 **React- 01**

**Trainer: Mr. Govind Namdev Date: 22/08/2023**

**NOTE: Variable names should be meaningful and camel Case**

**1 ) Add to cart functionality ---**

1.1 Using state you have to update cart value and price according to quantity.

1.2 In this you have to use both functional and class components.

1.3 Its your product JSON file ---

const products = [

{

emoji: '🍦',

name: 'ice cream',

price: 5

},

{

emoji: '🍩',

name: 'donuts',

price: 2.5,

},

{

emoji: '🍉',

name: 'watermelon',

price: 4

}

];

1.4 Follow this structure —

1.4.1 Create JSON folder -> inside it create products.on file then paste the products as above.

1.4.2 Create product folder -> src/components/Product inside this folder create index.js file and style.css file.  
1.4.3 Call product/index.js file in the App.js component.

1.4.4 (Image)

**2) Create Register and login form using Hooks with validation ---**

2.1 Register fields (name, username, contact,email, password,confirmPassword).

2.2 Login fields (email, password).

2.3 Use useRef().

2.4 Saved registered user in a JSON file for login use the same JSON file.

2.5 Folder structure —

2.5.1 Create two folders under src src/components/register , src/components/login

2.5.2 Under both the folders create a index.js file for code and your logics and

style.css file for styling of html.

2.5.3 Create a src/components/router router folder for routing inside this folder create

index.js file for routing functionality at connect it with app.js file.

2.5.4 For form validation you can use anything as per your understanding.

2.5 Your form design should look like this



**React- 02**

**Trainer: Mr. Govind Namdev Date: 23/08/2023**

1. **Building a Simple To-Do List**

1.1 Showing TaskList using Functional Component.

1.1.1 All the added todos shows from here with delete icon.

1.2 Showing TaskForm using Class Component.

1.3 Folder structure —

1.3.1 Create two folders under src src/components/taskList , src/components/taskForm

1.3.2 Under both the folders create a index.js file for code and your logics and

style.css file for styling of html.

1.3.3 Create a src/components/router router folder for routing inside this folder create

index.js file for routing functionality at connect it with app.js file.

1.3.4 For form validation you can use anything as per your understanding.

1.4 Your form design should look like this.



**React- 02**

**Trainer: Mr. Govind Namdev Date: 23/08/2023**

1. **Building a Simple To-Do List**

1.1 Showing TaskList using Functional Component.

1.1.1 All the added todos shows from here with delete icon.

1.2 Showing TaskForm using Class Component.

1.3 Folder structure —

1.3.1 Create two folders under src src/components/taskList , src/components/taskForm

1.3.2 Under both the folders create a index.js file for code and your logics and

style.css file for styling of html.

1.3.3 Create a src/components/router router folder for routing inside this folder create

index.js file for routing functionality at connect it with app.js file.

1.3.4 For form validation you can use anything as per your understanding.

1.4 Your form design should look like this.



**React- 03**

**Trainer: Mr. Govind Namdev Date: 24/08/2023**

1. **Calling Backend Services**

**API’s -**

1. <https://jsonplaceholder.typicode.com/users> - get Users
2. <https://jsonplaceholder.typicode.com/posts> - get Posts
3. <https://jsonplaceholder.typicode.com/photos> - get Products

**Folder Structure -**

1) Create all folders under src src/components/users , src/components/posts and

src/components/products

2) Under 3 the folders create a index.js file for code and your logics and

style.css file for styling of html in particular folder.

3) Create a src/components/router Router folder for routing inside this folder create

index.js file for routing functionality at connect it with app.js file.

1. **API calling method “XMLHttpRequest”**

1.1 Call the get user API and get the data

1.2 Show all the list of user with (image (‘<https://cdn.pixabay.com/photo/2020/07/01/12/58/icon-5359553_1280.png>’) name, username, email, contact, address)

1.3 Add sorting functionality on name

1.4 Your list should look like this

1.5 Deploy on vercel

1. **API calling method “Fetch API”**

2.1 Call the get post API and store the data in a JSON file.

2.2 Show all the list of post with (no., title, body,Action)

2.3 Add sorting functionality on no. and edit, delete functionality.

2.4 Your list should look like this

2.5 Deploy on vercel.

1. **API calling method “Axios”**

3.1 Call the get product API and store the data in a JSON file.

2.2 Show all the list of products with (image is url, title,price (price is based on Id ex. Id is 1 then price is 1000, if it is 5 then its 5000)).

2.3 Replace wishlist icon to view icon , when clicking on that icon then the image should open within a modal.

2.4 When hovering on the product , the product will be highlighted.

2.5 Add a cart button when clicking on ‘add to cart’ button cart count will increase and cart button text will change to ‘Go to cart’.

2.6 Your list should look like this.

2.7 Deploy on vercel



**React- 04**

**Trainer: Mr. Govind Namdev Date: 25/08/2023**

**Authentication using context api —**

1. Existing register and login functionality used here.
2. Update a field which is Role.
3. When a user registers, its role is ‘user’ and creates an admin manually.
4. Create a view profile page with role dropdown. Role dropdown is disabled if ‘user’ is logged in. If ‘admin’ login then the dropdown is enabled.
5. Give here an update button for updating user details, if role is user then all fields are editable except role dropdown and if role is admin then all fields are editable. After clicking on update the details are updated without refreshing the page.
6. When the user will login then his details will be set in Context API and you have to use the same details everywhere.
7. User proper validation when updating the user.
8. Admin profile page
9. User profile page

 **React - 05**

**Trainer: Mr. Govind Namdev Date: 04/09/2023**

1. Create customer registration on the ecart project with API integration.

 **React - 05**

**Trainer: Mr. Govind Namdev Date: 04/09/2023**

1. Create the login and forget password with API integration.

 **React - 06**

**Trainer: Mr. Govind Namdev Date: 05/09/2023**

1. Create a Home page with static data similar as   
http://ecart.mangoitsol.com/home

 **React - 06**

**Trainer: Mr. Govind Namdev Date: 05/09/2023**

1. Create a Customer Profile Page similar to this website.   
<http://ecart.mangoitsol.com/home>Here is the customer login details  
[govind.mangoitsolutions@gmail.com](mailto:govind.mangoitsolutions@gmail.com)G@vind@1234

 **React - 07**

**Trainer: Mr. Govind Namdev Date: 06/09/2023**

1. Convert the Home page with dynamic data from APIs.

 **React - 07**

**Trainer: Mr. Govind Namdev Date: 06/09/2023**

1. Create an Address Page for the customer and give him an option to add a Billing and Shipping address.

 **React - 08**

**Trainer: Mr. Govind Namdev Date: 07/09/2023**

1. Create a products page with static data similar as

<http://ecart.mangoitsol.com/home.>

 **React - 08**

**Trainer: Mr. Govind Namdev Date: 07/09/2023**

1. Create an Address List Page for the customer and list all address.

Also add an option to update the address.

 **React - 09**

**Trainer: Mr. Govind Namdev Date: 12/09/2023**

1. Create a Product page with API data.

 **React - 09**

**Trainer: Mr. Govind Namdev Date: 12/09/2023**

1. Create an About and contact page. Similar to the reference website.