M.C.A PROGRAM

DEPARTMENT OF STATISTICS

[Patna University]

3rd Cycle NAAC Accredited at 'A' Grade with CGPA 3.58/4

"College with Potential for Excellence" (CPE) status accorded by UGC

Ashok Rajpath, PATNA-800002

A PROJECT REPORT ON

CONSUMER MANAGEMENT SYSTEM

DEVELOPED BY: Imran Akhtar (15), Umar Qadeer (42)

Submitted to the Department of Computer Science

In partial fulfillment of the requirements for the award of degree

Masters of Computer Application

A Project Report On

CONSUMER MANAGEMENT SYSTEM

For the partial fulfillment of requirements for the award of the degree for Master's in Computer Application.

Submitted By Imran & Umar

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled "CONSUMER MANAGEMENT SYSTEM" submitted to the MCA Program, Department of statistics in partial fulfillment of the requirements for the award of the degree of **M.C.A in Computer Application**, is an original work carried out by **Imran Akhter** and **Umar Qadeer** under the guidance of **Mr. Sonu Kuswaha (D.B.A).**

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF THE DEPARTMENT

ACKNOWLEDGEMENT

We want to express our sincere thanks to Vidhyut Bhawan, Patna, for giving us an opportunity for the Web Application development on Consumer management system which gave us a pleasurable experience.

Mr, Sonu Kumar Kushwaha, D.B.A in Patna Vidhut Bhawan, to whom we owe profound sense of gratitude, because he helped us in providing all information require for the development of our project. His valuable suggestion for amendments helped our project to take its final form.

We would like to thank our Ex-Head of department **Dr. Binod Kumar Pandey** who's friendly, cooperative and inspirational words always helped us to complete the project in the best manner. We would also like to thank our co-ordinator **Mr Satyajeet Kumar** for providing us the best environment required for the proper completion of full fledged project.

Our endeavour is also a gift of faculty members who taught us the basics of the subject in friendly and supportive way.

We are also very thankful to our family, friends and seniors who helped us and supported us at every stage of completing this project.

Thank you all.

Submitted By Imran & Umar

CONTENTS

INTRODUCTION

We have made this software under the demand of the **M.C.A (Computer Applications)** course to undergo on-job-training in the final year. Training can be done by any private or public sector organization where We can employ all the studied computer languages and the tools. The objective of the training is that the students should experience or gain the practical knowledge of working environment. And, we have gained our very experience from **Vidhyut Bhawan (Patna).**

The project incharge has given us the responsibility of developing 'CONSUMER MANAGEMENT SYSTEM' for the company itself. We made a study and overviewed the section. We have gathered necessary and relevant information regarding the section with the cordial support of the project incharge.

"CONSUMER MANAGEMENT SYSTEM" is the outcome of that proposal. The frontend used in its development is **React JS**, **Python** as backend and **Mongo DB** as the database.

The project contains information about organization, which provides quantity assurance, software for the Consumer Management in, and other details as well as following the sequential actions of the different phases in the system development life cycle (SDLC).

EXISTING SYSTEM

Existing system is facing few problems. So, a need to the new system arose. The company was already running on a computer based system but some work also done manually which had certain limitations like data redundancy and lacked normalized database. At present the system is slow; so many problems occur .The main difficulty in the present system is the slow work. This is the basis of our software development. As a result the service time increases, complexity occurred in maintaining files by which unnecessary redundancy occurred. Stock maintenance is to complex. A desired report generation takes long time. By these problems various other problems occurred, for example any stock updation, query, there product details, purchase details, stock details, are complex to retrieve. Each and every time these type of information is needed to retrieve, also they are required to stored at many places for other purposes, so it increases redundancy, complexity and takes a large amount of extra time is exhausted in a small work.

PROPOSED SYSTEM

The existing system leads to many errors like data redundancy, data inconsistency, and also much more paper works that waste very valuable time and money. So I develop software and named as “CONSUMER MANAGEMENT SYSTEM”. This computerization improves efficiency of office work, and also helps to keep data too many years without damage and can be recollected as and when needed without much time. In this system more human involvement can be reduced. We can also increase the processing speed. Easy to maintain all the information. Data Security is assured. The employee’s work is simplified and the most important the stock of products is maintain properly. We can obtain high efficiency from the computerization of this system. Also help to reduce overhead cost, time, and manpower. This system is user friendly and easy to use. Database is a collection of related organized tables so as to do easily store, retrieve and maintain data in databases without introducing redundancy.

TOOLS & ENVIRONMENT USED

APPLICATION: CONSUMER MANAGEMENT SYSTEM

OPERATING SYSTEM: WINDOWS 10 (32-bit, 64-bit)

FRONT END/GUI TOOLS: REACT JS

BACK END : PYTHON

RDBMS : MONGO DB

Submitted By Imran & Umar

SYSTEM ANALYSIS & DESIGN
of
CONSUMER MANAGEMENT SYSTEM

Submitted By Imran & Umar

STEP 1: FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.
The system has been tested for feasibility in the following points:-

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility

1. Technical Feasibility: - The project entitles "Stock Management System" is technically feasible because of the below mentioned feature. The project was developed in Java which is GUI (Graphical User Interface). It provides the high level of reliability, availability and compatibility. All these make JAVA an appropriate language for this project. Thus the existing software JAVA is a powerful language.

2. Economical Feasibility: - The system being developed is economic with the oragnization's point of view. It is cost effective in the sense that has eliminated the paper work completely. The result obtained contains minimum errors and are highly accurate as the data is required.

3. **Operational Feasibility: -** Its check whether the introduced technical equipments is performing well or giving benefits or not. The user is accepting this facility or not. The technical equipments

were performing well and the facility was very helpful for the users. So, our project was operationally feasible.

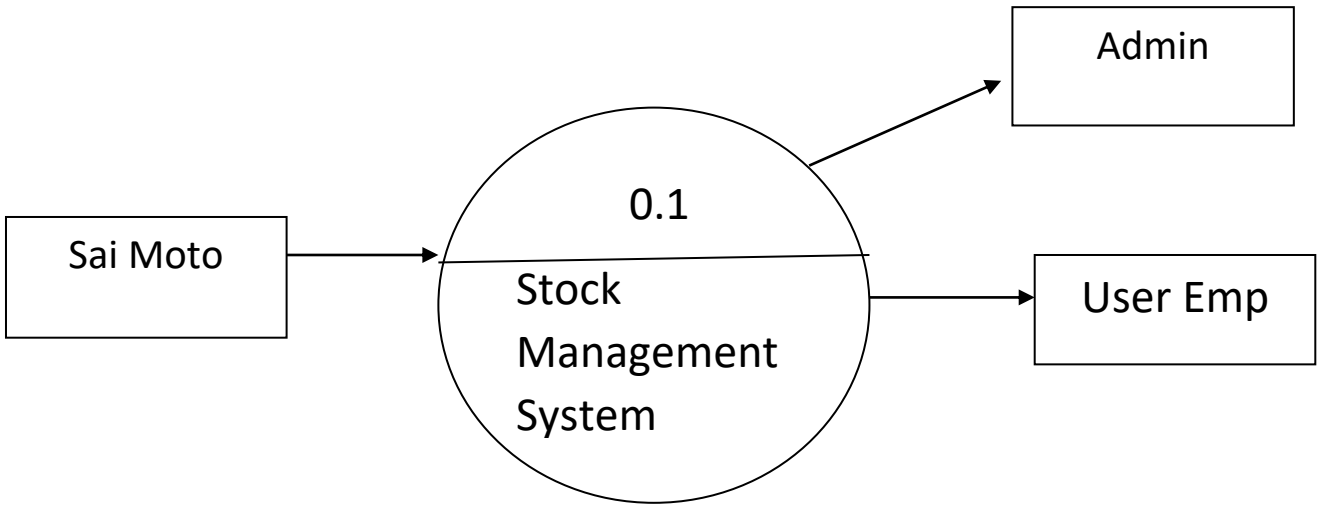
STEP 2: SOFTWARE REQUIREMENT SPECIFICATION

Minimum Requirement

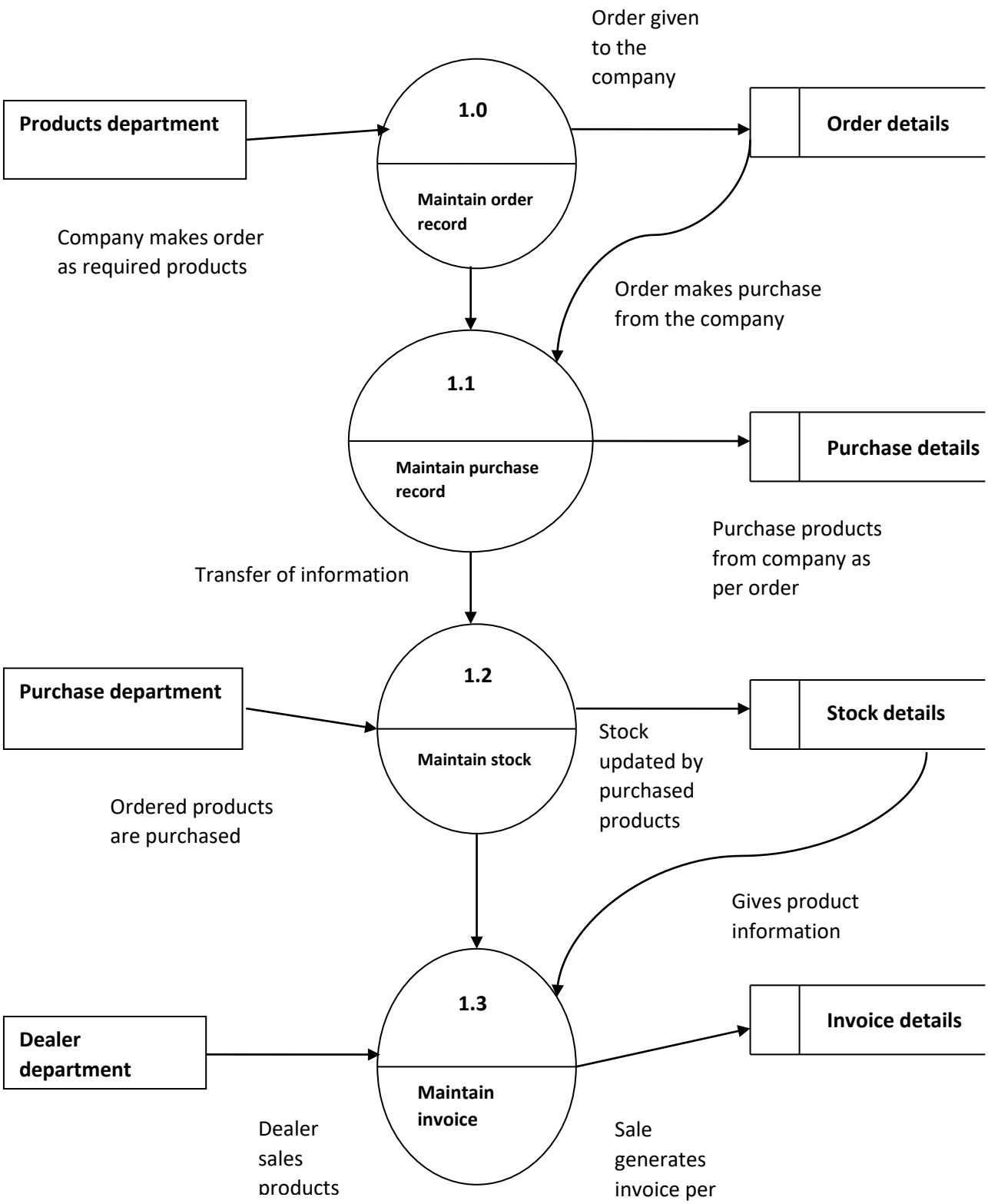
PROCESSOR	: 2.40 GHz
MEMORY	: 4GB RAM
VIDEO ADAPTER AND MONITOR	: 1366 × 768
HARD DRIVE DISK FREE SPACE	: 16GB or higher
DRIVES	: DVD-ROM or DVD
INPUT DEVICES	: Keyboard, Mouse
SOUND	: Sound cards, Speakers or Head Phones

STEP 3: DFD (DATA FLOW DIAGRAM)

0 LEVEL DFD

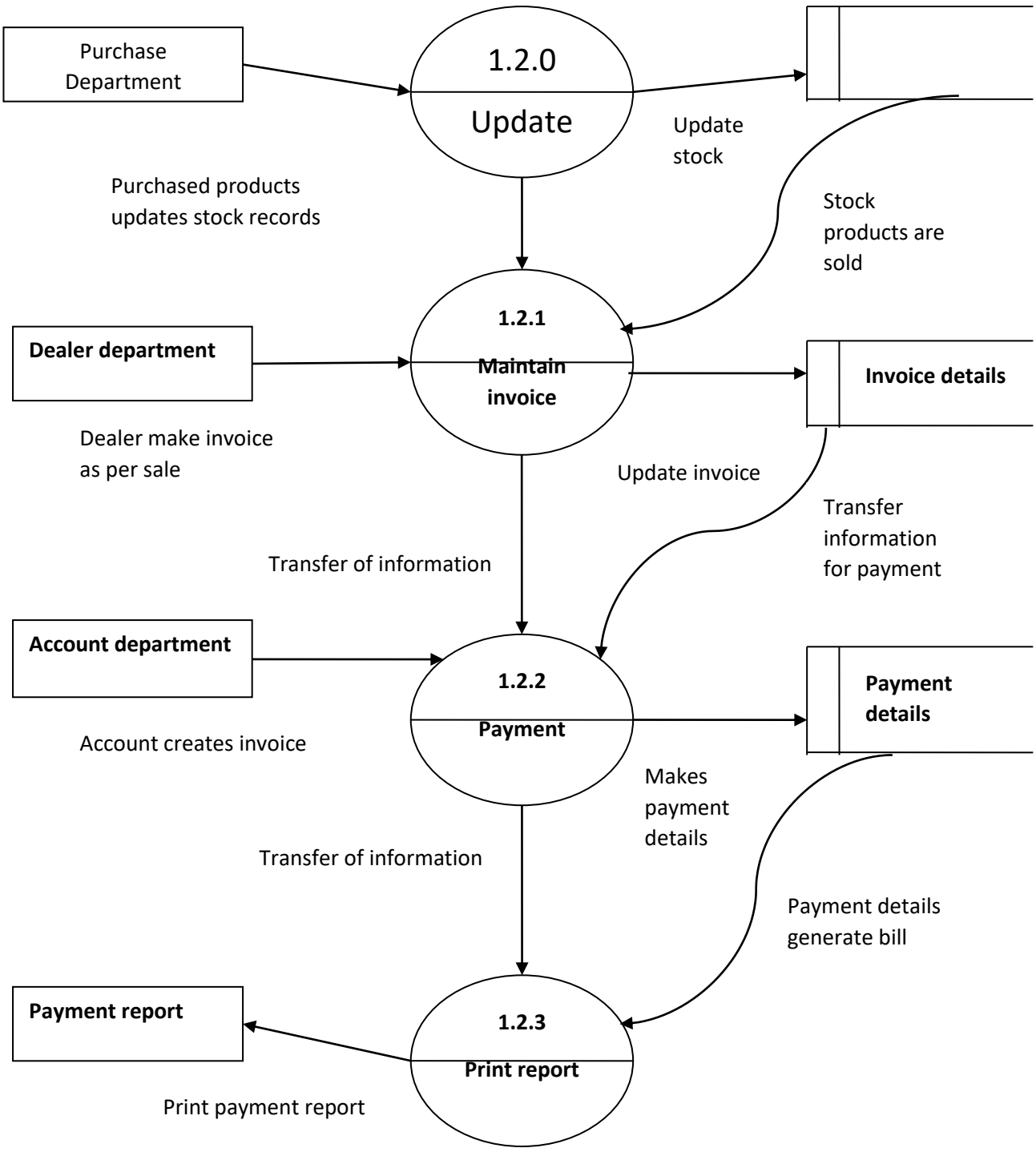


1st Level DFD



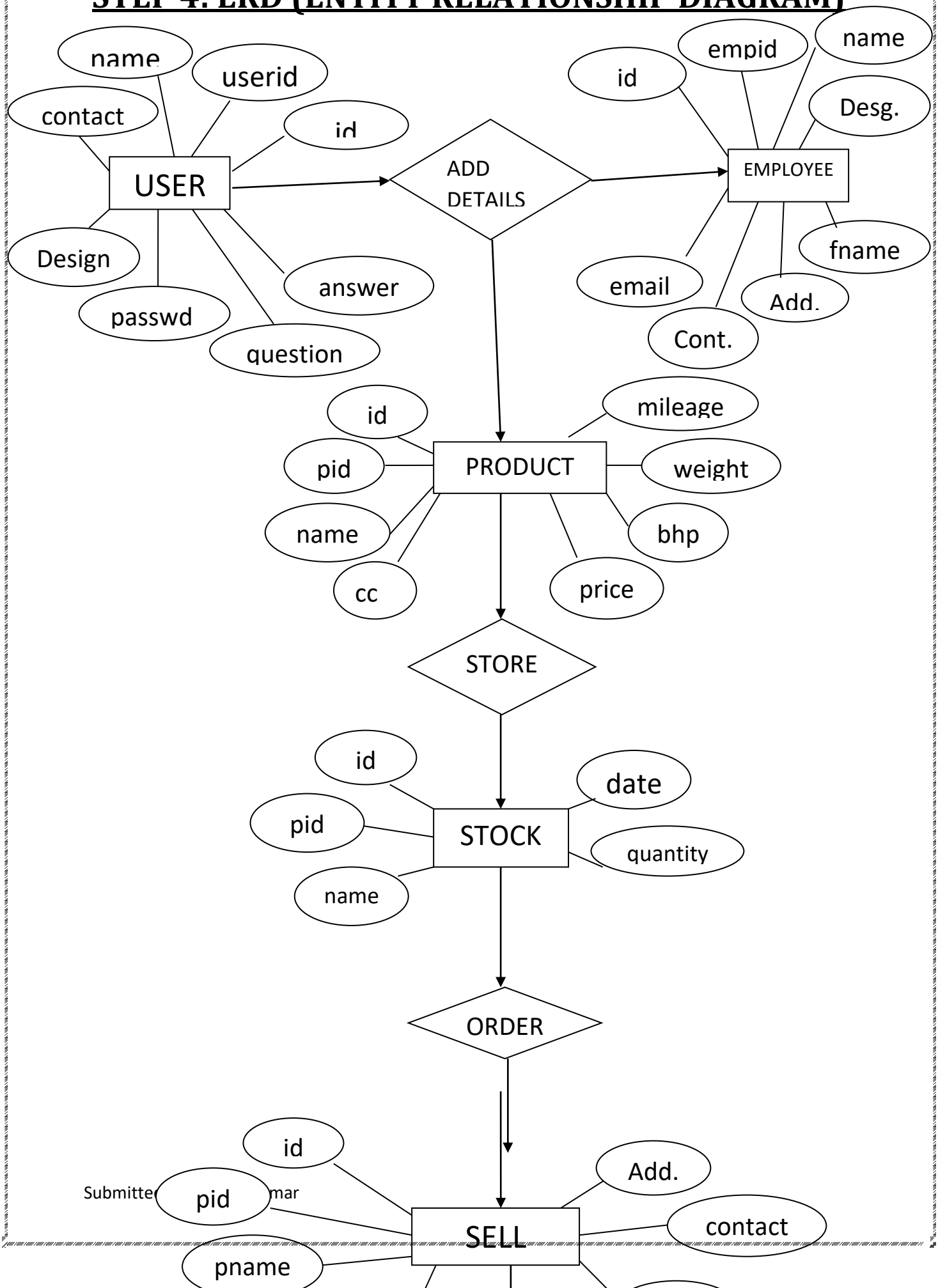
Submitted By Imran & Umar

2nd Level DFD



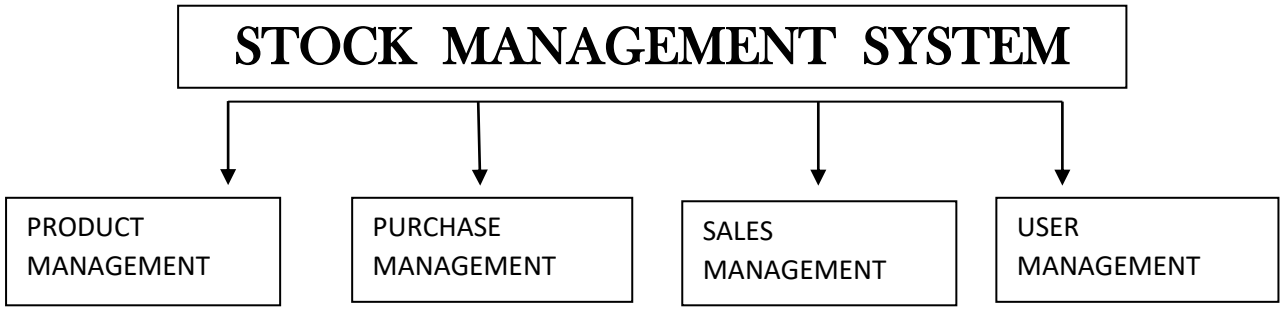
Submitted By Imran & Umar

STEP 4: ERD (ENTITY RELATIONSHIP DIAGRAM)



Submitte mar

MODULE DESCRIPTION



There are main 4 modules in the Stock Management system. They are:

- **User Management**
- **Product Management**
- **Purchase Management**
- **Sales Management**

Before discussing these modules in detail I would like to tell you that Stock Management System can have vast number of functionalities.
Each organization has different set of requirements. That's why a company using this software can have different features from another company using it.

USER MANAGEMENT

Before discussing the sub-modules of this Module let me tell you the users of Stock Management System. It is required here to understand the functionalities of this module. The users are Admin,

Manager, Sales staff, Purchasing Staff. This module is only available to the Admin or Owner of the company.

- **Add Users**

This option is used by the Admin to add new Users into the system. It will ask for all the details of the user like his/her name, email id, phone number, gender. The admin will have the option to set the type of User.
The Admin have to select that whether the new user created will be a Admin, Manager, Sales staff or Purchasing staff. The admin also have to set login credentials of the new user. For that a unique username and password will be set.
After the new user account is successfully created, that user will be notified by email.

- **List Users**

This option will show the admin all the users using the system. This will show all the basic details of the user with its date of creation and last login time. It will also have an option to view all login details of each user.
Other important feature in this module is to Activate or deactivate users account. So the admin has option to deactivate the account of any user. After which that particular user won't be able to login into his account.

- **Admin**

Admin is basically the master controller of the Stock Management System. He has the rights to manage all the modules of the system. He can add users, delete users, check the total sales in a particular month, check the pending orders, cancel a order and all the other functionalities present in the Stock management system. In short we can say that the admin has total command over the system. Generally there is only one admin, but admin has the right to give

any other user the admin rights. Apart from admin all other users will have limited access to the system.

- **Manager**

This user will have limited access. There can be many type of manager in a big organization like sales manager, product manager. So each manager will have its own set of rights. Product manager will only have access to Product management module whereas sales manager will have access rights to sales management module.

- **Sales Staff**

These are the staff which will do the billing at the point of sales. So this type of user will have access to Sell Items and do the billing work of customers.

- **Purchasing Staff**

This type of user will only have access to Purchase management module of the Stock management system. They can make orders, check previous orders and add other expenditures.
According to the needs of an organization, there can be other users as well. Stock management system is a more of a need based system.

PRODUCT MANAGEMENT

Each company can have different requirements and that's why there is a variety of this software available in the industry.

This module is used to manage the items being stocked in warehouse or in the mart. So let us dig deeper and look into the features of this module.

- **Add Product**

This sub-module is used for adding new products to the system. It will require some basic details like Product Category, Product Name, cost price, selling price, its quantity.
We can have other additional features like product image, supplier's name and its bar code.
One other interesting feature this system can have is an alert system.
You can set a particular quantity for each item. Now a notification or alert will be given to the user if a particular item's quantity gets below the set quantity. This will help the user in getting notifications of the items getting low in stock.

- **List Product**

This sub-module lists all the items present in the database of the Stock Management System. It will have options to edit the details of each item or delete a particular item from the list.

PURCHASE MANAGEMENT

This module is used to manage all purchases done by the retailer. It helps the retailer in maintaining the records of all its purchases from different suppliers.

- **Add Order**

It is used by the manager to add an order into the system. It require details like items ordered and their quantities, date of ordering, suppliers details, total cost.

It will also have a feature of current status of the order which will have options like Ordered, Pending and Received.
It can have an additional feature of importing order details from a csv or excel file.

- **List Orders**

This feature is used to display all the orders made by the merchant. It displays all the details like date of ordering, supplier details, payment status, order status.
There will be a option to view details of each order or to download those orders in pdf file format.

SALE MANAGEMENT

- **Sell Items**

This option is available to the sales persons who are at the point of sale. Point of sale is the place where billing and other transactions are done.
So here bar code of each product is read using a bar code scanner and billing of the total items is done.

After the billing the quantities of each item bought by the customer is also deducted from the available stock. The main benefit Stock Management system provides here is that the tasks performed here is all automated.
The sales persons just need to scan the bar code of the items being purchased and all other calculations will be performed by the system.

- **List Sales**

This option, as the name suggests shows all the sales made by the retailer. It shows details like date, time, total bill, payment status, view bill.

DATABASE STRUCTURE
&
CODING

Submitted By Imran & Umar

INTERFACES
&
CODING

Submitted By Imran & Umar

INDEX.JS

```
import ReactDom from "react-dom/client";
import "./index.css";
import App from './app.js';
import { BrowserRouter } from "react-router-dom";
import { Provider } from 'react-redux'
import { store, persistor } from './store'
import "../node_modules/react-bootstrap/dist/react-bootstrap";
import "../node_modules/bootstrap/dist/css/bootstrap.css";
import { PersistGate } from 'redux-persist/integration/react';
const root = ReactDom.createRoot(document.getElementById('root'))
root.render(
  <BrowserRouter>
    <Provider store={store}>
      <App />
    </Provider>
  </BrowserRouter>
)
```

APP.JS

```
import { Route, Routes } from 'react-router-dom';
import { About } from './component/About';
import { RTI } from './component/RTI';
import { Home } from './component/Home';
import { Sign_In } from './component/Sign_In';
import { New_conn } from './component/New_conn';
import { New_conn_form } from './component/New_conn_form';
import { Sign_Up } from './component/Sign_Up';
```

```
// import { For_pass } from './component/For_pass';
import { Chng_pass } from './component/Chng_pass';
import { Homee } from './component/admin/Homee';
import { Block } from './component/admin/Block';
import { Block_d } from './component/admin/Block_d';
import { Village } from './component/admin/Village';
import { Village_d } from './component/admin/Village_d';
import { District } from './component/admin/District';
import { Panchayat_d } from './component/admin/Panchayat_d';
import { Division_d } from './component/admin/Division_d';
import { Division } from './component/admin/Division';
import { Section_d } from './component/admin/Section_d';
import { Section } from './component/admin/Section';
import { Sub_Division_d } from './component/admin/Sub_Division_d';
import { Sub_Division } from './component/admin/Sub_Division';
import { Panchayat } from './component/admin/Panchayat';
import { District_d } from './component/admin/District_d';
import { Tariff } from './component/admin/Tariff';
import { Tariff_d } from './component/admin/Tariff_d';
import { Conn_type_d } from './component/admin/Conn_type_d';
import { Conn_type } from './component/admin/Conn_type';
import { User } from './component/admin/User';
import { User_d } from './component/admin/User_d';
import { Temp_app } from './component/admin/Temp_app';
import { Temp_app_d } from './component/admin/Temp_app_d';
import { DistrictEdit } from './component/admin/DistrictEdit';
import { ApplicantDetails } from './component/ApplicantDetails';
import { BlockEdit } from './component/admin/BlockEdit';
import { PanchayatEdit } from './component/admin/PanchayatEdit';
import { VillageEdit } from './component/admin/VillageEdit';
```

```
import { DivisionEdit } from './component/admin/DivisionEdit';
import { SubdivisionEdit } from './component/admin/SubdivisionEdit'
import { SectionEdit } from './component/admin/SectionEdit';
import { ConnectionTypeEdit } from './component/admin/ConnectionTypeEdit';
import { Tension_type } from './component/admin/Tension_type';
import { Tension_type_d } from './component/admin/Tension_type_d';
import { TensionTypeEdit } from './component/admin/TensionTypeEdit';
import { TariffEdit } from './component/admin/TariffEdit';
import { Phase } from './component/admin/Phase';
import { Phase_d } from './component/admin/Phase_d';
import { PhaseEdit } from './component/admin/PhaseEdit';
import { Load } from './component/admin/Load';
import { Load_d } from './component/admin/Load_d';
import { LoadEdit } from './component/admin/LoadEdit';
import { ServiceType } from './component/admin/ServiceType';
import { ServiceType_d } from './component/admin/ServiceType_d';
import { ServiceTypeEdit } from './component/admin/ServiceTypeEdit';
import { SignUp } from './component/admin/SignUp';
import { SignIn } from './component/admin/SignIn';
import { OfficerSignUp } from './component/admin/OfficerSignUp';
import { OfficerSignIn } from './component/officer/OfficerSignIn';
import { AdminChangePassword } from './component/admin/AdminChangePassword';
import { AdminForgetPassword } from './component/admin/AdminForgetPassword';
import './app.css'
import { useSelector } from 'react-redux'
import { ChangePassword } from './component/ChangePassword';
import { CheckStatus } from './component/CheckStatus';
import { Status } from './component/Status';
import { Navigate } from 'react-router-dom'
import { PageNotFound } from './component/PageNotFound';
```

```
import { ApplicantEditProfile } from './component/ApplicantEditProfile';
import { AdminEditProfile } from './component/admin/AdminEditProfile'
import { OfficerChangePassword } from './component/officer/OfficerChangePassword'
import { OfficerEditProfile } from './component/officer/OfficerEditProfile'
import { OfficerDashboard } from './component/officer/OfficerDashboard';
import { ApplicationPrint } from './component/ApplicationPrint';
import { CheckApplication } from './component/CheckApplication'
import { MyApplication } from './component/MyApplication'
import { ApplicationVerify } from './component/officer/ApplicationVerify'
import { ApplicationVerifyDetail } from './component/officer/ApplicationVerifyDetail';
import { ApplicationReVerify } from './component/officer/ApplicationReverify';
import { ApplicationReverifyDetail } from
'./component/officer/ApplicationReverifyDetail';
import { ProtectedOfficer } from './component/Protected'
import { CheckForgetPassword } from './component/CheckForgetPassword';
import { ForgetPassword } from './component/ForgetPassword';
import { AdminCheckForgetPassword } from
'./component/admin/AdminCheckForgetPassword';
import { OfficerCheckForgetPassword } from
'./component/officer/OfficerCheckForgetPassword';
import { OfficerForgetPassword } from './component/officer/OfficerForgetPassword';


function App() {
 return (
  <div className='htmll'>
   <Routes>
    <Route path='/' element={<Home />} />
    <Route path='/about' element={<About />} />
    <Route path='/rti' element={<RTI />} />
    <Route path='/new_conn' element={<New_conn />} />
    <Route path='/new_conn_form' element={< New_conn_form />} />
```

```
<Route path='/sign_in' element={<Sign_In />} />
<Route path='/sign_up' element={<Sign_Up />} />
<Route path='/change_password' element={<ChangePassword />} />
<Route path='/check_status' element={<CheckStatus />} />
<Route path='/check_forgetpassword' element={<CheckForgetPassword />} />
<Route path='/status' element={<Status />} />
<Route path='/print' element={<ApplicationPrint />} />
<Route path='/check_application' element={<CheckApplication />} />
<Route path='/my_application' element={<MyApplication />} />
<Route path='/applicantDetails' element={<ApplicantDetails />} />
<Route path='/edit_profile' element={<ApplicantEditProfile />} />
<Route path='/for_pass' element={<ForgetPassword />} />
<Route path='/chng_pass' element={<Chng_pass />} />
<Route path='/admin/signup' element={<SignUp />} />
<Route path='/admin' element={<SignIn />} />
<Route path='/admin/change_password' element={<AdminChangePassword />} />
<Route path='admin/check_forgetpassword'
element={<AdminCheckForgetPassword />} />
<Route path='admin/forget_password' element={<AdminForgetPassword />} />
<Route path='/admin/edit_profile' element={<AdminEditProfile />} />
<Route path='/admin/block' element={<Block />} />
<Route path='/admin/block/edit/:id' element={<BlockEdit />} />
<Route path='/admin/block_d' element={<Block_d />} />
<Route path='/admin/district' element={<District />} />
<Route path='/admin/district/edit/:id' element={<DistrictEdit />} />
<Route path='/admin/district_d' element={<District_d />} />
<Route path='/admin/division_d' element={<Division_d />} />
<Route path='/admin/division' element={<Division />} />
<Route path='/admin/division/edit/:id' element={<DivisionEdit />} />
<Route path='/admin/sub_Division_d' element={<Sub_Division_d />} />
```

```
<Route path='/admin/sub_division' element={<Sub_Division />} />
<Route path='/admin/subdivision/edit/:id' element={<SubdivisionEdit />} />
<Route path='/admin/Panchayat_d' element={<Panchayat_d />} />
<Route path='/admin/panchayat/edit/:id' element={<PanchayatEdit />} />
<Route path='/admin/Panchayat' element={<Panchayat />} />
<Route path='/admin/dashboard' element={<Homee />} />
<Route path='/admin/section' element={<Section />} />
<Route path='/admin/section_d' element={<Section_d />} />
<Route path='/admin/section/edit/:id' element={<SectionEdit />} />
<Route path='/admin/tariff' element={<Tariff />} />
<Route path='/admin/tariff_d' element={<Tariff_d />} />
<Route path='/admin/tariff/edit/:id' element={<TariffEdit />} />
<Route path='/admin/village' element={<Village />} />
<Route path='/admin/village_d' element={<Village_d />} />
<Route path='/admin/village/edit/:id' element={<VillageEdit />} />
<Route path='/admin/phase' element={<Phase />} />
<Route path='/admin/phase_d' element={<Phase_d />} />
<Route path='/admin/phase/edit/:id' element={<PhaseEdit />} />
<Route path='/admin/conn_type' element={<Conn_type />} />
<Route path='/admin/connectionType/edit/:id' element={<ConnectionTypeEdit
/>} />
<Route path='/admin/conn_type_d' element={<Conn_type_d />} />
<Route path='/admin/tension_type' element={<Tension_type />} />
<Route path='/admin/tension_type_d' element={<Tension_type_d />} />
<Route path='/admin/tensionType/edit/:id' element={<TensionTypeEdit />} />
<Route path='/admin/load' element={<Load />} />
<Route path='/admin/load_d' element={<Load_d />} />
<Route path='/admin/load/edit/:id' element={<LoadEdit />} />
<Route path='/admin/service_type' element={<ServiceType />} />
<Route path='/admin/service_type_d' element={<ServiceType_d />} />
```

```
      <Route path='/admin/service_type/edit/:id' element={<ServiceTypeEdit />} />
      <Route path='/admin/User' element={<User />} />
      <Route path='/admin/User_d' element={<User_d />} />
      <Route path='/admin/temp_app' element={<Temp_app />} />
      <Route path='/admin/temp_app_d' element={<Temp_app_d />} />
      <Route path='/admin/temp_app_d' element={<Temp_app_d />} />
      <Route path='/officer/signup' element={<OfficerSignUp />} />
      <Route path='/officer' element={<OfficerSignIn />} />
      <Route path='/officer/dashboard' element={< OfficerDashboard />} />
      <Route path='/officer/edit_profile' element={<OfficerEditProfile />} />
      <Route path='/officer/change_password' element={<OfficerChangePassword />}
/>
      <Route path='officer/check_forgetpassword'
element={<OfficerCheckForgetPassword />} />
      <Route path='officer/forget_password' element={<OfficerForgetPassword />} />
      <Route path='/application/verify' element={<ApplicationVerify />} />
      <Route path='/application/reverify' element={<ApplicationReVerify />} />
      <Route path='/application/verifydetail' element={<ApplicationVerifyDetail />} />
      <Route path='/application/reverifydetail' element={<ApplicationReverifyDetail />}
/>
      <Route path='/*' element={<PageNotFound />} />
    </Routes>
   </div>
 )
}
export default App;
```

HOME.JS

```
import { Slider } from './Slider';
import { Content } from './Content';
```

```
import { Footer } from './Footer';
import { Navbar_r } from './Navbar';
export function Home() {
  return (
  <>
  <Navbar_r />
   <Slider/>
   {/* <p>HOME</p> */}
   <Content/>
   <Footer/>
    </>
  );
}
```

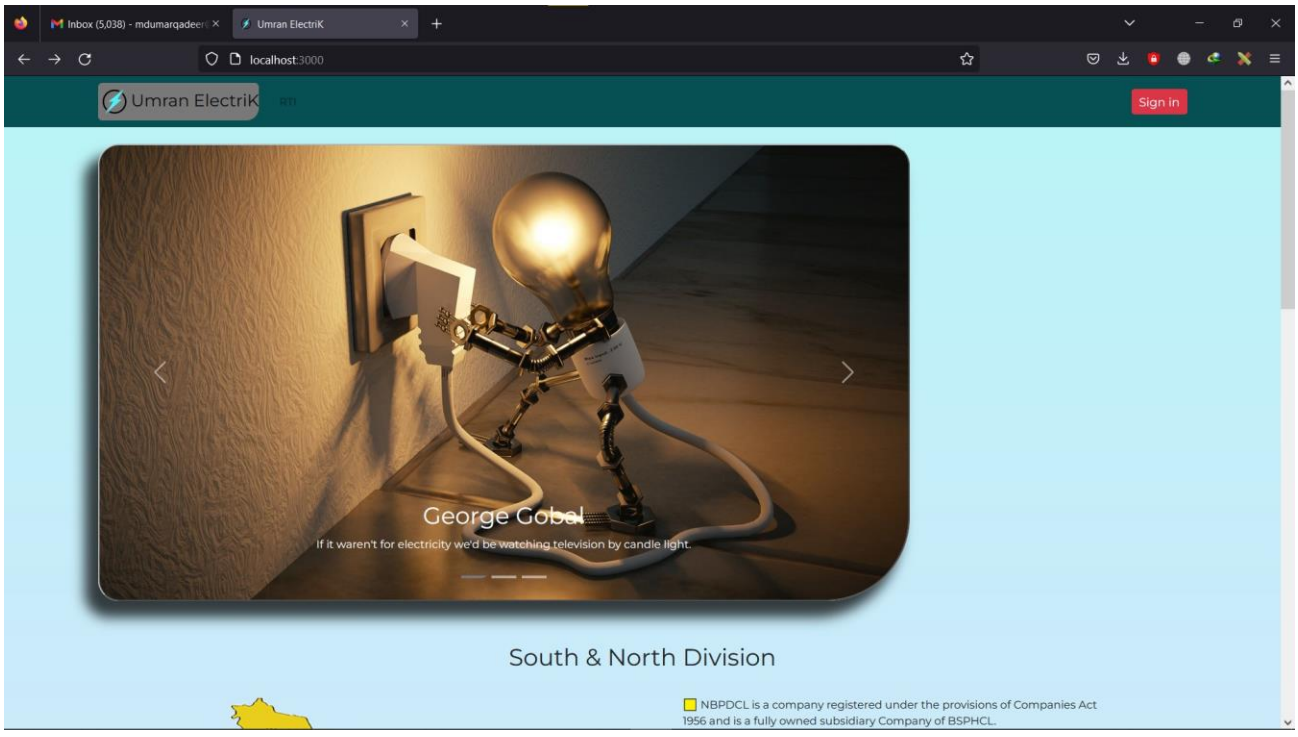
NAVBAR.JS

```
import Container from 'react-bootstrap/Container';
import Nav from 'react-bootstrap/Nav';
import Navbar from 'react-bootstrap/Navbar';
import NavDropdown from 'react-bootstrap/NavDropdown';
import { NavLink, Link } from 'react-router-dom';
import aa from '../slider/aa.png';
import a from '../slider/a.png';
import { useState } from 'react';
import "./css/Navbar.css";
import { useListadminGetUserQuery } from
'../services/applicant/applicantSignUpService';
import Button from 'react-bootstrap/Button';
import AsyncLocalStorage from '@createnextapp/async-local-storage'
import Spinner from 'react-bootstrap/Spinner';
import Image from 'react-bootstrap/Image'
```

```
export function Navbar_r() {


  const { refresh, access = null } = localStorage.getItem("applicant") ?
JSON.parse(localStorage.getItem("applicant")) : {}

  const { data: getUser = null, isSuccess, isLoading } =
useListadminGetUserQuery(access)

  return (

   <>

    {isLoading ? (

     <>

      <Spinner animation="grow" variant="info" />

     </>

    ) : (

     <Navbar className="navbar_background" expand="lg">

      <Container>

       <Navbar.Brand as={NavLink} to="/" end>

        <img

         className="logo"

         src={a}

         alt="logo"

        />

        Umran ElectriK

       </Navbar.Brand>

       <Navbar.Toggle aria-controls="basic-navbar-nav" />

       <Navbar.Collapse id="basic-navbar-nav">

        <Nav className="me-auto">

         <Nav.Link as={NavLink} to="/rti" end>RTI</Nav.Link>

        </Nav>

       </Navbar.Collapse>

       {getUser ? (
```

```
<Nav>
 <Image
  className="user_pic"
  src={getUser.photo}
  alt="logo"
  roundedCircle
 >
 </Image>
 <NavDropdown title={getUser.name} id="basic-nav-dropdown">
  <NavDropdown.Item as={Link} to="/edit_profile">
   Edit Profile
  </NavDropdown.Item>
  <NavDropdown.Item as={Link} to="/check_application">
   My Application
  </NavDropdown.Item>
  <NavDropdown.Item as={Link} to="/check_status">
   Check Status
  </NavDropdown.Item>
  <NavDropdown.Item as={Link} to="/print">
   Print Application
  </NavDropdown.Item>
  <NavDropdown.Item as={Link} to="/new_conn">
   New Connection
  </NavDropdown.Item>
  <NavDropdown.Item as={Link} to="/change_password">
   Change Password
  </NavDropdown.Item>
  <NavDropdown.Divider />
  <NavDropdown.Item as={Link} to="/" onClick={() =>
AsyncLocalStorage.removeItem('applicant')}>
```

```
            Logout
           </NavDropdown.Item>
          </NavDropdown>
         </Nav>
        ) : (
         <Button as={Link} to="/sign_in" variant="danger" size="sm">Sign in</Button>
        )}
       </Container>
      </Navbar>
    )}
   </>
  )
}
```



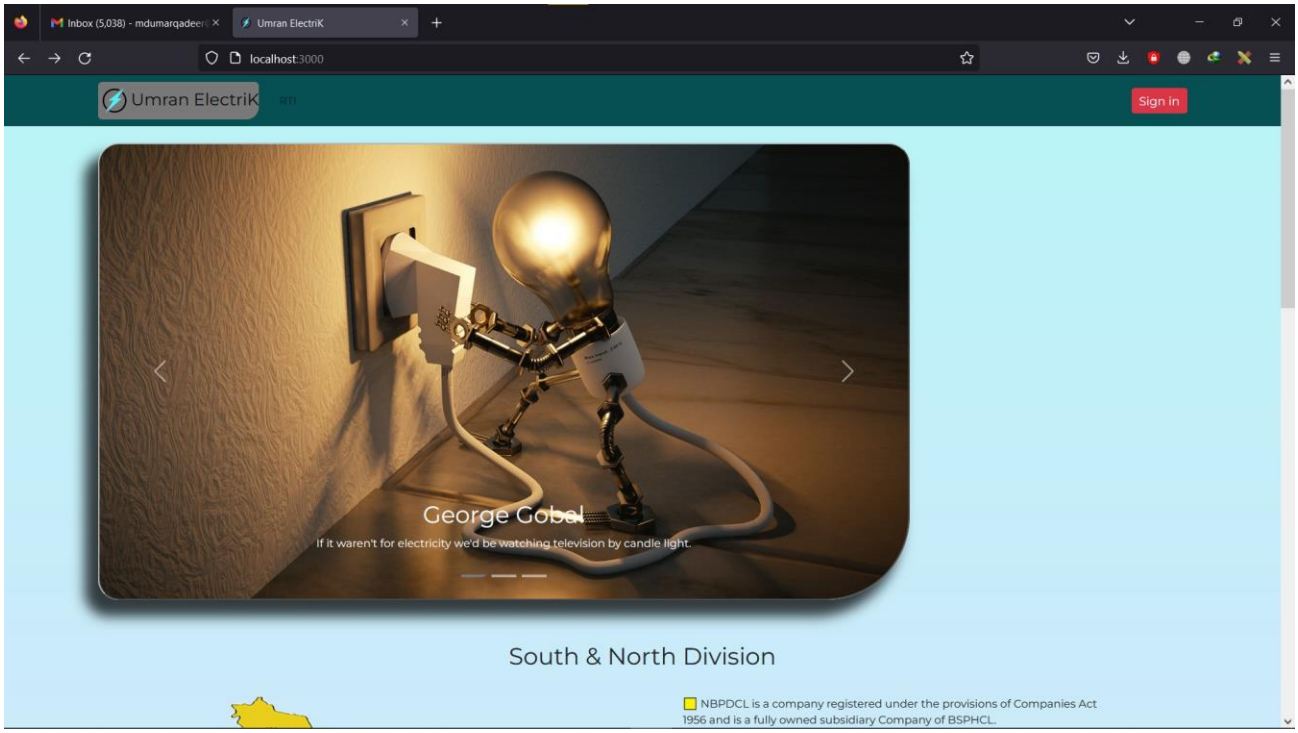
****

Submitted By Imran & Umar

RTI

```
import Container from 'react-bootstrap/Container';
import { Navbar_r } from './Navbar';
import Table from 'react-bootstrap/Table';
import Form from 'react-bootstrap/Form';
export function RTI() {
 return (
  <>  <Navbar_r />
   <Container >
    <Form className='common_csss' >
     <h5>Name and address of the officers related to RTI :</h5>
     <br />
      <Table bordered hover size="lg">
       <thead>
        <tr>
         <th>Name.</th>
         <th>Designation</th>
         <th>Contact No.</th>
        </tr>
       </thead>
       <tbody>
        <tr>
         <td>Shri Raamish Tauseef</td>
         <td>Appellate Authority</td>
         <td>0612-2504489 (O) </td>
        </tr>
        <tr>
```

```
            <td>Shri Dipak Kumar</td>
            <td>Public Information Officer</td>
            <td>0612-2950374 (O)</td>
           </tr>
           <tr>
            <td>Shri Amit Kumar Laha</td>
            <td>Asst. Public Information Officer</td>
            <td>0612-2950374 (O)</td>
           </tr>
          </tbody>
         </Table>
         <br />
         ☛ Address:
         <br />
         Bihar Electricity Regulatory Commission
         <br />
         Ground Floor, Vidyut Bhawan-2
         <br />
         J.L. Nehru Marg, Patna - 800021
       </Form>
     </Container>
    </>
  )
}
```



****

ADMIN LOGIN

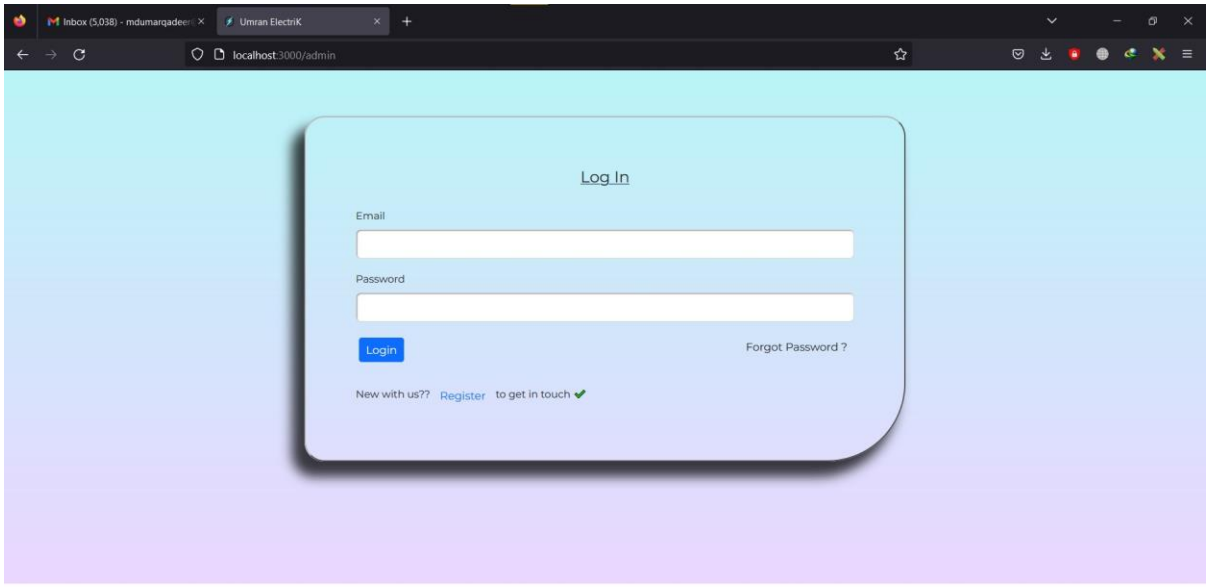
```
 import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Nav from 'react-bootstrap/Nav';
import { NavLink, useNavigate } from 'react-router-dom';
import { useCookies } from 'react-cookie'
import { useFormik } from 'formik';
import Alert from 'react-bootstrap/Alert'
import { useState, useMemo, useEffect, useCallback } from 'react'
import { useCreateadminSignInMutation } from
'../../services/admin/adminSignInService';
import { adminSignInSchema } from '../validation/admin/validation';
import AsyncLocalStorage from '@createnextapp/async-local-storage'
```

Submitted By Imran & Umar

```
import { setToken, unSetToken } from '../../features/token';
import { useSelector, useDispatch } from 'react-redux'
const initialValues = {
  email: "",
  password: ""
}
export function SignIn() {
  const [signin, data] = useCreateadminSignInMutation()
  const [show, setShow] = useState(false);
  const [alert, setAlert] = useState("")
  const navigate = useNavigate()
  const dispatch = useDispatch()
  const { values, errors, handleChange, handleSubmit, touched, handleBlur } =
useFormik({
    initialValues: initialValues,
    validationSchema: adminSignInSchema,
    onSubmit: async (values, action) => {
      Object.keys(values).forEach((key) => {
        values[key] = values[key].trim()
      })
      try {
        const response = await signin(values)
        if (response.data) {
          setShow(false)
          const token = response.data
          await AsyncLocalStorage.setItem('user', JSON.stringify(token))
          dispatch(setToken("imran"))
          navigate('/admin/dashboard')
```

```
            }
            if (response.error) {
              if (response.error.status === 401) {
                setShow(true)
                setAlert("email or password incorrect")
              }
            }
          }
          catch {
            console.log("error")
          }
        }
      })
      const closeTab = () => {
        window.opener = null;
        window.open("", "_self");
        window.close();
      }
      return (
        <>
          <Form className='common_css sign_in_form' onSubmit={handleSubmit}>
            <Alert show={show} variant='danger'>{alert}</Alert>
            <h5>Log In</h5>
            <br />
            <Form.Group className="mb-3" controlId="formBasicEmail" >
              <Form.Label>Email</Form.Label>
              <Form.Control type="text" name="email" value={values.email}
onChange={handleChange} onBlur={handleBlur} placeholder="" />
```

```
            {errors.email && touched.email ? (<p className='text-
danger'>{errors.email}</p>) : null}
          </Form.Group>
          <Form.Group className="mb-3" controlId="formBasicPassword">
            <Form.Label>Password</Form.Label>
            <Form.Control type="password" name="password"
value={values.password} onChange={handleChange} onBlur={handleBlur}
placeholder="" />
            {errors.password && touched.password ? (<p className='text-
danger'>{errors.password}</p>) : null}
          </Form.Group>
          <Button variant="primary" type="submit" className="m-1" size="sm">
            Login
          </Button>
          <Button variant="Light" style={{ float: 'right' }} size="sm">
            <Nav.Link as={NavLink} onClick={closeTab}
to="/admin/check_forgetpassword">Forgot Password ?</Nav.Link>
          </Button>
          <br /><br />
          New with us??
          <Button variant="link" style={{ "text-decoration": 'none' }}
className="m-1" size="sm">
            <Nav.Link as={NavLink} onClick={closeTab}
to="/admin/signup">Register</Nav.Link>
          </Button>to get in touch ✔
        </Form>
      </>
  )
}
```



ADMIN SIGN UP

```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import { Navbar_r } from '../Navbar';
import Alert from 'react-bootstrap/Alert';
import Nav from 'react-bootstrap/Nav';
import { NavLink } from 'react-router-dom';
import { useCreateadminSignUpMutation } from
'../../services/admin/adminSignUpService';
import { useRef, useState } from 'react'
import swal from 'sweetalert'
import { useFormik } from 'formik'
```

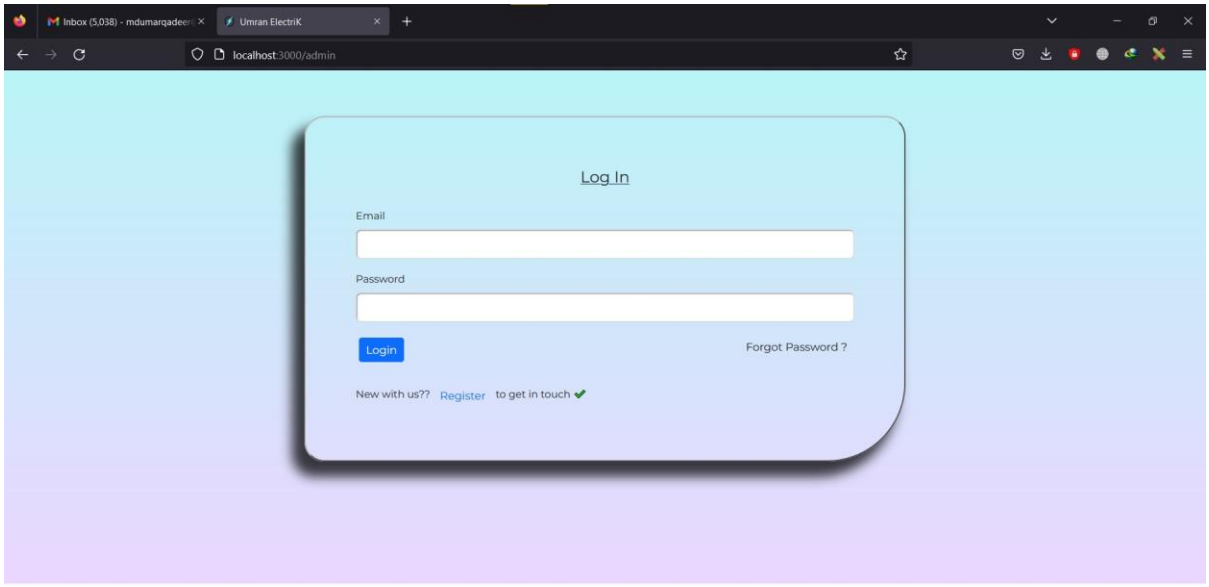
```
import { adminSignUpSchema } from
'../validation/admin/validation';
import { Navbar_admin } from './Navbarr';
const initialValues = {
   name: '',
   email: '',
   phone: '',
   password: '',
   confirm_password: '',
   is_superuser: true,
   is_staff: true
}
export function SignUp() {
   const [successShow, setSuccessShow] = useState(false)
   const [message, setMessage] = useState("")
   const [signup, data] = useCreateadminSignUpMutation()
   const inputRef = useRef(0)
   const [show, setShow] = useState(false);
   const [error, setError] = useState([])
   const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
      initialValues: initialValues,
      validationSchema: adminSignUpSchema,
      onSubmit: async (values, action) => {
         Object.keys(values).forEach((key) => {
```

```
            if (values[key] != true && values[key] != false) {
                values[key] = values[key].trim()
            }
        })
        const response = await signup(values)
        if (response.data) {
            setShow(false)
            setSuccessShow(true)
            setMessage("you have SignUp successfully")
            action.resetForm()
            setTimeout(() => {
                setSuccessShow(false)
            }, 1000)
        }
        else if (response.error.status === 406) {
            const e = Object.values(response.error.data)
            setShow(true)
            const currentError = []
            e.map(e => {
                currentError.push(...e)
            })
            setError(currentError)
        }
    },
})
```

```
  return (
    <>
      <Navbar_admin />
      <Container>
        <Form className='common_css' onSubmit={handleSubmit}>
          <Alert show={show} variant='danger'><ul>{error.map(error => <li>{error}</li>)}</ul></Alert>
          <h5>Register With Us</h5>
          <br />
          <Form.Group className="mb-3" controlId="formBasicEmail" ref={inputRef}>
            <Form.Label>Name <span className='text-danger'>*</span></Form.Label>
            <Form.Control type="text" name="name" value={values.name} placeholder="" onChange={handleChange} onBlur={handleBlur} />
            {errors.name && touched.name ? (<p className='text-danger'>{errors.name}</p>) : null}
          </Form.Group>
          <Form.Group className="mb-3" controlId="formBasicEmail">
            <Form.Label>Email <span className='text-danger'>*</span></Form.Label>
            <Form.Control type="email" name="email" value={values.email} placeholder="" onChange={handleChange} onBlur={handleBlur} />
```

```
                {errors.email && touched.email ? (<p
className='text-danger'>{errors.email}</p>) : null}

            </Form.Group>

            <Form.Group className="mb-3"
controlId="formBasicEmail">

                <Form.Label>Mobile No <span className='text-
danger'>*</span></Form.Label>

                <Form.Control type="text" name="phone"
value={values.phone} placeholder="" onChange={handleChange}
onBlur={handleBlur} />

                {errors.phone && touched.phone ? (<p
className='text-danger'>{errors.phone}</p>) : null}

            </Form.Group>

            <Form.Group className="mb-3"
controlId="formBasicPassword">

                <Form.Label>Password <span className='text-
danger'>*</span></Form.Label>

                <Form.Control type="password" name="password"
value={values.password} placeholder=""
onChange={handleChange} onBlur={handleBlur} />

                {errors.password && touched.password ? (<p
className='text-danger'>{errors.password}</p>) : null}

            </Form.Group>


            <Form.Group className="mb-3"
controlId="formBasicPassword">

                <Form.Label>Re-enter Password <span
className='text-danger'>*</span></Form.Label>
```

```
                <Form.Control type="password"
name="confirm_password" value={values.confirm_password}
onChange={handleChange} onBlur={handleBlur} placeholder=""
/>

                {errors.confirm_password &&
touched.confirm_password ? (<p className='text-
danger'>{errors.confirm_password}</p>) : null}

              </Form.Group>

              <Button type="submit" variant="primary"
className="m-1" size="sm">

                Register

              </Button>

            </Form>

            <Alert show={successShow}
variant="success">{message}</Alert>

          </Container>

      </>

    )
}
```

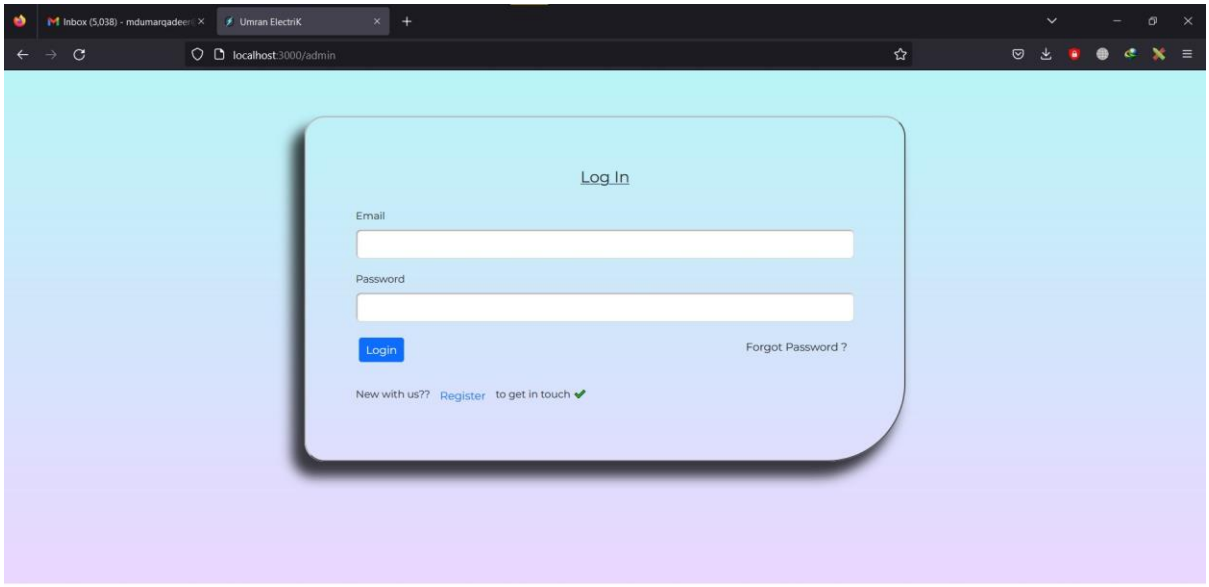


****

ADMIN DASHBOARD

```
// import { Link, Route, Routes } from 'react-router-dom';
import Container from 'react-bootstrap/Container';
import Row from 'react-bootstrap/Row';
import Col from 'react-bootstrap/Col';
import { Navbar_admin } from '../admin/Navbarr';
import { Sidebar } from '../admin/Sidebar';

export function Homee() {
  return (
    <>
      <Navbar_admin />
      <Container>
```

```
            <hr />
            <h5>Dashboard</h5>
            <hr />
            <br />
            <Row >
              <Col sm={2}>
                <Sidebar />
              </Col>
              <Col sm={10}>
              </Col>
            </Row>
          </Container>
        </>
    )
}
```



Submitted By Imran & Umar

****

ADMIN SIDEBAR

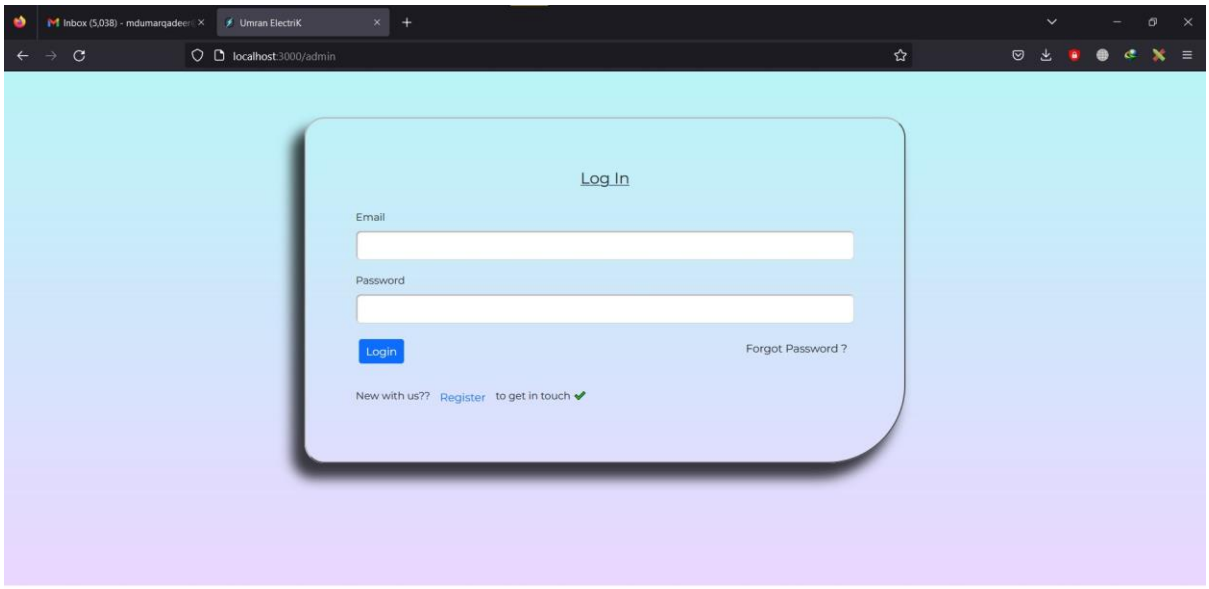
```
import Table from 'react-bootstrap/Table';
import Nav from 'react-bootstrap/Nav';
import { NavLink } from 'react-router-dom';
export function Sidebar() {
  return (
    <>
      <Table bordered hover>
        <thead>
          <tr>
            <th>Authentication & Authorization</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Groups    <span style={{ float: 'right' }}>+
</span> </td>
          </tr>
        </tbody>
      </Table>
      {/* table 2********************************************* */}
      <Table bordered hover>
        <thead>
          <tr>
```

```
            <th>Consumer</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <Nav.Link as={NavLink} to="/officer/signup">Officer
<span style={{ float: 'right' }}>+  </span></Nav.Link>
          </tr>
          <tr>
            <Nav.Link as={NavLink}
to="/admin/conn_type_d">Connection Type <span style={{ float:
'right' }}>+  </span></Nav.Link>
          </tr>
          <tr>
            <Nav.Link as={NavLink}
to="/admin/tension_type_d">Tension Type <span style={{ float:
'right' }}>+  </span></Nav.Link>
          </tr>
          <tr>
            <Nav.Link as={NavLink}
to="/admin/phase_d">Phase <span style={{ float: 'right' }}>+
</span></Nav.Link>
          </tr>
          <tr>
            <Nav.Link as={NavLink} to="/admin/load_d">Load
<span style={{ float: 'right' }}>+  </span></Nav.Link>
```

```
            </tr>
            <tr>
              <Nav.Link as={NavLink} to="/admin/block_d">Block
<span style={{ float: 'right' }}>+  </span></Nav.Link>
            </tr>
            <tr>
              <Nav.Link as={NavLink}
to="/admin/district_d">District <span style={{ float: 'right' }}>+
</span></Nav.Link>
            </tr>
            <tr>
              <Nav.Link as={NavLink}
to="/admin/panchayat_d">Panchayat <span style={{ float: 'right'
}}>+  </span></Nav.Link>
            </tr>
            <tr>
              <Nav.Link as={NavLink}
to="/admin/service_type_d">Service Type <span style={{ float:
'right' }}>+  </span></Nav.Link>
            </tr>
            <tr>
              <Nav.Link as={NavLink} to="/admin/tariff_d">Tarrif
<span style={{ float: 'right' }}>+  </span></Nav.Link>
            </tr>
            <tr>
```

```
                <Nav.Link as={NavLink}
to="/admin/village_d">Village <span style={{ float: 'right' }}>+
</span></Nav.Link>
              </tr>
              <tr>
                <Nav.Link as={NavLink} to="/admin/user_d">User
<span style={{ float: 'right' }}>+  </span></Nav.Link>
              </tr>
            </tbody>

          </Table>
          <Table bordered hover>
            <thead>
              <tr>
                <th>Select Discom Office</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <Nav.Link as={NavLink}
to="/admin/division_d">Division <span style={{ float: 'right'
}}>+  </span></Nav.Link>
              </tr>
              <tr>
```

```
                <Nav.Link as={NavLink}
to="/admin/sub_division_d">Sub Division <span style={{ float:
'right' }}>+  </span></Nav.Link>
            </tr>
            <tr>
              <Nav.Link as={NavLink}
to="/admin/section_d">Section <span style={{ float: 'right' }}>+
</span></Nav.Link>
            </tr>
          </tbody>
        </Table>
      </>
    )
}
```



****

Submitted By Imran & Umar

ADMIN ENTRIES (BLOCK (CREATING))

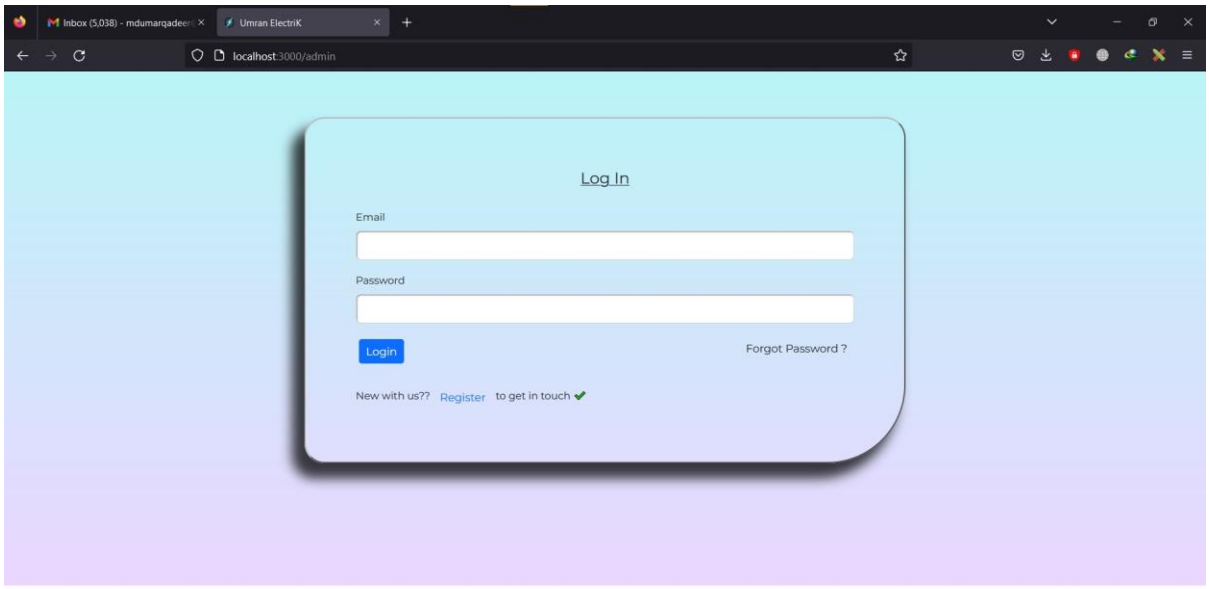
```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Col from 'react-bootstrap/Col';
import Row from 'react-bootstrap/Row';
import { Navbar_admin } from './Navbarr';
import { Sidebar } from './Sidebar';
import { useCreateblockMutation } from
'../../services/admin/blockService'
import { blockSchema } from '../validation/admin/validation';
import { useFormik } from 'formik';
import { useEffect, useState, useRef, useMemo } from 'react';
import Alert from 'react-bootstrap/Alert';
import { useNavigate, useParams } from 'react-router-dom';
import { useListdistrictQuery } from
'../../services/admin/districtService';
import Spinner from 'react-bootstrap/Spinner';
const initialValues = {
   block: '',
   district: '',
}

export function Block() {
```

```
  const [createBlock, createBlockFlag] =
useCreateblockMutation()
  const { data: districtList, isSuccess, isLoading } =
useListdistrictQuery()
  const [successShow, setSuccessShow] = useState(false)
  const [message, setMessage] = useState("")
  const [errorShow, setErrorShow] = useState(false)
  const [error, setError] = useState([])
  const navigate = useNavigate()
  const editParams = useParams()


  console.log(createBlock)
  const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
    initialValues: initialValues,
    validationSchema: blockSchema,
    onSubmit: async (values, action) => {
      Object.keys(values).map(e => values[e] =
values[e].toLowerCase())


      try {
        const response = await createBlock(values)
        if (response.data) {
          setErrorShow(false)
          setSuccessShow(true)
          setMessage("you have add successfully")
```

```
            action.resetForm()
            setTimeout(() => {
              setSuccessShow(false)
            }, 1000)
          }
          if (response.error.status == 406) {
            setErrorShow(true)
            setError(Object.values(response.error.data))
          }
        }
        catch {
          console.log("server down")
        }
        return values
      },
    })
    console.log(values)
    return (
      <>
        <Navbar_admin />
        {isLoading ? (<Spinner animation="grow" variant="info"
/>) : (
          <Container >
            <hr />
            <h5>Add Block</h5>
```

```
<hr />
<br />
<Row style={{ display: 'flex' }}>
  <Col sm={2}>
    <Sidebar />
  </Col>
  <Col sm={10}>
    <Alert show={successShow} variant="success">{message}</Alert>
    <Form className='common_css' onSubmit={handleSubmit} >
      <Row className="mb-3">
        <Alert show={errorShow} variant="danger"><ul>{error.map(error => <li>{error}</li>)}</ul></Alert>
        <Form.Group as={Col} controlId="formBasicblock">
          <Form.Label>Block Name</Form.Label>
          <Form.Control type="text" name="block" value={values.block} onChange={handleChange} onBlur={handleBlur} placeholder="" />
          {errors.block && touched.block ? (<p className="text-danger">{errors.block}</p>) : null}
        </Form.Group>
      </Row>
      <Row>
```

```
                        <Form.Group as={Col}
controlId="formBasicbdist">
                          <Form.Label>District</Form.Label>
                          <Form.Select name="district"
value={values.district} onChange={handleChange}
onBlur={handleBlur} aria-label="Default select example" >
                            <option>---select---</option>
                            {districtList.map(e =>
                              <option
value={e.district}>{e.district}</option>
                            )}
                          </Form.Select>
                        </Form.Group>
                      </Row>
                      <br />
                      <Button type="submit" name="save"
variant="primary" className="m-1" size="sm">
                        Save
                      </Button>
                    </Form>
                  </Col>
                </Row>
              </Container>
            )}
          </>
        ) }
```



****

ADMIN ENTRIES (BLOCK (VIEWING))

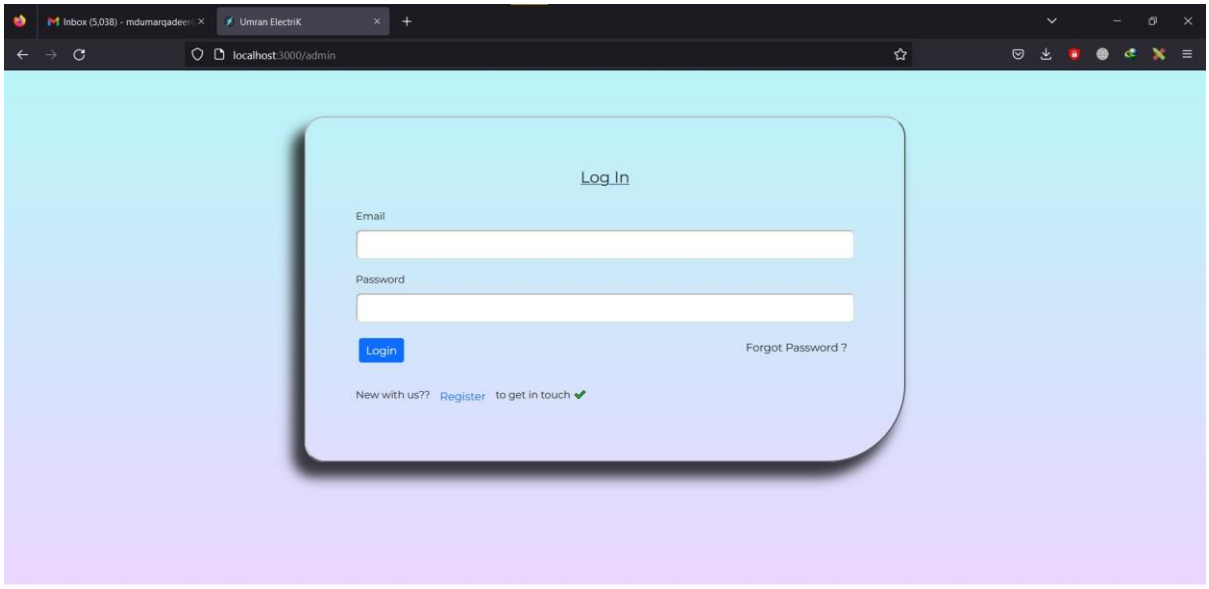
```
// import {Link,Route,Routes} from 'react-router-dom';
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Table from 'react-bootstrap/Table';
import Col from 'react-bootstrap/Col';
import Row from 'react-bootstrap/Row';
import { Navbar_admin } from './Navbarr';
import { Sidebar } from './Sidebar';
import Nav from 'react-bootstrap/Nav';
```

```
import { NavLink } from 'react-router-dom';
import { useDeleteblockMutation, useListblockQuery } from '../../services/admin/blockService';
import { useNavigate } from 'react-router-dom';
import Spinner from 'react-bootstrap/Spinner';
export function Block_d() {
  const { data: blockList, isLoading, isSuccess } = useListblockQuery()
  const navigate = useNavigate()
  const [blockDelete, blockDeleteFlag] = useDeleteblockMutation()
  console.log(blockList)
  return (
    <>
      <Navbar_admin />
      {isLoading ? (
        <>
          <Spinner animation="grow" variant="info" />
          <Spinner animation="grow" variant="success" />
          <Spinner animation="grow" variant="danger" />
          <Spinner animation="grow" variant="warning" />
          <Spinner animation="grow" variant="info" />
          <Spinner animation="grow" variant="light" />
        </>
      ) : (
```

```
<Container >
  <hr />
  <h5>Block Details</h5>
  <hr />
  <br />
  <Row style={{ display: 'flex' }}>
    <Col sm={2}>
      <Sidebar />
    </Col>
    <Col sm={10}>
      <Form className='common_csss' >
        <Row className="mb-3">
          <Button style={{ "background": "none",
"border": "none" }} variant="light" className="m-1" size="sm">
            <Nav.Link style={{ float: 'right' }}
as={NavLink} to="/admin/block">    Add Block   </Nav.Link>
          </Button>
          <Table bordered hover size="sm">
            <thead>
              <tr>
                <th>Serial No.</th>
                <th>Block</th>
                <th>District</th>
                <th>Action</th>
              </tr>
```

```
                    </thead>
                    <tbody>
                      {blockList ? blockList.map(e => (
                        <tr>
                          <td>{e.id}</td>
                          <td>{e.block}</td>
                          {e.district ? (<td>{e.district}</td>) :
(<td>null</td>)}
                          <td>
                            <Button style={{ "background":
"none", "border": "none" }} onClick={() =>
navigate(`/admin/block/edit/${e.id}`)} variant="light"
className="m-1" size="sm">
                              
                            </Button>
                            <Button style={{ "background":
"none", "border": "none" }} onClick={() => blockDelete(e.id)}
variant="light" className="m-1" size="sm">
                              ✕
                            </Button>

                          </td>
                        </tr>
                      )) : null}
                    </tbody>
                  </Table>
```

```
                    </Row>
                  </Form>
                </Col>
              </Row>
            </Container>
          )}
        </>
    )
}
```



****

Submitted By Imran & Umar

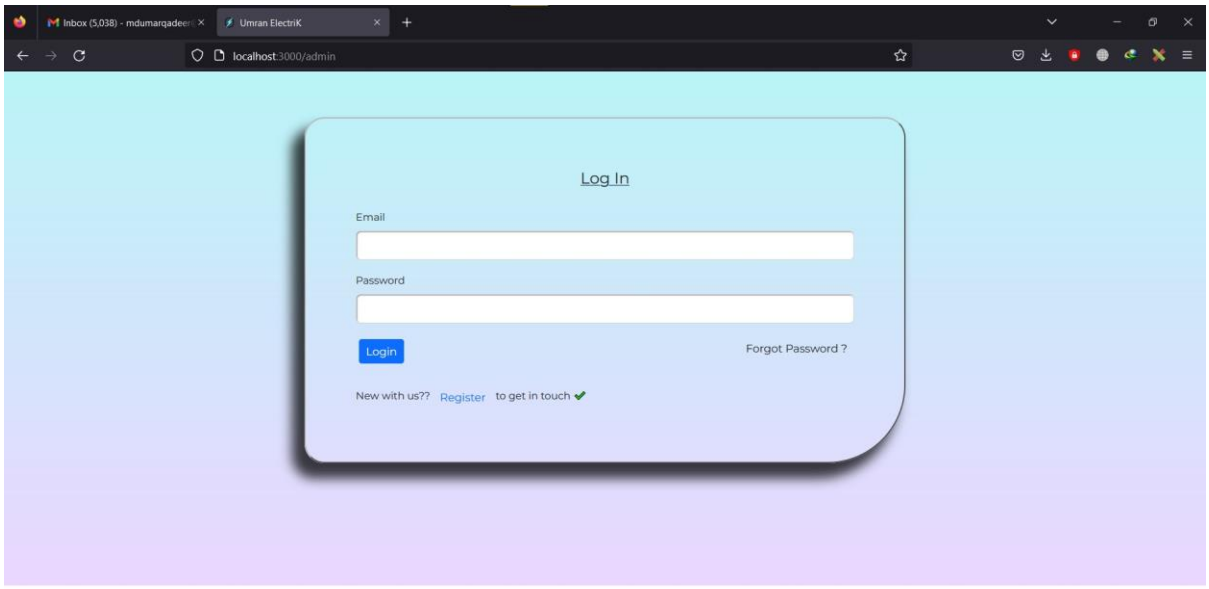
ADMIN FORGET PASSWORD

```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Nav from 'react-bootstrap/Nav';
import { NavLink } from 'react-router-dom';
import { Navbar_admin } from './Navbarr';
import { useFormik } from 'formik';
import { useEffect, useState, useRef, useMemo } from 'react';
import { useNavigate, useParams } from 'react-router-dom';
import { useUpdateadminSignUpMutation } from
'../../services/admin/adminSignUpService';
import { forgetPasswordSchema } from
'../validation/admin/validation';
import Alert from 'react-bootstrap/Alert';
const initialValues = {
   new_password: '',
   confirm_password: '',
}
export function AdminForgetPassword() {
   const [successShow, setSuccessShow] = useState(false)
   const [message, setMessage] = useState("")
   const [errorShow, setErrorShow] = useState(false)
   const [error, setError] = useState([])
```

```
  const navigate = useNavigate()
  const [updateAdmin, updateAdminFlag] =
useUpdateadminSignUpMutation()
  const id = localStorage.getItem("userforget")
  console.log(id)
  const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
    initialValues: initialValues,
    validationSchema: forgetPasswordSchema,
    onSubmit: async (values, action) => {
      try {
        const form = new FormData()
        form.append('password', values.new_password)
        form.append('id', id)
        const response = await updateAdmin(form)
        if (response.data) {
          setErrorShow(false)
          setSuccessShow(true)
          setMessage("you have changed password
successfully")
          setTimeout(() => {
            setSuccessShow(false)
            action.resetForm()
            navigate('/admin')
          }, 1000)
```

```
            }
            if (response.error.status == 406) {
              setErrorShow(true)
              setError(Object.values(response.error.data))
            }
          }
          catch {
            console.log("server down")
          }
          return values
        },
      })
      return (
        <>
          <Container>
            <Alert show={successShow}
variant="success">{message}</Alert>
            <Form className='common_css'
onSubmit={handleSubmit}>
              <Alert show={errorShow}
variant="danger"><ul>{error.map(error =>
<li>{error}</li>)}</ul></Alert>
              <h5>Change Password</h5>
              <br />
              <Form.Group className="mb-3"
controlId="formBasicNewpass">
```

```
                <Form.Label>Enter New Password</Form.Label>
                <Form.Control type="password"
name="new_password" value={values.new_password}
onChange={handleChange} onBlur={handleBlur} placeholder=""
/>
                {errors.new_password && touched.new_password ?
(<p className="text-danger">{errors.new_password}</p>) :
null}
            </Form.Group>
            <Form.Group className="mb-3"
controlId="formBasicNewpass2">
                <Form.Label>Confirm New Password</Form.Label>
                <Form.Control type="password"
name="confirm_password" value={values.confirm_password}
onChange={handleChange} onBlur={handleBlur} placeholder=""
/>
                {errors.confirm_password &&
touched.confirm_password ? (<p className="text-
danger">{errors.confirm_password}</p>) : null}
            </Form.Group>
            <Button variant="primary" type="submit"
className="m-1" size="sm">
                Submit
            </Button>
          </Form>
        </Container>
      </> ) }
```



ADMIN CHANGE PASSWORD

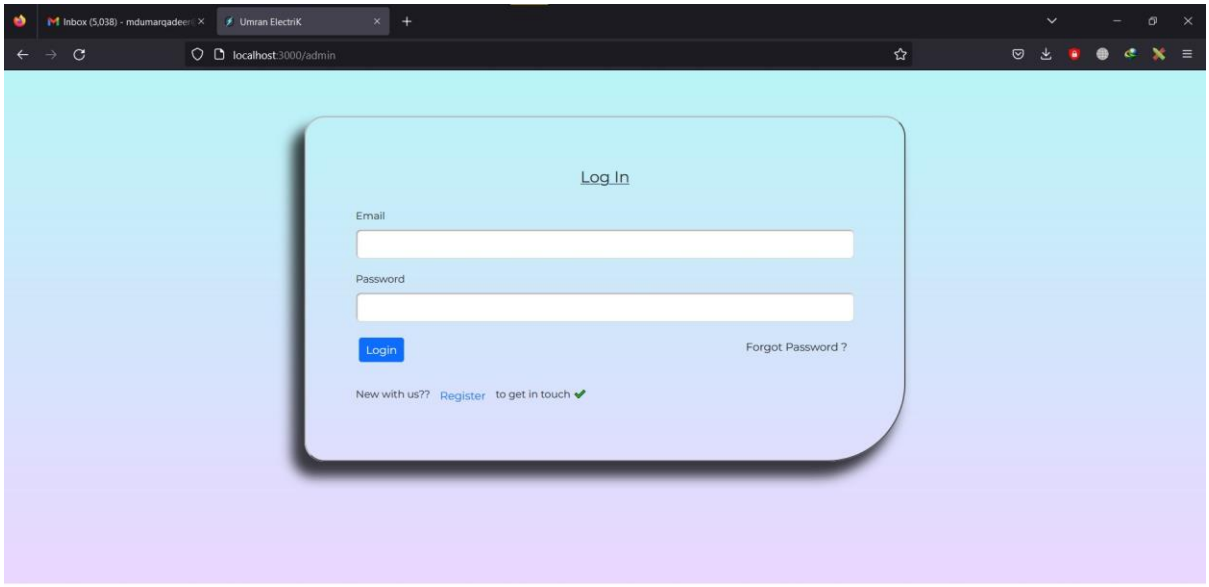
```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Nav from 'react-bootstrap/Nav';
import { NavLink } from 'react-router-dom';
import { Navbar_admin } from './Navbarr';
import { useFormik } from 'formik';
import { useEffect, useState, useRef, useMemo } from 'react';
import { useNavigate, useParams } from 'react-router-dom';
import { useCreateadminChangePasswordMutation } from
'../../services/admin/adminChangePasswordService';
import { changePasswordSchema } from
'../validation/admin/validation';
```

```
import Alert from 'react-bootstrap/Alert';
const initialValues = {
   password: '',
   new_password: '',
   confirm_password: '',
   access: ''
}
export function AdminChangePassword() {
   const [successShow, setSuccessShow] = useState(false)
   const [message, setMessage] = useState("")
   const [errorShow, setErrorShow] = useState(false)
   const [error, setError] = useState([])
   const navigate = useNavigate()
   const [updateAdmin, updateAdminFlag] =
useCreateadminChangePasswordMutation()
   const { refresh, access } =
JSON.parse(localStorage.getItem("user"))
   const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
      initialValues: initialValues,
      validationSchema: changePasswordSchema,
      onSubmit: async (values, action) => {
         try {
            values.access = access
            const response = await updateAdmin(values)
```

```
            if (response.data) {
              setErrorShow(false)
              setSuccessShow(true)
              setMessage("you have changed password
successfully")
              setTimeout(() => {
                setSuccessShow(false)
                action.resetForm()
                navigate('/admin/dashboard')
              }, 1000)
            }
            if (response.error.status == 406) {
              setErrorShow(true)
              setError(Object.values(response.error.data))
            }
          }
          catch {
            console.log("server down")
          }
          return values
        },
      })
      return (
        <>
          <Navbar_admin />
```

```
<Container>
<Alert show={successShow} variant="success">{message}</Alert>
<Form className='common_css' onSubmit={handleSubmit}>
<Alert show={errorShow} variant="danger"><ul>{error.map(error => <li>{error}</li>)}</ul></Alert>
<h5>Change Password</h5>
<br />
<Form.Group className="mb-3" controlId="formBasicOldpass">
<Form.Label>Enter Old Password</Form.Label>
<Form.Control type="password" name="password" value={values.password} onChange={handleChange} onBlur={handleBlur} placeholder="" />
{errors.password && touched.password ? (<p className="text-danger">{errors.password}</p>) : null}
</Form.Group>
<Form.Group className="mb-3" controlId="formBasicNewpass">
<Form.Label>Enter New Password</Form.Label>
<Form.Control type="password" name="new_password" value={values.new_password} onChange={handleChange} onBlur={handleBlur} placeholder="" />
{errors.new_password && touched.new_password ? (<p className="text-danger">{errors.new_password}</p>) : null}
```

```
            </Form.Group>
            <Form.Group className="mb-3"
controlId="formBasicNewpass2">
              <Form.Label>Confirm New Password</Form.Label>
              <Form.Control type="password"
name="confirm_password" value={values.confirm_password}
onChange={handleChange} onBlur={handleBlur} placeholder=""
/>
              {errors.confirm_password &&
touched.confirm_password ? (<p className="text-
danger">{errors.confirm_password}</p>) : null}
            </Form.Group>
<Button variant="primary" type="submit" className="m-1"
size="sm">
              Submit
            </Button>
          </Form>
        </Container>
      </>
    )
}
```



****

OFFICER LOGIN

```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import Nav from 'react-bootstrap/Nav';
import { NavLink, useNavigate } from 'react-router-dom';
// import tw from '../slider/tw.png';
import { useCookies } from 'react-cookie'
import { useFormik } from 'formik';
```

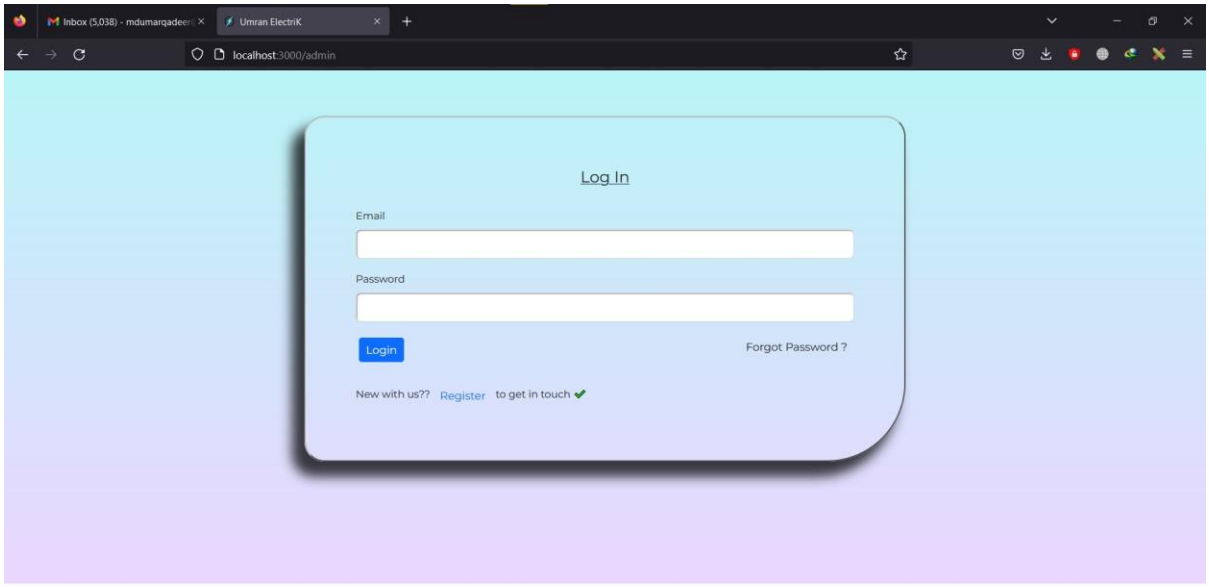
Submitted By Imran & Umar

```
import Alert from 'react-bootstrap/Alert'
import { useState, useMemo, useEffect, useCallback } from 'react'
import { useCreateofficerSignInMutation } from
'../../services/officer/officerSignInService';
import { officerSignInSchema } from
'../validation/officer/validation';
import AsyncLocalStorage from '@createnextapp/async-local-
storage'
const initialValues = {
   email: "",
   password: ""
}
export function OfficerSignIn() {
   const [signin, data] = useCreateofficerSignInMutation()
   const [show, setShow] = useState(false);
   const [alert, setAlert] = useState("")
   const navigate = useNavigate()
   const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
      initialValues: initialValues,
      validationSchema: officerSignInSchema,
      onSubmit: async (values, action) => {
         Object.keys(values).forEach((key) => {
            values[key] = values[key].trim()
         })
         try {
```

```
            const response = await signin(values)
            if (response.data) {
              setShow(false)
              const token = response.data
              try {
                const d = await AsyncLocalStorage.setItem('officer',
JSON.stringify(token))
              }
              catch (error) {
                console.log(error)
              }
              //
localStorage.setItem("officer",JSON.stringify(token))
              setTimeout(() => {
                navigate('/officer/dashboard')
              }, [1000])
            }
            if (response.error) {
              if (response.error.status === 401) {
                setShow(true)
                setAlert("email or password incorrect")
              }
            }
          }
          catch {
```

```
            console.log("error")
          }
        }
    })
    const closeTab = () => {
      window.opener = null;
      window.open("", "_self");
      window.close();
    }
    return (
  <>
          <Form className='common_css sign_in_form'
onSubmit={handleSubmit}>
            <Alert show={show} variant='danger'>{alert}</Alert>
            <h5>Log In</h5>
            <br />
            <Form.Group className="mb-3"
controlId="formBasicEmail" >
              <Form.Label>Email</Form.Label>
              <Form.Control type="text" name="email"
value={values.email} onChange={handleChange}
onBlur={handleBlur} placeholder="" />
              {errors.email && touched.email ? (<p className='text-
danger'>{errors.email}</p>) : null}

            </Form.Group>
```

```
            <Form.Group className="mb-3"
controlId="formBasicPassword">
                <Form.Label>Password</Form.Label>
                <Form.Control type="password" name="password"
value={values.password} onChange={handleChange}
onBlur={handleBlur} placeholder="" />
                {errors.password && touched.password ? (<p
className='text-danger'>{errors.password}</p>) : null}
            </Form.Group>


            <Button variant="primary" type="submit"
className="m-1" size="sm">
                Login
            </Button>
            <Button variant="Light" style={{ float: 'right' }}
size="sm">
                <Nav.Link as={NavLink} onClick={closeTab}
to="/officer/check_forgetpassword">Forgot Password
?</Nav.Link>
            </Button>
            <br /><br />
        </Form>
    </>
  )
}
```



****

OFFICER SIGNUP

```
import Container from 'react-bootstrap/Container';
import Form from 'react-bootstrap/Form';
import Button from 'react-bootstrap/Button';
import { Navbar_admin } from './Navbarr';
import Alert from 'react-bootstrap/Alert';
import Col from 'react-bootstrap/Col';
import Row from 'react-bootstrap/Row';
import Nav from 'react-bootstrap/Nav';
import { NavLink } from 'react-router-dom';
import { useCreateofficerSignUpMutation } from
'../../services/officer/officerSignUpService';
import { useRef, useState } from 'react'
```

```
import swal from 'sweetalert'
import { useFormik } from 'formik'
import { officerSignUpSchema } from
'../validation/admin/validation';
import { useListdivisionQuery } from
'../../services/admin/divisionService';
import { useGetByIdsubdivisionQuery } from
'../../services/admin/subDivisionService';
import { useGetByIdsectionQuery } from
'../../services/admin/sectionService';


const initialValues = {
   name: '',
   email: '',
   phone: '',
   designation: '',
   division: "",
   subdivision: "",
   section: "",
   password: '',
   confirm_password: '',
}
export function OfficerSignUp() {
   const [divisionid, setDivisionid] = useState(null)
   const [subdivisionid, setSubdivisionid] = useState(null)
   const division = useListdivisionQuery()
```

```
  const subdivision = useGetByIdsubdivisionQuery(divisionid)
  const section = useGetByIdsectionQuery(subdivisionid)
  const [signup, data] = useCreateofficerSignUpMutation()
  const [show, setShow] = useState(false);
  const [error, setError] = useState([])
  const [successShow, setSuccessShow] = useState(false)
  const [message, setMessage] = useState("")
  const { values, errors, handleChange, handleSubmit, touched,
handleBlur } = useFormik({
    initialValues: initialValues,
    validationSchema: officerSignUpSchema,
    onSubmit: async (values, action) => {
      Object.keys(values).forEach((key) => {
        if (values[key] != true && values[key] != false) {
          values[key] = values[key].trim()
        }
      })
      values.is_staff = true
      const response = await signup(values)
      console.log(response)
      if (response.data) {
        setShow(false)
        setSuccessShow(true)
        setMessage("you have SignUp successfully")
        action.resetForm()
```

```
            setTimeout(() => {
                setSuccessShow(false)
            }, 1000)

        }
        else if (response.error.status === 406) {
            const e = Object.values(response.error.data)
            setShow(true)
            const currentError = []
            e.map(e => {
                currentError.push(...e)
            })
            setError(currentError)

        }

    },
})
  return (
    <>
      <Navbar_admin />
      <Container>
        <Form className='common_css'
onSubmit={handleSubmit}>
```

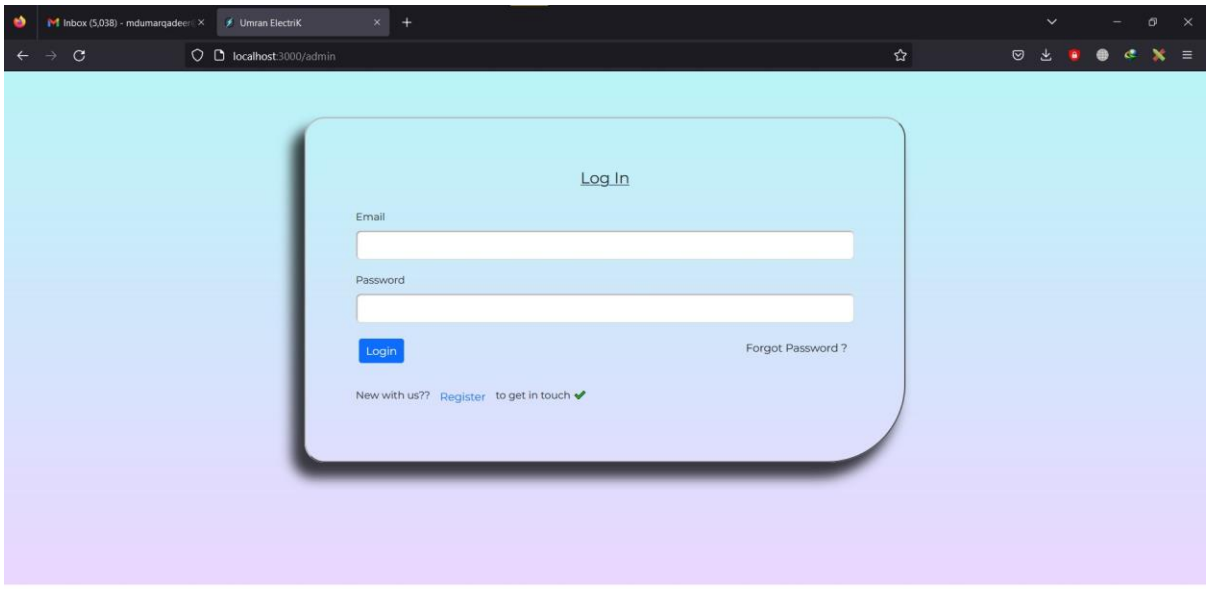
```
            <Alert show={show}
variant='danger'><ul>{error.map(error =>
<li>{error}</li>)}</ul></Alert>

            <h5>Register With Us</h5>

            <br />

            <Form.Group className="mb-3"
controlId="formBasicEmail" >

              <Form.Label>Name <span className='text-
danger'>*</span></Form.Label>

              <Form.Control type="text" name="name"
value={values.name} placeholder="" onChange={handleChange}
onBlur={handleBlur} />

              {errors.name && touched.name ? (<p
className='text-danger'>{errors.name}</p>) : null}

            </Form.Group>

            <Form.Group className="mb-3"
controlId="formBasicEmail">

              <Form.Label>Email <span className='text-
danger'>*</span></Form.Label>

              <Form.Control type="email" name="email"
value={values.email} placeholder="" onChange={handleChange}
onBlur={handleBlur} />

              {errors.email && touched.email ? (<p
className='text-danger'>{errors.email}</p>) : null}

            </Form.Group>

            <Form.Group className="mb-3"
controlId="formBasicEmail">

              <Form.Label>Mobile No <span className='text-
danger'>*</span></Form.Label>
```

```
                <Form.Control type="text" name="phone"
value={values.phone} placeholder="" onChange={handleChange}
onBlur={handleBlur} />
                {errors.phone && touched.phone ? (<p
className='text-danger'>{errors.phone}</p>) : null}
            </Form.Group>
            <Row className="mb-3">
              <Form.Group as={Col} controlId="formGridState">
                <Form.Label>Designation</Form.Label>
                <Form.Select name="designation"
value={values.designation} onChange={handleChange}
onBlur={handleBlur}>
                  <option>---select---</option>
                  <option value="officer">officer</option>
                </Form.Select>
              </Form.Group>
              <Form.Group as={Col}>
                <Form.Label>Division<span className="text-
danger">*</span></Form.Label>
                <Form.Select aria-label="Default select example"
name="division" value={values.division}
onChange={handleChange} onBlur={handleBlur} size="sm">
                  <option>---select---</option>
                  {division.data ? division.data.map(e => (
                    <option value={e.division} onClick={() =>
setDivisionid(e.id)}>{e.division}</option>
                  )) : null}
```

```
                </Form.Select>
                {errors.division && touched.division ? (<p
className='text-danger'>{errors.division}</p>) : null}
              </Form.Group>
            </Row>
            <Row className="mb-3">
              <Form.Group as={Col}>
                <Form.Label>Sub Division<span className="text-
danger">*</span></Form.Label>
                <Form.Select aria-label="Default select example"
name="subdivision" value={values.subdivision}
onChange={handleChange} onBlur={handleBlur} size="sm">
                  <option>---select---</option>
                  {subdivision.data ? subdivision.data.map(e => (
 <option value={e.subdivision} onClick={() =>
setSubdivisionid(e.id)}>{e.subdivision}</option>
                  )) : null}
                </Form.Select>
                {errors.subdivision && touched.subdivision ? (<p
className='text-danger'>{errors.subdivision}</p>) : null}
              </Form.Group>
              <Form.Group as={Col}>
                <Form.Label>Section<span className="text-
danger">*</span></Form.Label>
                <Form.Select aria-label="Default select example"
name="section" value={values.section}
onChange={handleChange} onBlur={handleBlur} size="sm">
```

```
                    <option>---select---</option>
                    {section.data ? section.data.map(e => (
                      <option value={e.section}
>{e.section}</option>
                    )) : null}
                  </Form.Select>
                  {errors.section && touched.section ? (<p
className='text-danger'>{errors.section}</p>) : null}
                </Form.Group>
              </Row>
              <Form.Group className="mb-3"
controlId="formBasicPassword">
                <Form.Label>Password <span className='text-
danger'>*</span></Form.Label>
                <Form.Control type="password" name="password"
value={values.password} placeholder=""
onChange={handleChange} onBlur={handleBlur} />
                {errors.password && touched.password ? (<p
className='text-danger'>{errors.password}</p>) : null}
              </Form.Group>
              <Form.Group className="mb-3"
controlId="formBasicPassword">
                <Form.Label>Re-enter Password <span
className='text-danger'>*</span></Form.Label>
                <Form.Control type="password"
name="confirm_password" value={values.confirm_password}
onChange={handleChange} onBlur={handleBlur} placeholder=""
/>
```

```
                {errors.confirm_password &&
touched.confirm_password ? (<p className='text-
danger'>{errors.confirm_password}</p>) : null}
            </Form.Group>
            <Button type="submit" variant="primary"
className="m-1" size="sm">
              Register
            </Button>
          </Form>
          <ALERT SHOW={SUCCESSSHOW}
VARIANT="SUCCESS">{MESSAGE}</ALERT>
        </CONTAINER>
      </>
    )
}
```



Submitted By Imran & Umar

OFFICER EDIT PROFILE

```
import Container from 'react-bootstrap/Container';
import Button from 'react-bootstrap/Button';
import { Navbar_admin } from '../admin/Navbarr';
import Table from 'react-bootstrap/Table';
import Form from 'react-bootstrap/Form';
import Nav from 'react-bootstrap/Nav';
import { NavLink, useNavigate } from 'react-router-dom';
import Col from 'react-bootstrap/Col';
import Row from 'react-bootstrap/Row';
import { useRef, useMemo, useState } from 'react'
import { useFormik } from 'formik'
import { editProfileSchema } from
'../validation/applicant/validation';
import { useListadminGetUserQuery } from
'../../services/officer/officerSignUpService';
import { useUpdateofficerSignUpMutation } from
'../../services/officer/officerSignUpService';
import Alert from 'react-bootstrap/Alert';
import { OfficerNavbar } from './OfficerNavbar';
import Image from 'react-bootstrap/Image'
const initialValues = {
  id: '',
  name: '',
```

```
  email: '',
  phone: '',
  photo: ''
}
export function OfficerEditProfile() {
  const { refresh, access = null } = localStorage.getItem("officer") ?
JSON.parse(localStorage.getItem("officer")) : {}
  const { data: getUser, isSuccess, isLoading } =
useListadminGetUserQuery(access)
  const [successShow, setSuccessShow] = useState(false)
  const [message, setMessage] = useState("")
  const [errorShow, setErrorShow] = useState(false)
  const [error, setError] = useState([])
  const inputRef = useRef()
  const navigate = useNavigate()
  const [preview, setPreview] = useState(null)
  const [updateAdmin, updateAdminFlag] =
useUpdateofficerSignUpMutation()
  const { values, errors, handleChange, handleSubmit, touched,
handleBlur, setFieldValue } = useFormik({
    initialValues: initialValues,
    validationSchema: editProfileSchema,
    onSubmit: async (values, action) => {
      const form = new FormData()
      console.log(values)
      form.append('name', values.name)
```

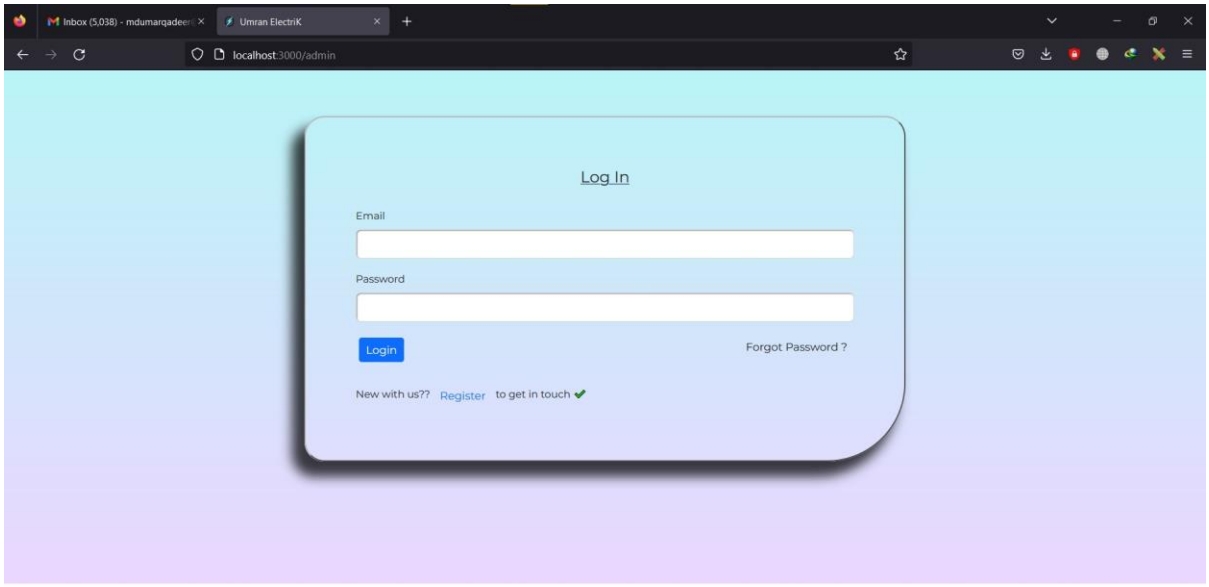
```
form.append('email', values.email)
form.append('phone', values.phone)
if (values.photo) {
 form.append('photo', values.photo)
}
form.append('id', values.id)
try {
 console.log(form.get("photo"))
 const response = await updateAdmin(form)
 console.log(response)
 if (response.data) {
  setErrorShow(false)
  setSuccessShow(true)
  setMessage("you have update successfully")
  setTimeout(() => {
   setSuccessShow(false)
   action.resetForm()
   navigate('/officer/dashboard')
  }, 1000)
 }
 if (response.error.status == 406) {
  setErrorShow(true)
  setError(Object.values(response.error.data))
 }
}
```

```
    catch {
      console.log("server down")
    }
    return values
  }
})
const handleClick = () => {
  inputRef.current.click();
}
const get = useMemo(() => {
  if (getUser) {
    console.log(getUser)
    initialValues.id = getUser.id.toString()
    initialValues.name = getUser.name
    initialValues.email = getUser.email
    initialValues.phone = getUser.phone
    setPreview(getUser.photo)
  }
}, [isSuccess, getUser])
if ('File' in window && values.photo instanceof File) {
  const reader = new FileReader()
  reader.readAsDataURL(values.photo)
  reader.onload = () => {
    setPreview(reader.result)
  }
```

```
 }
 return (
  <>
   <OfficerNavbar />
   <Alert show={successShow}
variant="success">{message}</Alert>
   <Container >
    <Alert show={errorShow}
variant="danger"><ul>{error.map(error =>
<li>{error}</li>)}</ul></Alert>
    <Image
     style={{ float: 'left' }}
     className="user_pic"
     src={preview}
     alt="logo"
     roundedCircle >
    </Image><span> <Button onClick={handleClick}
variant="">+</Button>
    </span>
   <Form className='common_css' onSubmit={handleSubmit}>
     <h5>Edit User Profile</h5>
     <br />
     <Form.Group className="mb-3"
controlId="formGridemail">
      <Form.Label>Name</Form.Label>
```

```
        <Form.Control style={{ width: '55%' }} name="name"
value={values.name} onChange={handleChange}
onBlur={handleBlur} size="sm" type="text" placeholder="" />

        {errors.name && touched.name ? (<p className='text-
danger'>{errors.name}</p>) : null}

      </Form.Group>

      <Form.Group className="mb-3"
controlId="formGridemail">

        <Form.Label>Email</Form.Label>

        <Form.Control style={{ width: '55%' }} name="email"
value={values.email} onChange={handleChange}
onBlur={handleBlur} size="sm" type="email" placeholder="" />

        {errors.email && touched.email ? (<p className='text-
danger'>{errors.email}</p>) : null}

      </Form.Group>

      <Form.Group className="mb-3"
controlId="formGridemail">

        <Form.Label>Phone</Form.Label>

        <Form.Control style={{ width: '55%' }} size="sm"
name="phone" value={values.phone} onChange={handleChange}
onBlur={handleBlur} type="text" placeholder="" />

        {errors.phone && touched.phone ? (<p className='text-
danger'>{errors.phone}</p>) : null}

      </Form.Group>

      <Form.Group className="mb-3" controlId="formGridAdd">

        <Form.Control accept='image/*' ref={inputRef}
name="photo" onChange={(event) => { setFieldValue("photo",
event.target.files[0]) }} style={{ width: '15%', display: 'none' }}
size="sm" type="file" />
```

```
        </Form.Group>
        <Button type="submit" variant="primary" size="sm">
         Submit
        </Button>
       </Form>
     </Container>
     </>
  )
}
```

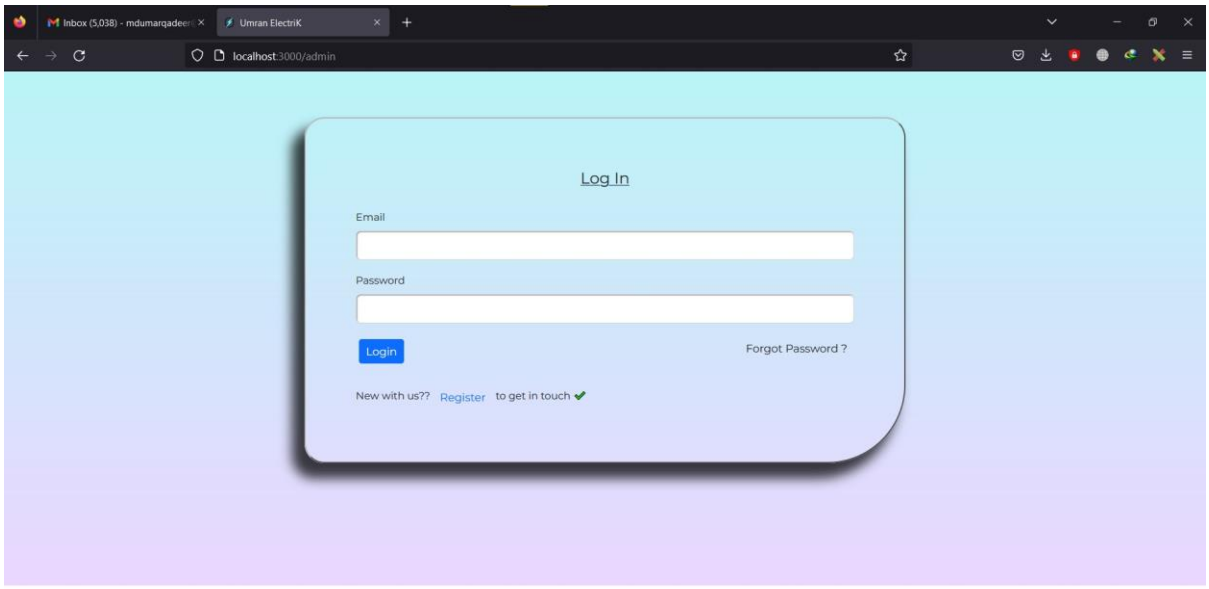


****

OFFICER DASHBOARD

Submitted By Imran & Umar

```
import { OfficerNavbar } from "./OfficerNavbar"
import AsyncLocalStorage from '@createnextapp/async-local-storage'
import { useEffect} from "react"
import { useNavigate } from "react-router-dom"
export function OfficerDashboard(){
  return(
    <>
    <OfficerNavbar/>
    </>
  ) }
```



****

OFFICER NAVBAR

```
import Container from 'react-bootstrap/Container';
```

Submitted By Imran & Umar

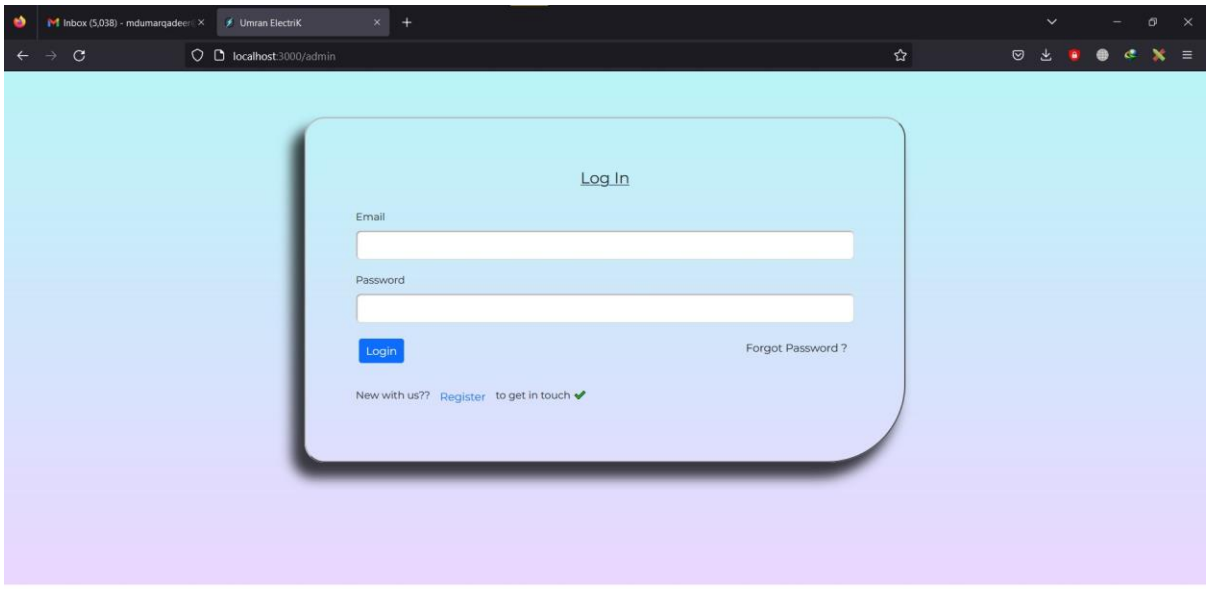
```
import Nav from 'react-bootstrap/Nav';
import Navbar from 'react-bootstrap/Navbar';
import NavDropdown from 'react-bootstrap/NavDropdown';
import { NavLink, Link } from 'react-router-dom';
import a from '../../slider/a.png'
import { useListadminGetUserQuery } from
'../../services/officer/officerSignUpService';
import Spinner from 'react-bootstrap/Spinner';
import Button from 'react-bootstrap/Button';
import AsyncLocalStorage from '@createnextapp/async-local-
storage'
import Image from 'react-bootstrap/Image'
export function OfficerNavbar() {
  const { refresh, access = null } = localStorage.getItem("officer") ?
JSON.parse(localStorage.getItem("officer")) : {}
  const { data: getUser, isSuccess, isLoading } =
useListadminGetUserQuery(access)
  return (
    <>
     {isLoading ? (
       <>
         <Spinner animation="grow" variant="info" />
       </>
     ) : (
       <Navbar className="navbarr_background" expand="lg">
         <Container>
```

```
<Navbar.Brand as={NavLink} to="/" end>
 <img
  className="logo"
  src={a}
  alt="logo"/>
 Umran ElectriK
</Navbar.Brand>
<Navbar.Toggle aria-controls="basic-navbar-nav" />
<Navbar.Collapse id="basic-navbar-nav">
 <Nav className="me-auto">
  <Nav.Link as={NavLink} to="/officer/dashboard" end>Go
to Home Page</Nav.Link>
  <Nav.Link as={NavLink}
to="/application/verify">Application Verify</Nav.Link>
  <Nav.Link as={NavLink}
to="/application/reverify">Application Reverify</Nav.Link>
 </Nav>
</Navbar.Collapse>
{getUser ? (
 <Nav>
  <Image
   className="user_pic"
   src={getUser.photo}
   alt="logo"
   roundedCircle          >
```

```
            </Image>
            <NavDropdown title={getUser.name} id="basic-nav-
dropdown"  >
              <NavDropdown.Item as={Link}
to="/officer/edit_profile">
                Edit Profile
              </NavDropdown.Item>
              <NavDropdown.Item as={Link}
to="/officer/change_password">
                Change Password
              </NavDropdown.Item>
              <NavDropdown.Divider />
              <NavDropdown.Item as={Link} to="/" onClick={() =>
AsyncLocalStorage.removeItem('officer')}>
                Logout
              </NavDropdown.Item>
            </NavDropdown>
          </Nav>
        ) : (
          <Button as={Link} to="/officer" variant="danger"
size="sm">Sign in</Button>
        )}
      </Container>
    </Navbar>
  )}
</>
```

)

}



****

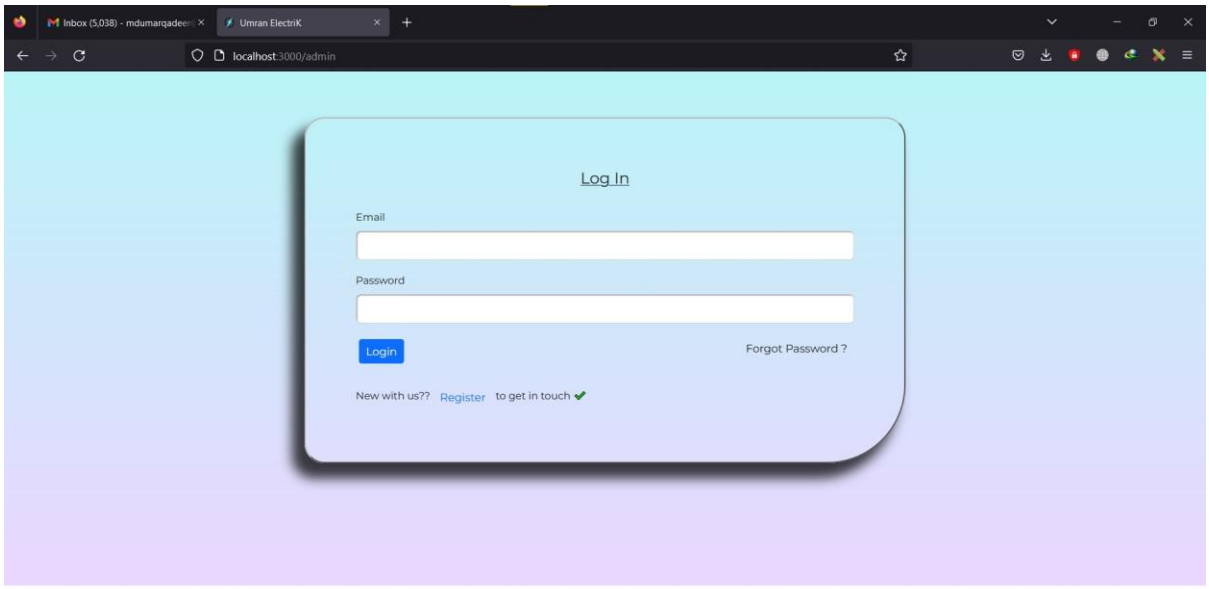
OFFICER APPLICATION VERIFY

```
// import { Link, Route, Routes } from 'react-router-dom';
import Container from 'react-bootstrap/Container';
import { OfficerNavbar } from './OfficerNavbar';
import Form from 'react-bootstrap/Form';
import Table from 'react-bootstrap/Table';
import { NavLink } from 'react-router-dom';
import Nav from 'react-bootstrap/Nav';
import Spinner from 'react-bootstrap/Spinner';
```

```
import { useListapplicationQuery } from
'../../services/applicant/appicantionService';
export function ApplicationVerify() {
  var { data, isLoading, isSuccess } = useListapplicationQuery()
  if (data) {
    data = data.filter(e => e.is_verified === false && e.is_defected
=== false && e.is_rejected === false)
  }
  return (
    <>
      {isLoading ? (
        <>
          <Spinner animation="grow" variant="info" />
        </>
      ) : (
        <>
          <OfficerNavbar />
          <Container>
            <Form className='common_css'>
              <h5>Application Verification List</h5>
              <Table style={{ "text-align": "center" }} border="1"
center striped bordered hover size="lg">
                <thead>
                  <tr>
                    <th >Application No.</th>
```

```
                    <th >Name</th>
                    <th >Email</th>
                    <th >Phone</th>
                    <th >View</th>
                  </tr>
                </thead>
                <tbody >
                  {data.length !== 0 ? data.map(e => (
                    <tr>
                      <td>{e.request_no}</td>
                      <td>{e.user.name}</td>
                      <td>{e.user.email}</td>
                      <td>{e.user.phone} </td>
                      <td>
                        <Nav.Link as={NavLink} onClick={() =>
localStorage.setItem("applicationVerify",
JSON.stringify(e.request_no))} to="/application/verifydetail"
end>👁</Nav.Link>
                      </td>
                    </tr>
                  )) : (<p>No data found</p>)}
                </tbody>
              </Table>
            </Form>
          </Container>
```

```
            </>
        )}
      </>
    )
}
```



****

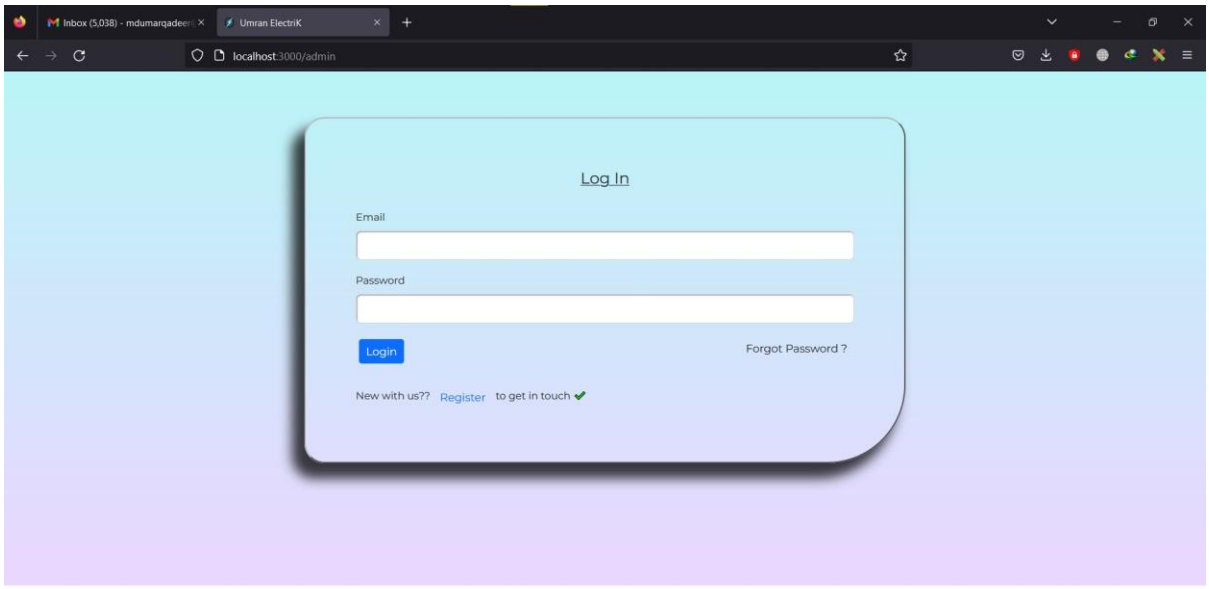
OFFICER RE VERFICATION

```
// import { Link, Route, Routes } from 'react-router-dom';
import Container from 'react-bootstrap/Container';
import { OfficerNavbar } from './OfficerNavbar';
import Form from 'react-bootstrap/Form';
import Table from 'react-bootstrap/Table';
```

```
import { NavLink } from 'react-router-dom';
import Nav from 'react-bootstrap/Nav';
import Spinner from 'react-bootstrap/Spinner';
import { useListapplicationQuery } from
'../../services/applicant/appicantionService';
export function ApplicationReVerify() {
  var { data, isLoading, isSuccess } = useListapplicationQuery()
  if (data) {
    data = data.filter(e => e.is_verified === false && e.is_defected
=== true && e.is_rejected === false)
  }
  return (
    <>
      {isLoading ? (
        <>
          <Spinner animation="grow" variant="info" />
          <Spinner animation="grow" variant="success" />
          <Spinner animation="grow" variant="danger" />
          <Spinner animation="grow" variant="warning" />
          <Spinner animation="grow" variant="info" />
          <Spinner animation="grow" variant="light" />
        </>
      ) : (
        <>
          <OfficerNavbar />
```

```
<Container>
  <Form className='common_css'>
    <h5>Application ReVerification List</h5>
    <br />
    <Table style={{ "text-align": "center" }} border="1"
center striped bordered hover size="lg">
      <thead>
        <tr>
          <th >Application No.</th>
          <th >Name</th>
          <th >Email</th>
          <th >Phone</th>
          <th >View</th>
        </tr>
      </thead>
      <tbody >
        {data.length !== 0 ? data.map(e => (
          <tr>
            <td>{e.request_no}</td>
            <td>{e.user.name}</td>
            <td>{e.user.email}</td>
            <td>{e.user.phone} </td>
            <td>
              <Nav.Link as={NavLink} onClick={() =>
localStorage.setItem("applicationVerify",
```

```
JSON.stringify(e.request_no))} to="/application/reverifydetail"
end>👁</Nav.Link>
                                </td>
                              </tr>
                            )) : (<p >No data found</p>)}
                          </tbody>
                        </Table>
                      </Form>
                    </Container>
                  </>
                )}
              </>    )   }
```



SYSTEM TESTING

Submitted By Imran & Umar

Software testing is an investigation conducted to provide stake holders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risk at implementation of the software. Test techniques include, but are limited to, the process of executing a program or application with the intent of finding software bucks. So, system testing is vital to the success of the system, The first step in the testing is to prepare a plan that tests all aspects of a system in a way that promotes its crediability among potential user.

TEST PLAN:-

The test plan is an important, but often overlooked, product of software design. A test plan prescribes various kinds of activities that will be performed to demonstrate that the software product meets its requirements.

TEST PLAN SPECIFIES

- The Objective Of Testing :-

It achieves error-free operation under stated conditions for the state period of time.

- The Test Completion Criteria:-

It helps to achieve a specified rate of error exposure, and a specified percent of logical path coverage.

- The System Integration Plan:-

It helps to specify strategy, schedule and responsible individuals of testing.

- Method To Be Used:-

It defines method of the test plan such as walkthroughs, inspections, static analysis, dynamic tests, formal verification and the particular test cases to be used.

TESTING LEVEL : Three levels of testing were carried out in the each phase:

 Unit Testing:- It tests a code function.
 Integration Function:-It checks independency of different modules.
 System Testing:- It tests the applications' functionality as was specified in the detailed functionality detail.

The testing level situation is illustrated below:-

Unit Testing:-

- The System Integration Plan:-

It helps to specify strategy, schedule and responsible individuals of testing.

- Method To Be Used:-

It defines method of the test plan such as walkthroughs, inspections, static analysis, dynamic tests, formal verification and the particular test cases to be used.

TESTING LEVEL : Three levels of testing were carried out in the each phase:

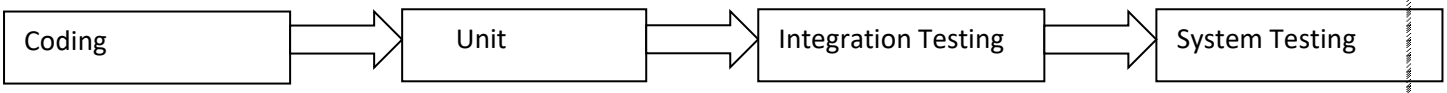
 Unit Testing:- It tests a code function.

 Integration Function:-It checks independency of different modules.

 System Testing:- It tests the applications' functionality as was specified in the detailed functionality detail.

The testing level situation is illustrated below:-

Unit Testing:-



Unit testing comprises the set of tests performed by an individual between different screens of the application in the same module or across module.

In the STOCK MANAGEMENT SYSTEM application, the sandwich integration of integration testing was implemented. Sandwich integration is predominantly Top-down, but bottom-up techniques are used on some modules and subsystems. This mix alleviates many of the problems integration at the subsystem and system level.

This integration tests are carried out and their result were recorded in the integration test report (ITR). The ITR gives success result after integration testing of STOCK MANAGEMENT SYSTEM application.

SYSTEM TESTING

System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. A small error can conceivably explode into a much larger problem. The best program is worthless if it does not meet user needs. Through system testing these problems can be reduced. System testing is actually series of different tests whose primary purpose is to fully exercise the computer-based system.

Although, each test has been properly integrated and perform the allocated functions.

Following major guidelines were taken under consideration while framing test cases for system testing:

- I anticipated the potentials interfacing problems that might occur in our application.

- I designed error-handling paths that tested all information coming from different elements of the system.

- I conducted a series of tests that simulated bad data or other potential errors at the software interface.

- Results of the entire test were recorded to use them as "Evidence" if finger pointing occurred.

MAINTAINANCE

It is the phase in which system are in place and operating, enhancements and/or modifications to the system are developed and tested & hardware and/or software is added or de-replaced. The system is monitored for continued performance in accordance with security requirements and needed system modifications are incorporated. The operational system is periodically assessed to determine how system made more effective, secure, and efficient. Operations continue as long as the system can be effectively adapted to respond to an organizations need. When necessary modifications or changes are identified the system may re-enter a previous phase of SDLC.
Key security activity include:

- Conduct an operational readiness review.
- Manage the configuration of the system.
- School processes and procedure for assured operational and continues monitoring of the information system security controls.
- Perform reauthorization as required.

CONCLUSION

The system has been developed with much care that it is free of errors and at the same time it is efficientand less time consuming. The important thing is that the system is robust and avoids malfunction from outsiders. It follows all the phases of software development cycle. So the product is accurate. As well as provision is provided for future development. The current application is developed in accordance with the request that has been provided by the organization.

LIMITATIONS

This software is developed for only Windows based environment. Actually not only Windows environment is sufficient rather Windows XP or onward version is needed, because here I have used JAVA as frontend and ORAVLE 10g Personal edition as backend, which doesn't support earlier versions of Windows.

Moreover some of the facts on which further assessments are required are:

1. The designed system is application based system.
2. Recovery management is less.
3. It is a single system.

MINIMUM HARDWARE REQUIREMENT

Computer System : Pentium Based System with at least 2.4 GHz speed or higher.

RAM : At least 128 MB, it is recommended to have 2565 MB of RAM for fast processing on data.

Hard Disk Space : At least 20GB of free space, it is recommended to have 40GB of free space for data storage.

MINIMUM SOFTWARE REQUIREMENT

Operating System : Windows XP or higher

RDBMS Packages : Oracle 10g pre installed

JDBC Driver : The system must have the capability of ODBC driver named as Microsoft JDBC for Oracle. There must exist an user DSN “LOCALHOST” in JDBC manager.

Submitted By Imran & Umar

FUTURE SCOPE

From the project point of view, it has several future scopes:
After the system is found to be working successfully as per the requirement. Further enhancement could be made in one of the following direction. These modifications can be in both ways, either hardware or software limitation or additional requirement made by the organization.
There is always room for the further enhancement of this system in the following areas:

- In future it can be turned into a web based application.

- More attention towards the recovery management can be done.

- More efficiency in the cost management can also be adopted.

- The designed system is single system, in future it can be converted into multiple system.

BIBLIOGRAPHY

Books reference:-

1. Software Engineering- Pressman

2. System Analysis And Design- Elias Awad

3. **Relational Database Management System-** B.C.Desai

4. **JAVA The Complete Reference-** Patrick Naughton and Herbert Schieldt

Websites I referred:-

1. GOOGLE

THANK YOU

Submitted By Imran & Umar