PROJECT PLAN

Team Members
Biswadeep Mazumder – 2860920 || Kevin Panchal – 2871507

1. Introduction

1.1 Software to be tested

Presta.com is an e-commerce platform that is one of the major e-commerce player in Europe and Latin America. With nearly 300,000 sites already using its software across the globe, PrestaShop sites generated more than 24 billion euros in online sales. This website https://www.prestashop.com/en is built using JavaScript, HTML5, CSS and node.js. This is hosted on cloudfare.com and the icons are fetched from material icon from google.

Unlike other e-commerce websites, Presta allows a user to create a shop online in their portal where user can visit and buy products which are exclusive to that particular shop owner.

Project link: https://github.com/BMayhew/awesome-sites-to-test-on/WebTesting/Presta

1.2 Major functions of the software

Create Products:

- Quickly create products.
- Configure features and values.
- Create packs of products.
- Sell customizable goods by letting your customers upload files.
- Create digital products.
- Attaching files or display additional information on product pages.
- Set a minimal quantity to purchase for some products.
- Choose which message to display for out-of-stock items.

Product Navigation:

- Create categories & subcategories.

Run stock:

- Track the inventory of each product.
- Create suppliers and associate them with products.
- Create manufacturers.
- Get an overview of manufacturers and suppliers with sorting orders and filters.

Configure Your Store:

- Offers payment methods.
- Configure the shipping methods.
- Run geographical delivery zones and apply shipping carriers.
- Determine the pricing of the offered shipping methods.
- Define the maximum dimension of the packages.

Content & Navigation

- Configure the display of the products.
- Organize your products in your store.
- Configure filters, sort order and pagination of products.

Checkout

- Configure the one page checkout
- Display a summary page before the customer validate the cart
- Display & allow your customer to navigate in the checkout process
- Allow your customers to quickly create an account
- Let your customer choose the addresses select shipping & payment methods
- Add a reinsurance text block
- Let customers validate their orders
- Display a confirmation page with related details

2. Tasks

1. Installation of the software.

This is a live project, for development and testing purpose there is a demo website which helps us to monitor the changes and perform testing after any changes before committing the changes to the production code.

2. Set up testing environment

The code is to be downloaded from github and on our local system, we can use visual studio code editor to access the code. For front end all the dependencies will be downloaded automatically when we use the npm start command.
For backend, we need to download and install the correct frameworks of node.js.
For these, more elaborate documentation is available on (https://nodejs.org/en/download/package-manager) but for this project, we need not to follow those rules extensively.

3. Design test cases

A. Functionality Testing:

1. Email verification at time of Sign-Up.
2. User ID and Password verification at time of Sign-In.
3. Forms testing, Input fields and proper validation of fields.
4. Button click events at every page to check for proper Anchor links.
5. Testing to ensure the proper Business Workflow is being followed for the events.
6. HTML and CSS testing for proper orientation of the page at every level.
7. Download of attendance sheet for individual events.

B. Usability Testing:

1. Menus, buttons and links for navigation to other pages to be tested across all the webpages.
2. Images with proper “ALT” text and no spelling mistakes.

C. API integration testing:

Three areas to be tested here are – Application, Web and Database Server.

1. **Application:** Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
2. **Web Server**: Test Web server is handling all application requests without any service denial.
3. **Database Server:** Make sure queries sent to the database give expected results.
4. **Test system response** when **connection between the three layers** (Application, Web and Database) **cannot be established** and appropriate message is shown to the end user.

D. **Compatibity Testing:**

1. **Browser Compatibility Test**: Same website in different browsers will display differently. So web application is tested accordingly so that contents being displayed correctly across browsers(Firefox, Edge, Chrome, Safari etc.), JavaScript, AJAX and authentication is working fine.

2. **Operating System Compatibility Test**: The rendering of web elements like buttons, text fields etc. changes with change in **Operating System**. This makes sure that every element works fine for various combination of Operating systems such as Windows, Linux and Mac.

4. Perform Tests.

Both functional and non-functional test cases are to be tested and will be produced in the subsequent documents and lastly in the final report document.

3. Schedule

Sr. No.	Activity	Status
1	Project finalization	Completed
2	Test plan preparation	Completed
3	Test plan submission	Work in Progress
4	Installation of the software	
5	Set up a testing environment	
7	Design test cases	
8	Perform tests	
9	Summarizing results	
10	Final report preparation	
12	Final submission	