**EE4717/IM4717 Web Application Design - Project Report**

**Design project group number**: F37-DG15

**Team members**: Zhang Linghan (IM4717)

Zhang Guobin (EE4717)

**Project Title**: THE SHOEBOX (Online ecommerce portal selling sports shoes)

**Summary of Project**:

The objective is to build an online ecommerce portal selling sports shoes. The web application allows consumers to view and search for products filtered by their preference, add products to shopping cart and place orders. The products are listed with details including images, brand information and choices of styles, sizes etc.

**Table of Contents**

**1. Application Requirements and Specifications**

**2. Functional Requirements and Specifications**

**3. Design of the web application**

3.1 Site Map

3.2 Storyboard

3.3 Wireframe

*3.3.1 Home Page*

*3.3.2 Product List Page*

*3.3.3 Product Detail Page*

*3.3.4 Shopping Cart Page*

3.4 Database

*3.4.1 products Table*

*3.4.2 product\_variants Table*

*3.4.3 users Table*

*3.4.4 cart\_items Table*

3.5 Web application testing plan

**4. Implementation**

4.1 Coding of the web page template

4.2 Coding of the pages

4.3 Coding of the each function

**5. Testing of web application**

5.1 Functionality Testing

5.2 Usability testing

5.3 Interface testing

5.4 Compatibility testing

5.5 Performance testing

**6. Conclusion**

**Appendices**

Appendix : Source Codes

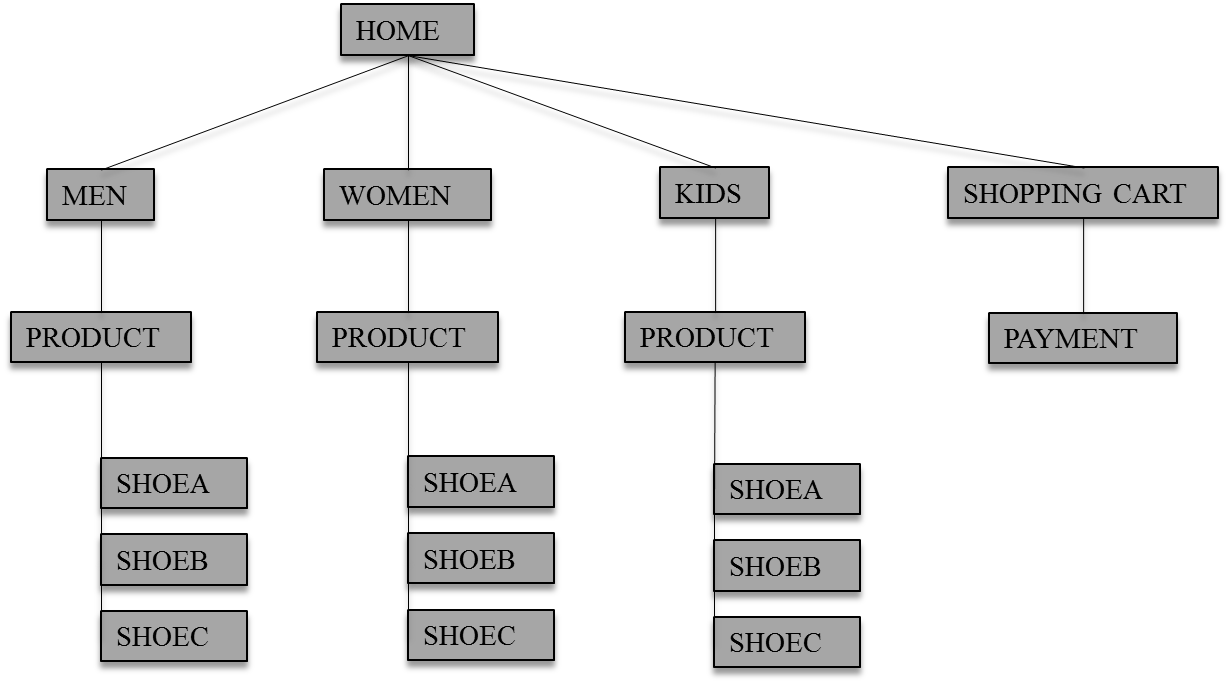
1. **Application Requirements and Specifications**

The application is an online ecommerce portal. The requirements are listed below. The application should:

1. Allow users to view list of products
2. Allow users to filter the products based their preferences.
3. Enable online purchase of products listed in the shop.
4. Provide contact information.
5. **Functional Requirements and Specifications**
6. **Allow users to view list of products**
   1. There will be pages that list all the available product for user. The product listed will have a thumbnail picture of the item and description including items name, brand, unit price etc.
7. **Allow users to filter the products based their preferences.** 
   1. There will be filter one the product list page. User can apply different filter options. The product list should show only items that fits the filter criteria.
   2. Filter can be dynamically applied and product list should be updated accordingly. If no filter is applied by user, all products are shown.
8. **Enable online purchase of products listed in the shop.**
   1. There should be a shopping cart page or component. User can add item to shopping cart while shopping.
   2. User should be able to visit shopping cart page to view all previously selected items with a subtotal.
   3. User should be able to update/delete orders in the shopping cart.
   4. User should be able to checkout if the shopping cart is not empty.
9. **Provide contact information.**
   1. The contact information (phone number, email, location) should be displayed on the page. All page footers should hold include all the above contact information plus social media links.

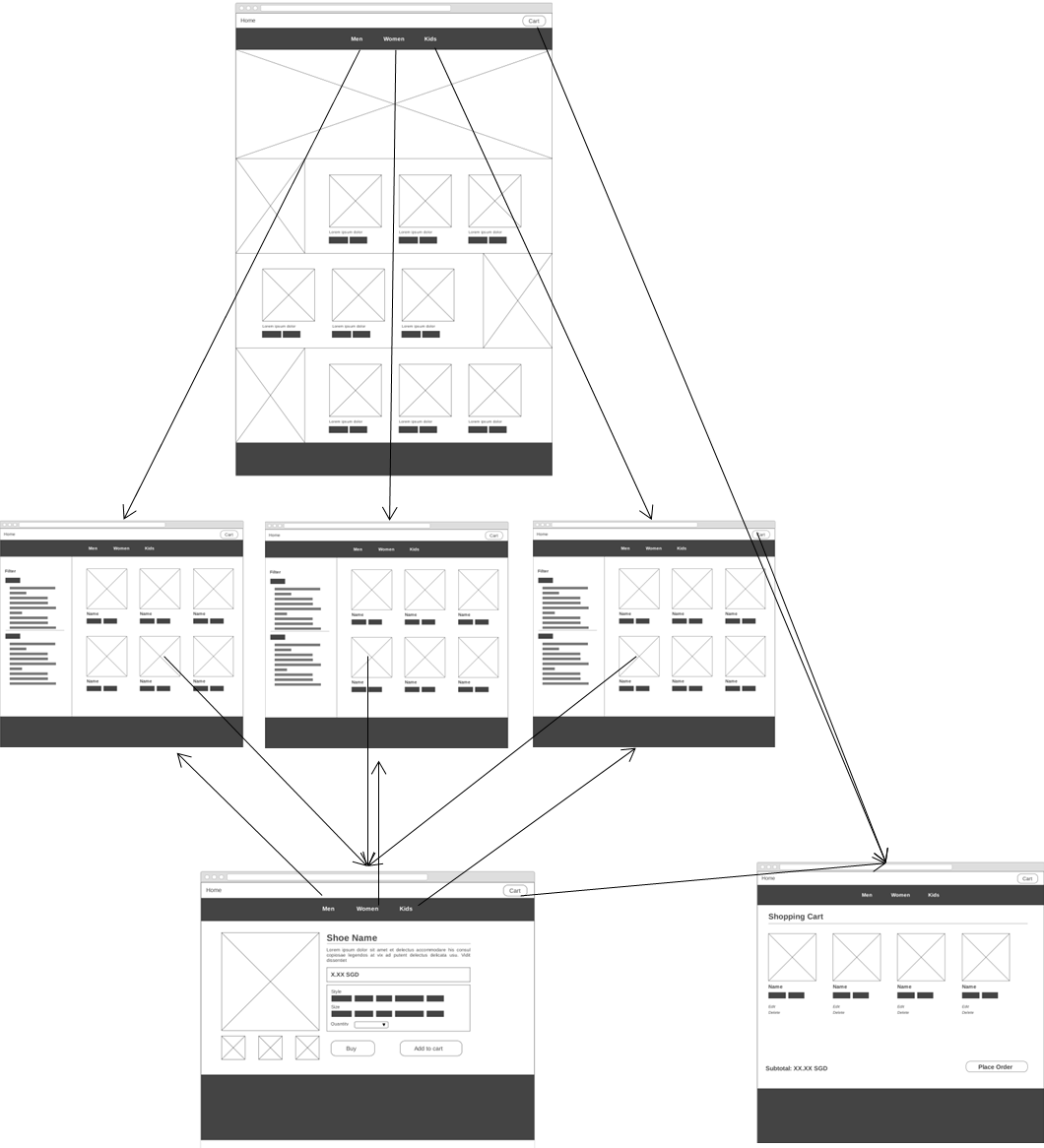
**3. Design of the web application**

**3.1 Site map**



**3.2 Storyboard**

This is the main **home page**. There is a summary of the products. There will be link to different category product list including men, women and kids. iheader



This is the **product list page**. There are more details for the product under each category. Each product links to its matching product formation page.

This is the **product detail page**. All the details for a product were listed inside this page including its name, image, price, description and adding to cart option.

This is the **shopping cart page**. From the shopping cart page, customer can remove/update products in the cart, check out and complete payment easily.

**WOMEN**

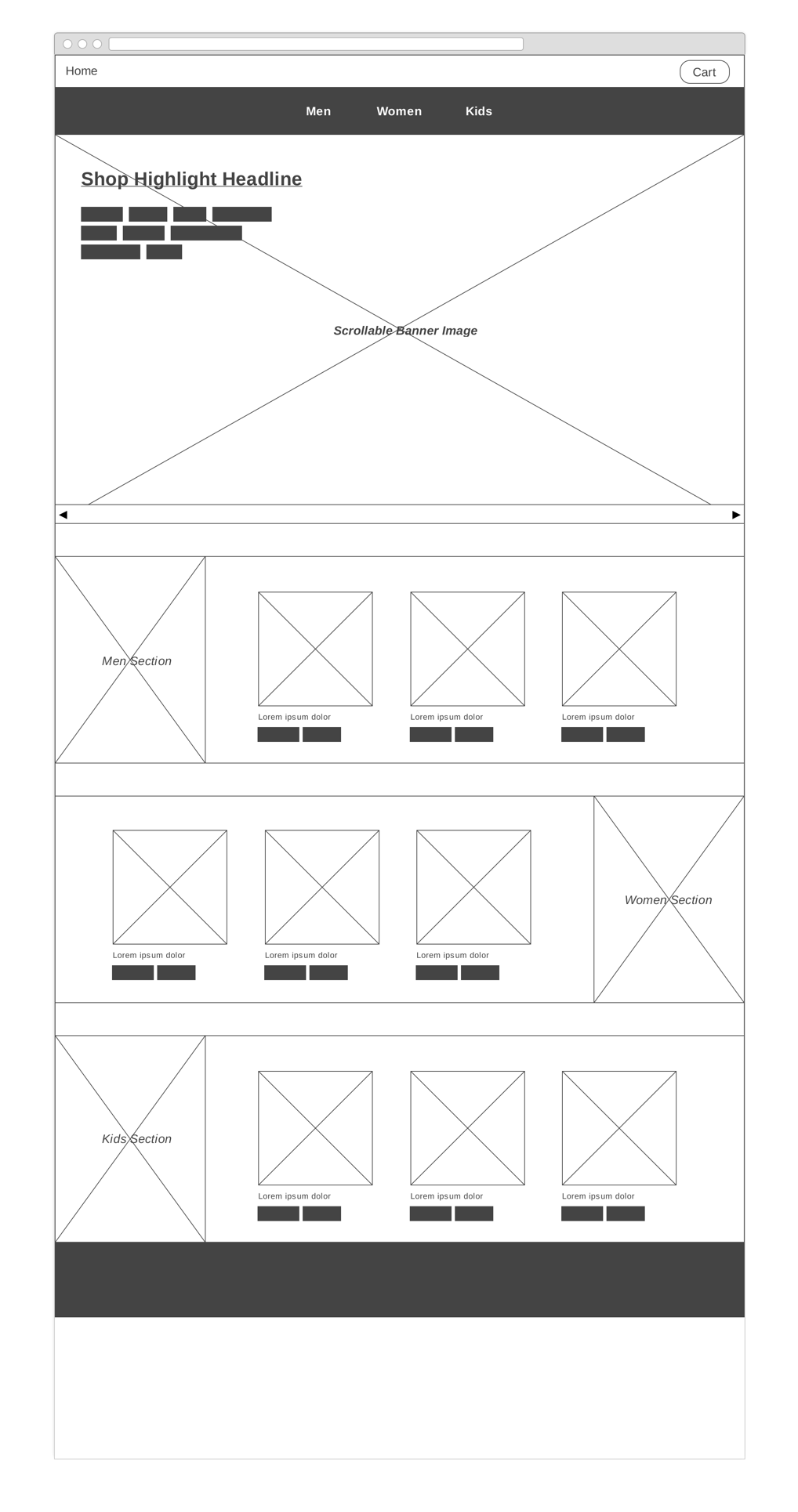
**MEN**

**KIDS**

**3.3 Wireframe**

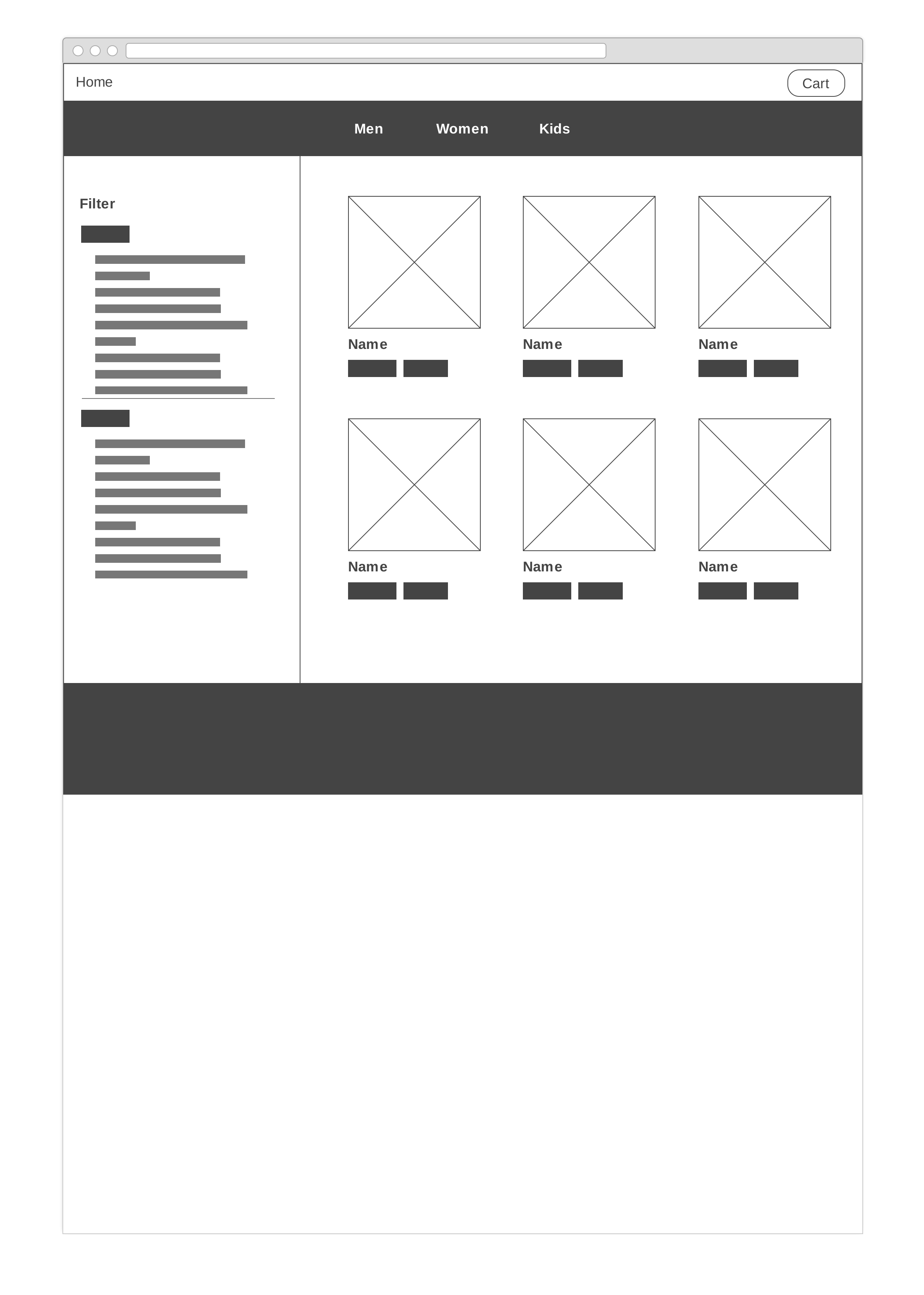
**3.3.1 Home Page**

Page comes with a hero banner slider featuring highlight items, including new arrivals, discounts, feature products. The rest of page lists a summary of 3 main sections (Men, Women, Kids). Featured items for each section are listed for consumer to view.



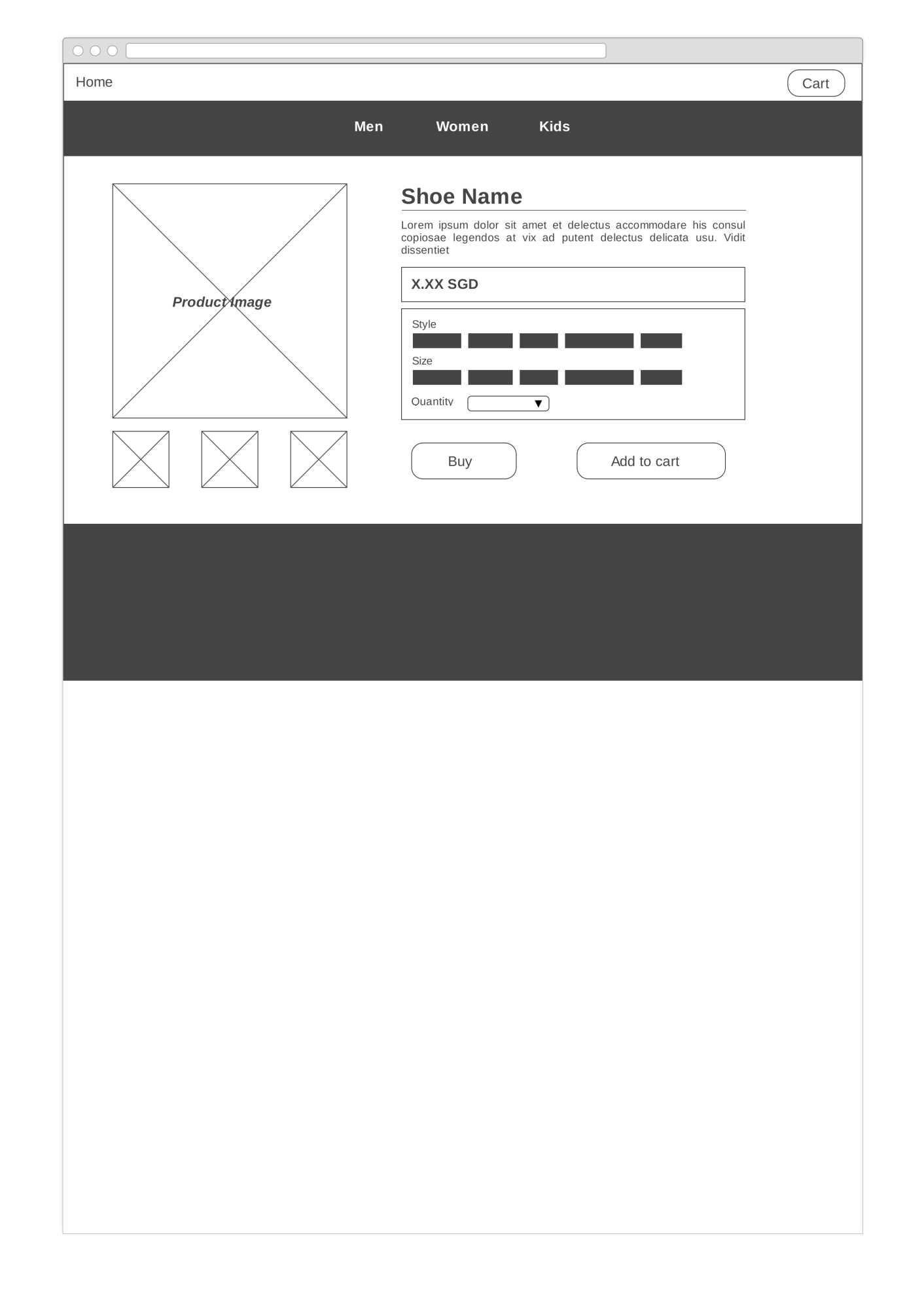
**3.3.2 Product List Page**

Men, women and kids section each has its own product list page with the same layout. The content section follows a tow-column design. Filters with different options are displayed on the left column of the page. On the right column, a scrollable list of product is shown. Based on the filter options, consumers can filter out the preferred product based on brands, colors and etc.



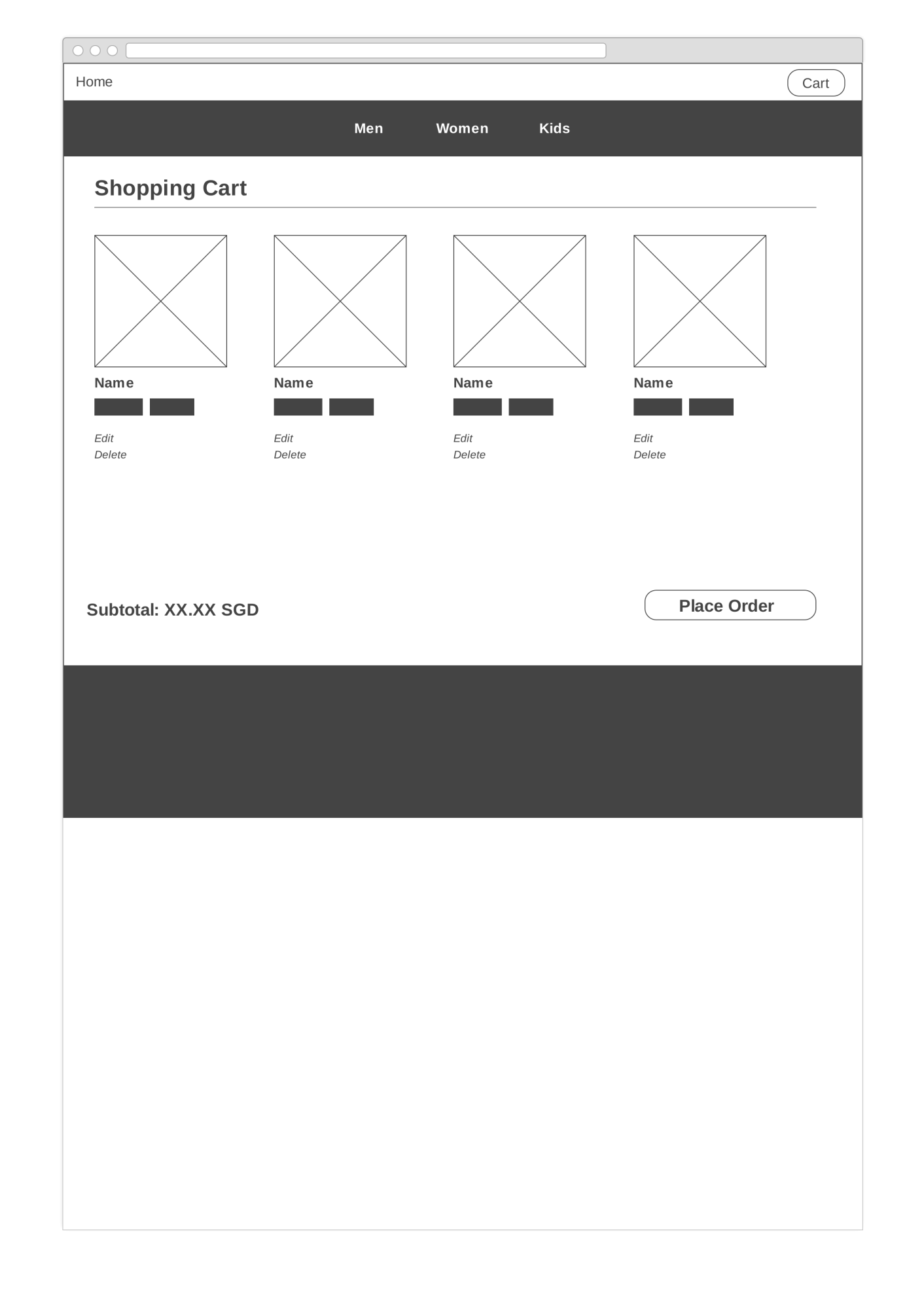
**3.3.3 Product Detail Page**

Product detail page display one single product. User can choose desired quantities, styles, sizes and the order to shopping cart.



**3.3.4 Shopping Cart Page**

All the chosen products and its related information can be found in shipped cart. From the shopping cart page, customer can remove/update products in the cart, check out and complete payment easily.



**3.4. Database**

There are four tables including ‘products’, ‘product\_variants’, ‘users’ and ‘cart\_items’ in the database ‘shoebox’. All the tables have multiple columns and rows.

**3.4.1 Products Table**

All the products displayed in this website were descripted in this table. The details for each product row including product id, product name, description, brand, gender and price. Each row has a primary key (product\_id).

*CREATE TABLE products (*

*product\_id SMALLINT UNSIGNED AUTO\_INCREMENT,*

*product\_name VARCHAR(20) NOT NULL,*

*description VARCHAR(200),*

*brand VARCHAR(20) NOT NULL,*

*gender ENUM('men','women','kids') NOT NULL,*

*price FLOAT(6,2) NOT NULL,*

*CONSTRAINT pk\_product\_id PRIMARY KEY (product\_id)*

*);*

**3.42 product\_variants Table**

The color and size of each product was descripted in a separate table ‘product\_variants’, since each product may have multiple sizes and colors. The details for each product include product variant id, product id, product size and color. Foreign key (product\_id) is also added to this table, which points to the primary key inside product table.

*CREATE TABLE product\_variants (*

*product\_variant\_id SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,*

*product\_id SMALLINT UNSIGNED NOT NULL,*

*size TINYINT UNSIGNED NOT NULL,*

*color VARCHAR(20) NOT NULL,*

*CONSTRAINT fk\_product\_id FOREIGN KEY (product\_id) REFERENCES products (product\_id)*

*);*

**3.43 Users Table**

All the registered user information was stored in the table. The details for each user include user id, name, email address and password. Each row has a primary key (user\_id). The unique constraints are also created to username and email, which is to make sure that every username and email address are unique.

*CREATE TABLE users (*

*user\_id VARCHAR(36) NOT NULL,*

*username VARCHAR(16) NOT NULL,*

*email VARCHAR(50) NOT NULL,*

*password VARCHAR(64) NOT NULL,*

*CONSTRAINT pk\_user\_id PRIMARY KEY (user\_id),*

*CONSTRAINT uc\_username UNIQUE (username),*

*CONSTRAINT uc\_email UNIQUE (email)*

*);*

**3.44 cart\_items Table**

All the items that was chosen and put in cart were descripted in this table. The details for each chosen item include item id, user id, product id, product variant id, quantity and orders. Each row has a primary key (item\_id). There are three foreign keys are added to this table including ‘item\_id’, ‘product\_id’ and ‘product\_variant\_id’, which points to the primary keys inside previous three tables respectively.

*CREATE TABLE cart\_items (*

*item\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,*

*user\_id VARCHAR(36),*

*product\_id SMALLINT UNSIGNED,*

*product\_variant\_id SMALLINT UNSIGNED,*

*quantity TINYINT UNSIGNED,*

*ordered BOOLEAN,*

*CONSTRAINT pk\_item\_id PRIMARY KEY (item\_id),*

*CONSTRAINT fk\_user\_id FOREIGN KEY (user\_id) REFERENCES users(user\_id),*

*CONSTRAINT fk\_product\_id\_cart FOREIGN KEY (product\_id) REFERENCES products(product\_id),*

*CONSTRAINT fk\_product\_variant\_id FOREIGN KEY (product\_variant\_id) REFERENCES product\_variants(product\_variant\_id)*

*);*

**3.5. Web application testing plan**

|  |  |
| --- | --- |
| **Testing** | **Testing content** |
| Functionality | The links, database connection, forms used for submitting or getting information from user in the web pages |
| Usability | Navigation and interface content |
| Interface | Web server, application server and database server interface |
| Compatibility | Browser compatibility and operating system compatibility |
| Performance | Web Load |

**4. Implementation**

4.1 Coding of the web page template

There are basically there parts of the page template including header, main content and footer. All the pages in this project are based on the page template.

1. Header: import the nav.php from partials folder.
2. Main content: all the information that was displayed in the medium. (this part will be discussed in coding of pages)
3. Footer: import the footer.php from partials folder.

|  |  |
| --- | --- |
| **folder** | **content** |
| assets | All the product information including image, product name, description color and size. |
| css | All the CSS code was written in styles.css. |
| includes | There are two php file db.php and session.php, which to provide all the php function to implement cart, login and sign up.   1. db.php: it is to provide all the php function to support cart implementation. 2. session.php: it is to provide all the login and signup function. (This part will be discussed more clear in codes of each function) |
| js | The JavaScript function was written script.js   1. script.js: it is to implement cart window moving animation. (This part will be discussed more clear in codes of each function) |
| partials | There are three repeat used elements in partials fold including cart, footer and navigation, which is repeated used in different pages.   1. cart.php: it is a right column page, which is to show all the products that costumers have chosen. (This part will be discussed more clear in codes of each function) 2. footer.php: There are five social medial icons implemented based on extra CSS library “font-awesome”. When user clicks each icon, it will change the color, shape and jump to each social media page respectively. Following icons, there is detail description and disclosure for this project. 3. nav.php: navigation bar is based on two row structure. The first row is home, login and cart. The second row is MEN, WOMEN and KIDS. The cart button connects with cart the page. There is a cart window moving animation, which will pull out the cart page on the present page. The other buttons will jump to another page respectively. |

4.2 Coding of the pages

There are in total four kinds of pages including home page, shop page (men, women and kids), product page and order page. All the pages in this project are based on the page templates and the only different are main content.

|  |  |
| --- | --- |
| **page** | **Main content** |
| home page | The main content are displayed based on up down structure.   1. An attractive video, which can show this brand’s features. 2. The most product including their image, name and price. |
| shop page (men, women and kids | The main content is displayed based on a tow-column design.   1. Filters with different options are displayed on the left column of the page. 2. On the right column, a scrollable list of product is shown. |
| product page | 1. Product detail page display one single product. |
| order page | 1. Costumer can finish their payment in tis page. |

4.3 Coding of the each function

|  |  |
| --- | --- |
| **function** | **algorithm** |
| cart window | 1. The connection with database ‘cart\_items Table’ needs to be built in each product page. And db.php is imported. If the customer choose the color, quantity and size. All the information will be sent to database ‘cart\_items Table’.   dbAddToCart($\_SESSION["userid"], $product\_id,$\_POST["color"], $\_POST["size"], $\_POST["qty"]);  function dbAddToCart($user\_id, $product\_id, $color, $size, $qty){  $insert\_query = "INSERT INTO cart\_items (user\_id, product\_id , product\_variant\_id, quantity, ordered) VALUES ('$user\_id', $product\_id, (SELECT product\_variant\_id FROM product\_variants WHERE product\_id = $product\_id AND color = '$color' AND size = $size LIMIT 1), $qty, false);";  return query($insert\_query);  }   1. The connection with database ‘cart\_items Table’ needs to be built. And import db.php.   <?php  $hasItem = false;  include\_once("./include/db.php");  $mysqli = db\_connect();  $items = dbGetCartItems($\_SESSION["userid"]);  if($items->num\_rows>0){  $hasItem = true;  }  ?>  function dbGetCartItems($user\_id){  $stmt = "SELECT p.product\_id, p.product\_name, pv.color, pv.size, c.item\_id, c.product\_variant\_id, c.quantity  FROM cart\_items c  INNER JOIN products p  ON c.product\_id = p.product\_id  AND c.user\_id = '$user\_id'  AND c.ordered = false  INNER JOIN product\_variants pv  ON c.product\_variant\_id = pv.product\_variant\_id;  ";  return query($stmt);  }   1. After the user is mated with user id. A while loop is used to display all the ordered products.   <?php  $items = dbGetCartItems($\_SESSION["userid"]);  if($items->num\_rows>0){  while($item = $items->fetch\_assoc()){  echo '  <li>  <div id="cart-item-container" class="container">  <div class="col-xs-12">  <div class="row">  <div class="col-xs-5">  <img src="/assets/products/air-jordan-1-retro-high-flyknit-shoe.jpg" alt="" class="img-fulid" style="max-width:100%;height:auto;">  </div>  <div class="col-xs-7">  <h6 class="text-warning">'.$item["product\_name"].'</h6>  <h6 id="size">Size: '.$item["size"].'</h6>  <h6 id="color">Color: '.$item["color"].'</h6>  <h6 id="qty">Quantity: '.$item["quantity"].'</h5>  <input class="btn btn-secondary" type="button" name="modify" value="Edit" onclick="editItem(\'e\', this)">  </div>  </div>  </div>  </div>  </li>  ';  }  }  ?> |
| cart window moving animation | 1. The event ‘onclick’ is added inside. If the button is clicked, the event ‘onclick’ will be called. Function openNav will start to run. If the button is clicked again, the event ‘onclick’ will be called again. Function closeNav will start to run.   onclick="openNav()" onclick="closeNav()"   1. Extra CSS is used for the content inside following div. Especially the width when the cart window is opened.   <div id="cartSidenav" class="cart-side-nav">.... </div>  .cart-side-nav{  height: 100%; /\* 100% Full-height \*/  width: 300px; /\* 0 width - change this with JavaScript \*/  position: fixed; /\* Stay in place \*/  z-index: 1; /\* Stay on top \*/  …  } |
| Footer social medium icon | There are five social medial icons implemented based on extra CSS library “font-awesome”.   1. Import the CSS library   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">   1. Import each social media class   <a href="#" class="fa fa-facebook"></a> |
| Login and sign up | 1. Login page: The connection with database ‘users Table’ needs to be built in each login page. And ‘session.php’ is imported. Once users inputted their information. It will help them to match the information with ‘users Table’. And if there is any mistake in user name and password input, the remainder will also been returned.     <?php  if(isLoggedIn()){  echo '<h3>You have already logged in. Redirecting back to home page in <span id="counter">3</span> seconds...<br></h3>If not redirecting, <a style="color:blue;" href="/index.php">click here</a>';  header( "refresh: 3;url=index.php" );  }else{  echo '  <form action="/include/session.php" method="post">  <h2>LOG IN</h2>  <div>  <input class="input-txt" type="text" name="username" placeholder="Username">  </div>  <div>  <input class="input-txt" type="password" name="password" placeholder="Password">  </div>';  if (isset($\_GET['login\_error'])){  echo '<div><span style="color:red;">Wrong username or password.</span></div>';  }  echo '<input class="btn" type="submit" value="login" name="login">';  echo '<p>Do not have an account yet? <a class="nav-link" href="/signup.php">Register here.</a></p>';  echo '</form>';  }  ?>   1. Sign up page: The connection with database ‘users Table’ needs to be built in each login page. And ‘session.php’ is imported. Once users inputted their information, ‘users Table’ will update the information. And if there is any mistake in user name and password input, the remainder will also been returned.   <?php  if(isLoggedIn()){  echo '<h3>You have already logged in. Redirecting back to home page in <p id="counter">3</p> seconds...<br></h3>If not redirecting, <a style="color:blue;" href="/index.php">click here</a>';  header( "refresh: 3;url=index.php" );  }else{  echo '  <form action="/include/session.php" method="post" onsubmit="return validateForm()">  <h2>REGISTER WITH US</h2>  <div>  <input id="email" class="input-txt" type="email" name="email" placeholder="Email" required>';  if (isset($\_GET['email\_error'])){  echo ' <p class="warning-txt">Email is already used. <a class="nav-link" href="/login.php">Login instead</a></p>';  }  echo '  <p id="email\_warning" class="warning-txt"></p>  </div>  ';    echo '  <div>  <input id="username" class="input-txt" type="text" name="username" placeholder="Username" required>  <p id="username\_warning" class="warning-txt"></p>';  if (isset($\_GET['user\_error'])){  echo '<p class="warning-txt">Username is already used.</p>';  }  echo '  </div>  ';    echo '  <div>  <input id="password" class="input-txt" type="password" name="password" placeholder="Password" required>  <p id="password\_warning" class="warning-txt"></p>  <input id="confirm\_password" class="input-txt" type="password" name="confirm\_password" placeholder="Confirm password" required>  <p id="confirm\_password\_warning" class="warning-txt"></p>  </div>  ';  echo '<input class="btn" type="submit" value="signup" name="signup">';  if (isset($\_GET['error'])){  echo '<div><p class="warning">Unknown error. Please retry later.</p></div>';  }  echo '<p>Already have an account? <a href="/login.php" class="nav-link">Login here.</a></p>';  echo '</form>';  }  ?> |

**5. Testing of web application**

5.1 Functionality Testing

5.11 Link testing

All the links need to be tested. The links includes:

1. Link jumping to different page, including MEN, WOMEN, KIDS, Home and Cart.
2. Internal links attached inside different product images, which is also used to jump to different pages.
3. Link jumping to different social mediums pages in footer icons.
4. Link to login and Register.

Result: All the links worked well.

5.12 Form testing

Form is used to fetch data from users, pass information to database and interact with users. The forms in this application include:

1. Login form
2. Product filter form

Both need to be test for validation and default value. For login form, there are two situations including previous user login and new user register. Wrong input needs to be test and see the remainder information for both situations. For product filter form, multiple choices, single choice and no choice need to be test.

Result: login form worked well when previous user login and new user register. Product filter form worked well with different choices input.

5.13 Database testing

Data integrity and errors were checked when the database is edited, deleted or modified. The point that we need to check includes all the database queries, data retrieved and also data updated through the interface. There is also some database related testing will be addressed in web load testing below. There are four database tables inside shoebox need to be tested including:

1. products
2. product\_variants
3. users
4. cart\_item

Result: The four database tables inside shoebox worked well.

5.2 Usability Testing

5.21 Navigation testing

Navigation is the buttons to jump to different user pages including MEN, WOMEN, KIDS, Home, Cart and Login. The usability inspection standard includes navigation convenience, navigation instruction and content consistency.

Result: The navigation satisfied usability inspection standard, which includes MEN, WOMEN, KIDS, Home, Cart and Login.

5.22 Content checking

Content checking includes text, image and color checking. Content is gone through several times to make sure it is meaningful, logical, spelled right and understand easily. All the anchor text links should be working properly. Images should be placed properly with proper sizes. The color that we used is also checked based on the common accepted standard such as dark colors annoys the users and should not be used in the site theme.

Result: The content of each page worked well based on text, image and color checking.

5.3 Interface Testing

Interfaces that need to be tested include web server, application server and database server interface. All the interactions between users and these servers were tested. If there is any error message was returned by database or web server, the detail of the error need to be displayed appropriately to the users.

Result: The web server, application server and database server interface worked well.

5.4 Compatibility Testing

5.41 Browser compatibility

The setting and configuration of different browsers were always different. And the application needs to be compatible with different browsers. Therefore, the web application was tested on different popular browsers including Internet explorer, Safari and Chrome.

Result: The web application worked well in Internet explorer, Safari and Chrome.

5.42 Operating systems compatibility

The setting and configuration of different operating systems were also different. There are a few technologies may not work in different operation system such as interface calls with different API. Hence the web application test was conducted on different operating systems including Windows and MAC.

Result: The web application worked well in different operating systems including Windows and MAC.

5.5 Performance testing

5.51 Web Load testing

Web application should sustain to different load. Web load testing was conducted to see the web application performance. Multiple computers were connected to the interface and use the application at the same time. It needs to see whether the application can support multiple users, handle multiple user requests and interact with database.

Result: The web application worked well with multiple users. It succeed to handle multiple user requests and interact with database

**6. Conclusion**

An online ecommerce portal selling sports shoes is successfully implemented. The web application allows consumers to view and search for products filtered by their preference, add products to shopping cart and place orders. The products are listed with details including images, brand information and choices of styles, sizes etc. Customer can remove/update products in the cart, check out and complete payment easily. The test has been done based on application functionality, usability, interface, compatibility and performance.

**Appendices**

Appendix : Source Codes

Contribution to the project by each team member shown in a table.