function () {
    this.model.fetch({data: $.param({"username": username}),
async:false});
    this.render();
  },
  render : function () {
    var html = '<span>' + this.model.get('fname') + ' ' +
this.model.get('lname') + '\s Wish List</span>';
    this.$el.html(html);
  }
});

var userView = new UserView({model:user});

var List = Backbone.Model.extend({
  url: "<?php echo base_url().'api/ListDetails/list' ?>",
  idAttribute: 'id',
  defaults: {"id": null, "name": "", "description": "", "occasionId":
null,
    "occasion": "", "userId": null}
});

var list = new List();

var ListView = Backbone.View.extend({
  el: '#listDetails',
  initialize: function () {
    this.model.fetch({data: $.param({"userId": user.get('id')}),
async:false});
    this.render();
  },
  render: function () {
    var html = '';

    if (this.model.get('id') != null) {
      html = '<div class="list-name" data-toggle="tooltip" data-
placement="bottom" ' +
        'title= "" + this.model.get('name') + "">' +
this.model.get('name') + '</div>' +
        '<div class="vl">|</div>' +
        '<div class="occasion">' + this.model.get('occasion') +
'</div>' +
        '<div class="vl">|</div>' +

```

```

        '<div class="list-note" data-toggle="tooltip" data-
placement="bottom" title="' +
        this.model.get('description') + '>' +
this.model.get('description') +
        '</div>';
    }

    this.$el.html(html);
    return this;
}

});

var listView = new ListView({model:list});

var WishItem = Backbone.Model.extend({
    url: "<?php echo base_url().'api/WishItem/wishItems' ?>",
    idAttribute: 'id',
    defaults: {"id": null, "title": "", "listId": null, "priorityId":
null, "itemUrl": "",
    "price": null, "quantity": null, "priorityLvl": null, "priority":
""}
});

var WishItems = Backbone.Collection.extend({
    model: WishItem,
    comparator: 'priorityLvl',
    url: "<?php echo base_url().'api/WishItem/wishItems' ?>"
});

var wishItems = new WishItems();

var WishItemsView = Backbone.View.extend({
    el: '#wishItems',
    initialize : function () {
        if (list.get('id') != null) {
            this.model.fetch({data: $.param({"listId": list.get('id')}),
async:false});
        }
        this.render();
    },
    render : function () {
        var html = '';
        if (list.get('id') == null ) {
            html += '<div class="no-list">' +
                '<div class="no-list-msg">Currently ' + user.get('fname')
+ ' doesn\'t have a Wish List</div>' +
                '</div>';
        } else if (this.model.length === 0) {
            html += '<div class="no-list">' +
                '<div class="no-list-msg">Currently ' + user.get('fname')
+ '\s Wish List is empty</div>' +
                '</div>';
        } else {
            html += '<div class="items">';
            this.model.each(function (item) {
                html += '<div class="item-card">' +
                    '<div class="card-top">' +

```

```

        '<div class="item-name" data-toggle="tooltip" data-
placement="bottom" title="' +
            item.get('title') + '">' + item.get('title') +
'</div>' +
            '<div class="item-url"><a href="' + item.get('itemUrl') +
'" target="_blank">' +
                item.get('itemUrl') + '</a></div>' +
'</div>' +
            '<div class="card-bottom">' +
                '<span class="item-detail"><span class="item-
lbl">Priority : </span>' +
                    '<span class="item-lbl-detail">' + item.get('priority') +
'</span>';
                if (item.get('priorityLvl') == 1) {
                    html += '<span class="fa fa-star checked"
style="color: #bf0a0a"></span>' +
                        '<span class="fa fa-star checked" style="color:
#bf0a0a"></span>' +
                            '<span class="fa fa-star checked" style="color:
#bf0a0a"></span>';
                } else if (item.get('priorityLvl') == 2) {
                    html += '<span class="fa fa-star checked"
style="color: orange"></span>' +
                        '<span class="fa fa-star checked" style="color:
orange"></span>' +
                            '<span class="fa fa-star checked"></span>';
                } else {
                    html += '<span class="fa fa-star checked"
style="color: yellow"></span>' +
                        '<span class="fa fa-star checked"></span>' +
                            '<span class="fa fa-star checked"></span>';
                }
                html += '</span>' +
                    '<span class="item-detail"><span class="item-lbl">Price :
</span>' +
                        '<span class="item-lbl-detail">' + item.get('price') +
'</span>' +
                            '</span>' +
                                '<span class="item-detail"><span class="item-
lbl">Quantity : </span>' +
                                    '<span class="item-lbl-detail">' + item.get('quantity') +
'</span>' +
                                        '</span>' +
                                            '</div>' +
                                                '</div>';
                    });
                html += '</div>';
            }
            $('#wishItems').html(html);
        });

        var wishItemsView = new WishItemsView({model:wishItems});
</script>

</body>
</html>

```

#### 4.6.1. Share wish list page CSS (sharedWishList.css)

```
body {
  background: radial-gradient(circle, #b537c4e0, #2c0241);
  font-family: serif;
  margin: 0;
}

.nav-bar {
  background-color: #0a0009cc;
  display: flex;
  margin: auto;
}

.page-name {
  color: white;
  display: flex;
  width: 25%;
}

.page-name img {
  width: 50px;
  height: 55px;
}

.name {
  font-size: 250%;
  font-weight: bold;
  margin-top: auto;
  margin-bottom: auto;
}

.username {
  font-size: xx-large;
  color: white;
  margin: auto;
  width: 65%;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;;
}

.list-details {
  color: white;
  display: flex;
  margin: 1% 5%;
}

.list-left {
  display: flex;
  float: left;
  width: 100%;
}
```

```
.list-name {
  font-size: 250%;
  font-weight: bold;
  width: 25%;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  margin: auto 1% auto 0;
}

.v1 {
  font-size: 200%;
  font-weight: bold;
  font-family: cursive;
}

.occasion {
  font-size: 150%;
  width: 25%;
  font-weight: bold;
  margin: auto 1%;
}

.list-note {
  display: block;
  font-size: 125%;
  width: 100%;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  margin: auto 1%;
}

.no-list {
  margin: 5% auto;
  width: 90%;
  display: flex;
  color: white;
  font-size: 350%;
  background-color: #0a0009cc;
  padding: 5%;
  border-radius: 15px;
}

.no-list-msg {
  margin: auto;
}

.items {
  margin: auto auto 10%;
}

.item-card {
  background-color: #e8dfecc4;
  margin: 2% 8%;
  padding: 2% 4%;
  border-radius: 5px;
}
```

```
    font-weight: bold;
    border-color: floralwhite;
    border-style: outset;
    border-width: thick;
}

.card-top {
    margin: 0 auto 1% auto;
    display: flex;
}

.item-name {
    font-size: 150%;
    margin-right: 2%;
    color: #55176f;
    width: 15%;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
}

.item-url {
    margin: auto;
    width: 60%;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
}

.item-url a {
    color: #0e1dcc;
}

.item-detail {
    margin: auto 5% auto auto;
    background-color: #4f4b5085;
    padding: 1%;
    border-radius: 5px;
}

.item-lbl {
    font-size: 120%;
    margin-right: 1%;
}

.checked {
    margin: auto 1%;
}
```

## 5. Controllers

### 5.1. HomePage

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class HomePage extends CI_Controller {

    public function index() {
        $this->load->view('homePage');
    }

    public function myCV(){
        $this->load->view('samashaCV');
    }

}
```

### 5.2. Login

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI_Controller {

    public function index() {
        $this->load->view('login');
    }

}
```

### 5.3. Register

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Register extends CI_Controller {

    public function index() {
        $this->load->view('signUp');
    }

}
```

## 5.4. WishList

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class WishList extends CI_Controller {

    public function myWishList() {
        $this->load->model('UserModel');

        if (!$this->UserModel->isLoggedIn()) {
            $this->load->view('login');
        } else {
            $this->load->view('wishList');
        }
    }

    public function sharedList() {
        $this->load->view('sharedWishList');
    }
}
```

## 6. API Controllers

### 6.1. Authentication

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class Authentication extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();
    }

    public function authenticate_post() {
        $username = $this->post('username');
        $password = $this->post('password');

        $this->load->model('AuthenticateModel');

        $result = $this->AuthenticateModel->authenticate($username,
        $password);

        if ($result == 1) {
            $authenticated = new stdClass();
        }
    }
}
```



```

        $authenticated->username = $username;
        $this->set_response($authenticated,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    } else {
        $this->set_response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }
}

public function logout_get() {
    $this->load->model('AuthenticateModel');

    $this->AuthenticateModel->logout();

    $this->set_response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_OK);
}
}

```

## 6.2. User

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class User extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();

        $this->load->model('UserModel');
    }

    public function users_post() {
        $fname = $this->post('fname');
        $lname = $this->post('lname');
        $username = $this->post('username');
        $password = $this->post('password');

        $this->load->model('UserModel');

        $result = $this->UserModel->registerUser(ucfirst($fname),
ucfirst($lname), $username, $password);

        if ($result != 0) {
            $user = new stdClass();
            $user->id = $result;
            $user->fname = $fname;
            $user->lname = $lname;
            $user->username = $username;
            $this->set_response($user,
\Restserver\Libraries\REST_Controller::HTTP_CREATED);

```

```

    } else {
        $this->set_response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_CONFLICT);
    }
}

public function users_get(){
    $username = $this->get('username');
    $user = new stdClass();

    if ($username == null) {
        if ($this->UserModel->isLoggedIn()) {
            $username = $this->session->user->username;
            $user = $this->session->user;
        }
    } else {
        $resultUser = $this->UserModel->getUserByUsername($username);

        $user->id = $resultUser->id;
        $user->fname = $resultUser->first_name;
        $user->lname = $resultUser->last_name;
    }

    if ($username != null) {
        $this->response($user,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    } else {
        $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }
}
}

```

### 6.3. Occasion

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class Occasion extends \Restserver\Libraries\REST_Controller {

    public function occasion_get() {
        $this->load->model('OccasionModel');
        $occasionList = $this->OccasionModel->getOccasions();
        $this->response($occasionList,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    }
}

```

## 6.4. Priority

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class Priority extends \Restserver\Libraries\REST_Controller {

    public function priority_get() {
        $this->load->model('PriorityModel');
        $priorityList = $this->PriorityModel->getPriorities();
        $this->response($priorityList,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    }
}
```

## 6.5. ListDetails

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class ListDetails extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();

        $this->load->model('UserModel');
        $this->load->model('ListModel');
        $this->load->model('OccasionModel');
    }

    public function list_get() {
        $userId = $this->get('userId');

        if ($userId == null) {
            if ($this->UserModel->isLoggedIn()) {
                $userId = $this->session->user->id;
            }
        }

        if ($userId === NULL) {
            $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
        } else {
            if ($userId <= 0) {
                $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_BAD_REQUEST);
            } else {
                $result = $this->ListModel->getListDetails($userId);
```

```

        $list = new stdClass();
        $list->id = $result->id;
        $list->name = $result->name;
        $list->description = $result->description;
        $list->occasionId = $result->occasion_id;
        $list->occasion = $result->occasion;
        $list->userId = $result->user_id;

        $this->response($list,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    }
}

public function list_post() {
    if ($this->UserModel->isLoggedIn()) {
        $name = $this->post('name');
        $description = $this->post('description');
        $occasionId = $this->post('occasionId');
        $userId = $this->post('userId');

        $listId = $this->ListModel->saveList($name, $description,
$occasionId, $userId);

        $list = new stdClass();
        $list->id = $listId;
        $list->name = $name;
        $list->description = $description;
        $list->occasionId = $occasionId;
        $list->occasion = $this->OccasionModel-
>setOccasionName($occasionId);
        $list->userId = $userId;

        $this->set_response($list,
\Restserver\Libraries\REST_Controller::HTTP_CREATED);
    } else {
        $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }
}

public function list_put() {
    if ($this->UserModel->isLoggedIn()) {
        $id = $this->put('id');
        $name = $this->put('name');
        $description = $this->put('description');
        $occasionId = $this->put('occasionId');
        $userId = $this->put('userId');

        $this->ListModel->updateList($id, $name, $description,
$occasionId);

        $list = new stdClass();
        $list->id = $id;
        $list->name = $name;
        $list->description = $description;

```

```

        $list->occasionId = $occasionId;
        $list->occasion = $this->OccasionModel->
>setOccasionName($occasionId);
        $list->userId = $userId;

        $this->set_response($list,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    } else {
        $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }

}

public function list_delete() {
    if ($this->UserModel->isLoggedIn()) {
        $id = (int) $this->get('id');

        if ($id <= 0) {
            $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_BAD_REQUEST);
        }

        $this->ListModel->deleteListById($id);

        $message = array('id' => $id, 'message' => 'Deleted the
resource');

        $this->set_response($message,
\Restserver\Libraries\REST_Controller::HTTP_NO_CONTENT);
    } else {
        $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }

}

public function shareableLink_get() {
    if ($this->UserModel->isLoggedIn()) {
        $username = $this->session->user->username;
        $link = base_url()."WishList/sharedList/".$username;

        $this->set_response($link,
\Restserver\Libraries\REST_Controller::HTTP_OK);
    } else {
        $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }

}

}

```

## 6.6. WishItem

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class WishItem extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();

        $this->load->model('UserModel');
        $this->load->model('ListModel');
        $this->load->model('WishItemModel');
        $this->load->model('PriorityModel');
    }

    public function wishItems_get() {
        $id = $this->get('id');

        if ($id === NULL) {
            $items = array();

            $listId = $this->get('listId');
            if ($listId == null) {
                if ($this->UserModel->isLoggedIn()) {
                    $userId = $this->session->user->id;

                    $list = $this->ListModel->getListIdBYUserId($userId);
                    $listId = $list->id;
                }
            }

            if (!empty($listId)) {
                $result = $this->WishItemModel->getWishList($listId);

                foreach ($result as $row) {
                    $item = array("id" => $row->id, "title" => $row->title,
                        "listId" => $row->list_id,
                        "priorityId" => $row->priority, "itemUrl" => $row->
                        item_url, "price" => $row->price,
                        "quantity" => $row->quantity, "priorityLvl" => $row->
                        p_level, "priority" => $row->p_name);

                    array_push($items, $item);
                }
            }

            if (!empty($items)) {
                $this->response($items,
                    \Restserver\Libraries\REST_Controller::HTTP_OK);
            } else {
                $this->response(['status' => FALSE, 'message' => 'No wish
                    items were found'],

```

```

        \Restserver\Libraries\REST_Controller::HTTP_NOT_FOUND);
    }

    } else {
        $id = (int) $id;

        if ($id <= 0) {
            $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_BAD_REQUEST);
        } else {
            $result = $this->WishItemModel->getWishItem($id);

            $item = array("id" => $result->id, "title" => $result->title,
"listId" => $result->list_id,
                "priorityId" => $result->priority, "itemUrl" => $result-
>item_url, "price" => $result->price,
                "quantity" => $result->quantity, "priorityLvl" =>
$result->p_level, "priority" => $result->p_name);

            $this->response($item,
\Restserver\Libraries\REST_Controller::HTTP_OK);
        }
    }

    public function wishItems_post() {
        if ($this->UserModel->isLoggedIn()) {
            $title = $this->post('title');
            $listId = $this->post('listId');
            $priorityId = $this->post('priorityId');
            $itemUrl = $this->post('itemUrl');
            $price = $this->post('price');
            $quantity = $this->post('quantity');

            $newItemId = $this->WishItemModel->saveWishItem($title, $listId,
$priorityId, $itemUrl, $price, $quantity);

            $priority = $this->PriorityModel->setPriority($priorityId);

            $newItem = array("id" => $newItemId, "title" => $title, "listId"
=> $listId, "priorityId" => $priorityId,
                "itemUrl" => $itemUrl, "price" => $price, "quantity" =>
$quantity, "priorityLvl" => $priority->priority,
                "priority" => $priority->name);

            $this->set_response($newItem,
\Restserver\Libraries\REST_Controller::HTTP_CREATED);
        } else {
            $this->response(NULL,
\Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
        }
    }

    public function wishItems_put() {
        if ($this->UserModel->isLoggedIn()) {
            $id = $this->put('id');

```

```

        $title = $this->put('title');
        $priorityId = $this->put('priorityId');
        $itemUrl = $this->put('itemUrl');
        $price = $this->put('price');
        $quantity = $this->put('quantity');

        $this->WishItemModel->updateWishItem($id, $title, $priorityId,
        $itemUrl, $price, $quantity);

        $priority = $this->PriorityModel->setPriority($priorityId);

        $item = array("id" => $id, "title" => $title, "priorityId" =>
        $priorityId, "itemUrl" => $itemUrl,
                "price" => $price, "quantity" => $quantity, "priorityLvl" =>
        $priority->priority,
                "priority" => $priority->name);

        $this->set_response($item,
        \Restserver\Libraries\REST_Controller::HTTP_OK);
    } else {
        $this->response(NULL,
        \Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }
}

public function wishItems_delete() {
    if ($this->UserModel->isLoggedIn()) {
        $id = (int) $this->get('id');

        if ($id <= 0) {
            $this->response(NULL,
            \Restserver\Libraries\REST_Controller::HTTP_BAD_REQUEST);
        }

        $this->WishItemModel->deleteWishItem($id);

        $message = array('id' => $id, 'message' => 'Deleted the
        resource');

        $this->set_response($message,
        \Restserver\Libraries\REST_Controller::HTTP_NO_CONTENT);
    } else {
        $this->response(NULL,
        \Restserver\Libraries\REST_Controller::HTTP_UNAUTHORIZED);
    }
}
}

```



## 7. Models

### 7.1. AuthenticateModel

```
<?php

class AuthenticateModel extends CI_Model {

    public function __construct() {
        $this->load->database();
    }

    public function authenticate($username, $password){
        $query = $this->db->get_where('wi_users', array('username' =>
$username));

        if ($query->num_rows() != 1) {
            return -1;
        } else {
            $row = $query->row();

            $hashPassword = $row->password;

            if (password_verify($password, $hashPassword)) {
                $user = new stdClass();
                $user->id = $row->id;
                $user->fname = $row->first_name;
                $user->lname = $row->last_name;
                $user->username = $row->username;

                $this->session->set_userdata('user', $user);
                return 1;
            } else {
                return 0;
            }
        }
    }

    public function logout() {
        $this->session->unset_userdata('user');
        $this->session->sess_destroy();
    }
}
```

## 7.2. UserModel

```
<?php

class UserModel extends CI_Model {

    public function __construct() {
        $this->load->database();
    }

    public function registerUser($fname, $lname, $username, $password) {
        $query = $this->db->get_where('wi_users', array('username' =>
$username));

        if ($query->num_rows() != 0) {
            return 0;
        } else {
            $hashPassword = password_hash($password, PASSWORD_DEFAULT);
            $data = array('first_name' => $fname, 'last_name' => $lname,
'username' => $username,
            'password' => $hashPassword);

            $this->db->insert('wi_users', $data);

            return $this->db->insert_id();
        }
    }

    function isLoggedIn() {
        if (isset($this->session->user)) {
            return true;
        } else {
            return false;
        }
    }

    function getUserByUsername($username) {
        $query = $this->db->get_where('wi_users', array('username' =>
$username));
        return $query->row();
    }
}
```

## 7.3. OccasionModel

```
<?php

class OccasionModel extends CI_Model{

    public function __construct(){
        $this->load->database();
    }
}
```

```

    }

    function getOccasions() {
        $query = $this->db->get('wi_occasions');
        return $query->result();
    }

    function setOccasionName($id) {
        $occasionName = '';
        $occasionList = $this->getOccasions();

        foreach ($occasionList as $occasion) {
            if ($occasion->id == $id) {
                $occasionName = $occasion->name;
            }
        }
        return $occasionName;
    }
}

```

## 7.4. PriorityModel

```

<?php

class PriorityModel extends CI_Model {

    public function __construct() {
        $this->load->database();
    }

    function getPriorities() {
        $this->db->order_by('priority', 'asc');
        $query = $this->db->get('wi_priorities');
        return $query->result();
    }

    function setPriority($id) {
        $priorityObj = 0;
        $priorityList = $this->getPriorities();

        foreach ($priorityList as $priority) {
            if ($priority->id == $id) {
                $priorityObj = $priority;
            }
        }
        return $priorityObj;
    }
}

```

## 7.5. ListModel

```
<?php

class ListModel extends CI_Model {

    public function __construct() {
        $this->load->database();
    }

    function getListDetails($userId) {
        $this->db->select('l.id, l.name, l.description, l.occasion_id,
l.user_id, o.name occasion');
        $this->db->join('wi_occasions o', 'l.occasion_id = o.id', 'left');
        $query = $this->db->get_where('wi_list_details l', array('user_id' =>
$userId));
        return $query->row();
    }

    function saveList($name, $description, $occasionId, $userId) {
        $list = array('name' => $name, 'description' => $description,
'occasion_id' => $occasionId,
        'user_id' => $userId);
        $this->db->insert('wi_list_details', $list);

        return $this->db->insert_id();
    }

    function updateList($id, $name, $description, $occasionId) {
        $data = array('name' => $name, 'description' => $description,
'occasion_id' => $occasionId);
        $this->db->update('wi_list_details', $data, array('id' => $id));
    }

    function getListIdBYUserId($userId) {
        $this->db->select('id');
        $query = $this->db->get_where('wi_list_details', array('user_id' =>
$userId));
        return $query->row();
    }

    function deleteListById($id) {
        $this->db->delete('wi_list_details', array('id' => $id));
    }

}
```

## 7.6. WishItemModel

```
<?php

class WishItemModel extends CI_Model {

    public function __construct() {
        $this->load->database();
    }

    function getWishList($listId) {
        $this->db->select('wi.id, wi.title, wi.list_id, wi.priority,
wi.item_url, wi.price,
wi.quantity, p.name p_name, p.priority p_level');
        $this->db->join('wi_priorities p', 'wi.priority = p.id', 'left');
        $this->db->order_by("p.priority", "asc");
        $query = $this->db->get_where('wi_wish_items wi', array('list_id' =>
$listId));
        return $query->result();
    }

    function getWishItem($id) {
        $this->db->select('wi.id, wi.title, wi.list_id, wi.priority,
wi.item_url, wi.price,
wi.quantity, p.name p_name, p.priority p_level');
        $this->db->join('wi_priorities p', 'wi.priority = p.id', 'left');
        $query = $this->db->get_where('wi_wish_items wi', array('id' =>
$id));
        return $query->row();
    }

    function saveWishItem($title, $listId, $priorityId, $itemUrl, $price,
$quantity) {
        $newItem = array('title' => $title, 'list_id' => $listId, 'priority'
=> $priorityId,
        'item_url' => $itemUrl, 'price' => $price, 'quantity' =>
$quantity);
        $this->db->insert('wi_wish_items', $newItem);

        return $this->db->insert_id();
    }

    function updateWishItem($id, $title, $priorityId, $itemUrl, $price,
$quantity) {
        $item = array('title' => $title, 'priority' => $priorityId,
'item_url' => $itemUrl,
        'price' => $price, 'quantity' => $quantity);
        $this->db->update('wi_wish_items', $item, array('id' => $id));
    }

    function deleteWishItem($id) {
        $this->db->delete('wi_wish_items', array('id' => $id));
    }
}
```

## 8. Other configurations

### 8.1. Set the base URL

- Set “base\_url” in config.php file.

```
$config['base_url'] =  
'https://w1697763.users.ecs.westminster.ac.uk/WishIT/index.php';
```

- Add url as a helper in autoload.php file to load “base\_url”.

```
$autoload['helper'] = array('url');
```

### 8.2. Set the landing page

- Set the “default\_controller” route in routes.php file to load “homepage” view.

```
$route['default_controller'] = 'homePage';
```

### 8.3. Logging

- Provide a threshold value (in my case I set it to 2 which is the debug log messages) and set the log path in config.php file.

```
$config['log_threshold'] = 2;  
  
$config['log_path'] =  
'/home/student/763/w1697763/public_html/WishIT/application/logs/';
```

### 8.4. Session

- Edit the config.php file as follows (session is stored in the database).

```
$config['sess_driver'] = 'database';  
$config['sess_cookie_name'] = 'ci_session';
```

```
$config['sess_expiration'] = 7200;  
$config['sess_save_path'] = 'wi_ci_sessions';  
$config['sess_match_ip'] = FALSE;  
$config['sess_time_to_update'] = 300;  
$config['sess_regenerate_destroy'] = FALSE;
```

## 8.5. Database

- Edit the database.php file in order to connect to the database.

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'phpmyadmin.ecs.westminster.ac.uk',  
    'username' => 'w1697763',  
    'password' => 'gAQGSHhZxJom',  
    'database' => 'w1697763_0',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),  
    'save_queries' => TRUE  
);
```

## 9. SQL file

```
-- phpMyAdmin SQL Dump  
-- version 4.9.0.1  
-- https://www.phpmyadmin.net/  
--  
-- Host: localhost  
-- Generation Time: Jan 17, 2021 at 02:16 PM  
-- Server version: 5.7.32-0ubuntu0.18.04.1  
-- PHP Version: 7.3.25-1+ubuntu18.04.1+deb.sury.org+1
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";  
SET AUTOCOMMIT = 0;
```

```
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `w1697763_0`
--

-----

--
-- Table structure for table `wi_ci_sessions`
--

CREATE TABLE `wi_ci_sessions` (
  `id` varchar(128) COLLATE utf8_bin NOT NULL,
  `ip_address` varchar(45) COLLATE utf8_bin NOT NULL,
  `timestamp` int(10) NOT NULL DEFAULT '0',
  `data` blob NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

-----

--
-- Table structure for table `wi_list_details`
--

CREATE TABLE `wi_list_details` (
  `id` int(5) NOT NULL,
  `name` varchar(50) COLLATE utf8_bin NOT NULL,
  `description` varchar(250) COLLATE utf8_bin NOT NULL,
  `occasion_id` int(5) NOT NULL,
  `user_id` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

-----

--
-- Table structure for table `wi_occasions`
--

CREATE TABLE `wi_occasions` (
  `id` int(5) NOT NULL,
  `name` varchar(25) COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```



```
--
-- Dumping data for table `wi_occasions`
--

INSERT INTO `wi_occasions` (`id`, `name`) VALUES
(1, 'Birthday'),
(2, 'Christmas'),
(3, 'Wedding'),
(4, 'Baby Shower'),
(5, 'Moving House');

-----

--
-- Table structure for table `wi_priorities`
--

CREATE TABLE `wi_priorities` (
  `id` int(5) NOT NULL,
  `priority` int(5) NOT NULL,
  `name` varchar(25) COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

--
-- Dumping data for table `wi_priorities`
--

INSERT INTO `wi_priorities` (`id`, `priority`, `name`) VALUES
(1, 1, 'I want it'),
(2, 2, 'Good to have'),
(3, 3, 'If you can');

-----

--
-- Table structure for table `wi_users`
--

CREATE TABLE `wi_users` (
  `id` int(5) NOT NULL,
  `first_name` varchar(25) COLLATE utf8_bin NOT NULL,
  `last_name` varchar(25) COLLATE utf8_bin NOT NULL,
  `username` varchar(15) COLLATE utf8_bin NOT NULL,
  `password` varchar(255) COLLATE utf8_bin NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

-----

--
-- Table structure for table `wi_wish_items`
```

```
--

CREATE TABLE `wi_wish_items` (
  `id` int(5) NOT NULL,
  `title` varchar(100) COLLATE utf8_bin NOT NULL,
  `list_id` int(5) NOT NULL,
  `priority` int(5) NOT NULL,
  `item_url` mediumtext COLLATE utf8_bin NOT NULL,
  `price` decimal(10,2) NOT NULL,
  `quantity` int(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `wi_ci_sessions`
--
ALTER TABLE `wi_ci_sessions`
  ADD KEY `ci_sessions_timestamp` (`timestamp`);

--
-- Indexes for table `wi_list_details`
--
ALTER TABLE `wi_list_details`
  ADD PRIMARY KEY (`id`),
  ADD KEY `user_fk` (`user_id`),
  ADD KEY `occasion_fk` (`occasion_id`);

--
-- Indexes for table `wi_occasions`
--
ALTER TABLE `wi_occasions`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `wi_priorities`
--
ALTER TABLE `wi_priorities`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `wi_users`
--
ALTER TABLE `wi_users`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `wi_wish_items`
--
```

```
ALTER TABLE `wi_wish_items`
  ADD PRIMARY KEY (`id`),
  ADD KEY `list_fk` (`list_id`),
  ADD KEY `priority_fk` (`priority`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `wi_list_details`
--
ALTER TABLE `wi_list_details`
  MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=37;

--
-- AUTO_INCREMENT for table `wi_occasions`
--
ALTER TABLE `wi_occasions`
  MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `wi_priorities`
--
ALTER TABLE `wi_priorities`
  MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `wi_users`
--
ALTER TABLE `wi_users`
  MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=32;

--
-- AUTO_INCREMENT for table `wi_wish_items`
--
ALTER TABLE `wi_wish_items`
  MODIFY `id` int(5) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=46;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `wi_list_details`
--
ALTER TABLE `wi_list_details`
  ADD CONSTRAINT `occasion_fk` FOREIGN KEY (`occasion_id`) REFERENCES `wi_occasions`
(`id`),
  ADD CONSTRAINT `user_fk` FOREIGN KEY (`user_id`) REFERENCES `wi_users` (`id`);
```

```
--  
-- Constraints for table `wi_wish_items`  
--  
ALTER TABLE `wi_wish_items`  
  ADD CONSTRAINT `list_fk` FOREIGN KEY (`list_id`) REFERENCES `wi_list_details` (`id`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  ADD CONSTRAINT `priority_fk` FOREIGN KEY (`priority`) REFERENCES `wi_priorities` (`id`) ON  
DELETE CASCADE ON UPDATE CASCADE;  
COMMIT;  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```