## 

## **Inventory Management System**

**Course Name: DAC**

|  |  |
| --- | --- |
| **PRN** | **Name** |
| 200250120047 | Madhur Budhwani |
| 200250120019 | Bhupendra Kumar |

**About Project:**

1. The project is mainly focused on building an e-commerce website which provides items of various categories for sale.
2. There are mainly three kinds of users:
   1. New User
      1. User without login credentials
      2. Every unregistered user is initially new user only
      3. Can sign-up by providing their details like name, e-mail ID, password by doing so the new user will become the registered user.
   2. Admin User:
      1. These users are registered user with some extra functionality
      2. Can see all the registered users, can update their details, can delete any of the registered users.
      3. Can see all the available items, can update their details, can delete the item and can add new item
      4. Can see all the placed orders by the registered users and approve or reject.
   3. Registered Customer:
      1. These are the user with login credential
      2. Can update their profile
      3. Can see their order details which are placed by them
      4. Can browse through the site
      5. Can search for items
      6. Can see the item details
      7. Can place order request and post admin approval gets order confirmation or decline message.

**Flow of the Project:**

1. The project is based on MEAN stack.
2. EJS is used as view engine.
3. NodeJS is used as serve server-side technology.
4. ExpressJS framework is used for service layer and MYSQL for database.
5. Here we consider a scenario where registered user wishes to buy any item.
6. When the user visits Inventory Management System web application, he will land on login page.
7. On the login page user will be able login or go to register page. When they login their sessions are created in two ways i.e., admin or customer accordingly to how they were registered. Then home page loads with different controls as per admin or customer.
8. In the admin.js admin can operate with all editorial privileges. Admin can perform actions like ITEM\_LIST\_REQUEST where a GET(/admin/items) request is made to server. The server routes the req to items routes from where req is transferred to getItems() of admin controller and itemModel DB model. If the request to database successful then all items are fetched from the database as Item object array. The Item array is Send as JSON object to the client and items will be displayed to admin by using Item component. Admin can manipulate customer or item details using createitem , customerSearch , getitem , updateitem , deleteitem functions from itemModel DB model and createUser, getUser, updateUser, updatePassword, getAll, searchBy, updateCustomer, deleteUser functions from userModel DB model.
9. In the customer.js customer can operate with limited privileges such as view, search and order request of items. Customer can perform actions ITEM\_LIST\_REQUEST where a GET(/customer/items) request is made to server. The server routes the req to items routes from where req is transferred to getItems() of customer controller and itemModel DB model. If the request to database successful then all items are fetched from the database as Item object array. The Item array is Send as JSON object to the client and items will be displayed to customer by using Item component. Customer can view and request for item and wait for approval.
10. Admin then approves the request and stock is managed accordingly.

**A few problems faced and how we overcame them:**

1) After completing the whole checkout and payment process of an order, when we want to place next order, we perform the checkout process and want to proceed to next then last order details page is rendered automatically. This was basically due to after completing the payment we were not resetting the orders state and shipping address state. To solve this issue, we dispatched two actions of type ORDER\_DETAILS\_RESET and ITEM\_DETAILS\_RESET to reset the Item details and order details state to its initial value.

2) As each user can see their order list. Any user visits orders list screen after list screen and log out .In the same browser when new user logs in an want see their order list, the order list page was rendering on the previous state of the previous user to solve this we dispatch actions of type MY\_ORDER\_LIST\_RESET to initialize order list stat to its value.

3) While admin tried to create a new item, Create Item button wasn't working. It was because during the ItemCreate action we made a POST request to get the sample entry for the item, but we did not send any object to the server. It is necessary to send an object while mode of request is POST, and this solved our issue.

**Learnings from the project:**

1. We followed the Agile methodology in project development
2. ASANA project management tool was used as scrum board and a schedule oriented approach was followed.
3. Layout of presentation design using Pencil tool.
4. Maintaining the version of application using Git
5. Express and NodeJS on server side.
6. EJS is used as view engine and HTML5 and CSS3 for frontend.
7. MYSQL is used for database management.
8. Use of Chrome DevTools in browser.
9. Working in collaboration as a team by using tools like GitHub, Google Drive, Zoom, AnyDesk.
10. Learnt how to work as a team and how project development works in real-time remote environment