|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2020 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | tildascode | | face-to-face |

|  |  |
| --- | --- |
| Project name | Online E-Commerce Platform for small businesses |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| We live in an era where most of the people almost turned into a hobby online shopping. Moreover, starting an e-commerce business is becoming more and more accessible to everybody. But still most of the existing platforms are too complicated for people’s small businesses or too expensive. The platform solves exactly this problem. It is fairly priced and a small retailer could setup its online shop in less than an hour. In addition to that it allows customers to register and buy the different retailer’s products.  The system will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using ***React Router***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization.  The main user roles (actors in UML) are:  • *Anonymous User* – can view all product listing pages, product detail pages and retailer’s pages.  • *Customer* (extends *User*) – can buy products provided from different retailers and view Order Pages, can create its own retailer’s page where they can publish, manage and sell their products.  • *Administrator* (extends *User*) – can manage homepage and create, edit user data and delete all *Registered Users*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Login** | The *User* login in the system by e-mail and password. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. | *Anonymous User* |
| * 1. **View Product Listing Pages (PLPs)** | The *User* can view all Product Listing Pages, click on Products, filter Products on Category, Price etc., sort Products by Price, Newest etc.  PLPs can be Category Pages or Search Pages. Category Pages can be accessed through the navigation and Search Pages can be accessed via a search through the search bar. | *Anonymous User* |
| * 1. **View Product Details Pages PDPs** | The *User* can view all Product Details Pages, view products’ description, ratings, rate themselves and add products to basket. | *Anonymous User* |
| * 1. **View Cart Page** | The *User* can add products to basket and view them in the Cart Page. The *User* can decrease or increase the quantity of the products in the basket and remove them. | *Anonymous User* |
| * 1. **Place Order** | The *Registered User* can place orders via the Order Page. | *Registered User* |
| * 1. **View Order History Page** | The *Registered User* can view their Order History all previous orders are listed. | *Registered User* |
| * 1. **Manage/View Shop Page** | The *Registered User* can edit their own Shop Page. They can add and remove products, put a title, description, logo and banner.  The *Anonymous User* can view all other *Registered User’s* Shop Pages and the products listed there. | *Registered User*  *Anonymous User* |
| * 1. **Add Product Page** | The *Registered User* can add new products with a particular category, price, media etc. which appear in their Shop Page and in the respective PLPs. | *Registered User* |
| * 1. **View My Profile** | *Registered User* can view and edit their profile data in the Profile Page. | *Registered User* |
| * 1. **Mange Homepage** | *Administrator can manage the Homepage, add products and banners and edit the overall appearance.* | *Administrator* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home Page** | Gives a brief information what are the products being sold in the platform and the last deals. | / |
| * 1. **Category Page** | Presents all products of a category. Can be accessed through the navigation. | */c/{category-id}* |
| * 1. **Search Page** | Presents all products for a search. Can be accessed via a search through the search bar. | */?query={search-term}* |
| * 1. **Product Page** | Presents a product details, images, reviews and allows a user to add to cart. | */p/{product-id}* |
| * 1. **Shop Page** | Presents *all products of a Registered User’s shop and its description.* | */user/store* |
| * 1. **Create Product Page** | Presents a view allowing the user to create a product | */user/create* |
| * 1. **User Registration Page** | Presents a view allowing the *Anonymous Users* to register in *the platform.* | */register* |
| * 1. **Login Page** | Presents a view allowing the users to login. | */login* |
| * 1. **User Profile Page** | Presents ability to view and edit personal *User Data*. | */profile* |
| * 1. **Cart Page** | Presents all products added to the cart with their quantities. | */cart* |
| * 1. **Order Page** | Presents the order form where a user can place an order after filling their details. | */place-order* |
| * 1. **Order History Page** | Presents the list of previous orders for the current customer. | */orders* |
| * 1. **Order Details Page** | Presents information about previous order details. | */orders/{id}* |
| * 1. **Wishlist Page** | Presents logged in user’s wishlist | */wishlist* |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Category** | GET request for retrieving all products for a category and page. | */api/c/{category-id}* |
| * 1. **Search** | GET request for retrieving all products for a search term and page. | */api/search?q={search-term}* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/login/* |
| * 1. **Logout** | POST a logout request for ending the active session with *OKTS,* and invalidating the issued *Security Token*. | */api/logout/* |
| * 1. **Shop** | GET, PUT, DELETE requests for retrieving shop information like product, title, description. | */api/user/products* |
| * 1. **Cart** | GET, PUT, DELETE requests for retrieving and editing cart entries information. | */api/cart/* |
| * 1. **Place Order** | PUT requests for placing an order | */api/place-order/* |
| * 1. **Order History** | GET request for retrieving all previous orders for a user. | */api/orders/* |
| * 1. **Order Details** | GET request for retrieving a particular order. | */api/orders/{order-id}/* |
| * 1. **Profile** | GET, PUT, DELETE requests for editing, deleting and retrieving of user personal information. | */api/profile/* |
| * 1. **Product** | GET, PUT, DELETE requests for creating, editing, deleting and retrieving of a product information. | */api/products/create*  */api/products/{product-id}* |
| * 1. **Wishlist** | POST,GET,DELETE requests for adding, viewing and deleting a product to a user’s wishlist | */api/user/wish/{product-id}*  */api/user/wished* |
| * 1. **Ratings** | POST,GET requests for rating a product | */api/r/{product-id}* |