

SPRINT-1

TEAM ID	PNT2022TMID53401
Project Title	Inventory Management System for Retailers
DATE	28-10-2022
MARK	20 Points

Functional Requirement:

- Registration page
- Login page
- Signup page
- Warning alerts
- Success alert

CODE:

```
<div class="outer">
<div class="container">
  <div class="row">
    <div class="offset-md-2 col-lg-5 col-md-7 offset-lg-4 offset-md-3">
      <div class="panel border bg-white">
        <div class="panel-heading">
          <h2 class="pt-3 font-weight-bold">Login</h2>
        </div>
        <div class="panel-body pt-3">
          <form [formGroup]="loginForm">
            <div class="form-group pt-3 p-3 ">
              <label style="font-weight: bold;">User Name</label>
              <div class="input-field mt-1"
                [class.error]="loginForm.controls['username'].dirty &&
loginForm.hasError('required','username')">
                <span class="fa fa-user pt-2"></span>
                <input formControlName="username" type="text"
placeholder="username">
              </div>
            </div>
            <small
              *ngIf="loginForm.controls['username'].dirty &&
loginForm.hasError('required','username')
              class="text-danger p-3">*username is required</small>
          </div class="form-group pt-3 p-3">
```

```

        <label style="font-weight: bold;">Password</label>
        <div class="input-field mt-1"
            [class.error]="loginForm.controls['password'].dirty &&
loginForm.hasError('required','password')">
            <span class="fa fa-lock pt-2"></span>
            <input formControlName="password" [type]="type"
placeholder="password">
            <span (click)="hideShowPass()" class="fa {{eyelcon}}"></span>
        </div>
    </div>
    <small
        *ngIf="loginForm.controls['password'].dirty &&
loginForm.hasError('required','password')
        class="text-danger p-3">*password is required</small>
    <div class="form-inline fr p-3 pt-1">
        <a id="forget" class="font-weight-bold">forget password ?</a>
    </div>
    <div class="p-3">
        <button (click)="(onSubmit())" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Login</button>
    </div>
    <div class="p-3 pt-0">
        <button routerLink="/signup" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Register</button>
    </div>
</form>
</div>
<div class="mx-3 my-2 py-2 bordert">
    <div class="text-center py-3">
        <a href="https://www.facebook.com" target="_blank" class="px-2"></a>
        <a href="https://www.google.com" target="_blank" class="px-2"></a>
        <a href="https://www.github.com" target="_blank" class="px-2"></a>
    </div>
</div>
</div>
</div>
</div>

```

```

<div class="outer">
  <div class="container">
    <div class="row">
      <div class="offset-md-2 col-lg-5 col-md-7 offset-lg-4 offset-md-3">
        <div class="panel border bg-white">
          <div class="panel-heading">
            <h2 class="pt-3 font-weight-bold">Create New Account</h2>
          </div>
          <div class="panel-body pt-3 p-3 ">
            <form [formGroup]="signupForm">
              <div class="form-group mt-1 p-3 ">
                <label style="font-weight: bold;">User Name</label>
                <div class="input-field">
                  <span class="fa fa-user pt-2"></span>
                  <input
                    [class.error]="signupForm.controls['username'].dirty &&
signupForm.hasError('required','username')"
                    formControlName="username" type="text"
placeholder="username">
                </div>
              </div>
              <div class="form-group mt-1 p-3 pt-0">
                <label style="font-weight: bold;">Email</label>
                <div class="input-field">
                  <span class="fa fa-user pt-2"></span>
                  <input
                    [class.error]="signupForm.controls['email'].dirty &&
signupForm.hasError('required','email')"
                    formControlName="email" type="email" placeholder="Email">
                </div>
              </div>
              <div class="form-group p-3 pt-0">
                <label style="font-weight: bold;">Password</label>
                <div class="input-field">
                  <span class="fa fa-lock pt-2"></span>
                  <input
                    [class.error]="signupForm.controls['password'].dirty &&
signupForm.hasError('required','password')"
                    formControlName="password" [type]="type"
placeholder="password">
                  <span (click)="hideShowPass()" class="fa {{eyelcon}}"></span>
                </div>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <div class="p-3">
            <button (click)="(onSubmit())" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Register</button>
        </div>
        <div class="col-12 text-center">
            <span>Already Have an Account</span>
        </div>
        <div class="p-3 pt-0">
            <button routerLink="/" type="submit"
class="btn btn-primary btn-block mt-4 w-100">Login</button>
        </div>
    </form>
</div>
<!-- <div class="mx-3 my-2 py-2 bordert">
    <div class="text-center py-3">
        <a href="https://www.facebook.com" target="_blank" class="px-2" ></a>
        <a href="https://www.google.com" target="_blank" class="px-2" ></a>
        <a href="https://www.github.com" target="_blank" class="px-2" ></a>
    </div>
</div>--->

</div>
    </div>
</div>
</div>
</div>

```

COMMON CSS

```
.outer { height: 100vh; background: linear-gradient(to top, #c9c9ff 50%,
#6e6ef6 90%) no-repeat;
}
.panel-heading { text-align: center; margin-bottom: 10px;
}
#forget { min-width: 100px; margin-left: auto; text-decoration: none; cursor: pointer;
}
a:hover { text-decoration: none;
}
.form-inline label { padding-left: 10px; margin: 0%; cursor: pointer;
}
.btn { margin-top: 20px; border-radius: 15px;
}
.panel { min-height: 380px; box-shadow: 20px 20px 80px rgb(218, 218, 218); border-radius: 12px; margin-top: 100px;
}
.input-field { border-radius: 5px; padding: 5px; display: flex; align-items: center; cursor: pointer; border: 1px solid #ddd; color: #4343ff;
}
.input-field.error { border: 1px solid red; color: red;
} input[type="text"],
input[type="email"],
input[type="password"] { border: none; outline: none; box-shadow: none; width: 100%;
}
```

```

.fa-eye-slash.btn {
border: none; outline:
none; box-shadow:
none;
}
img { width: 40px;
height: 40px;
object-fit: cover;
border-radius: 50%;
position: relative;
}
.bordert { border-top: 1px
solid #aaa; position:
relative;
}
.bordert::after { content:
"or connect with";
position: absolute; top: -
13px; left: 40%;
background-color: #fff;
padding: 0 8 px;
}
/*--font-family: 'Montserrat', sans-serif;--*/
@import
url("https://fonts.googleapis.com/css2?family=Nunito:wght@200;300;400;600;700;800;9
00&display=swap"); /* @import
url("https://fonts.googleapis.com/css2?family=Montserrat:wght@100;200;300;400;500;6
00;700;800;900&display=swap"); */

/*===== //--//-->
STYLES GENERALES
=====*/

body {
padding: 0; margin:
0; font-family:
"Nunito", sans-
serif; font-size:
14px;
background: #fff;
}
h1, h2, h3, h4,
h5, h6 { color:
#000000;
}

a { color: #0276fd;
transition: all 0.5s ease;
}

```

```

a:hover { color:
#006be7; text-
decoration: none;
}
.form-control::placeholder { /* Chrome,
Firefox, Opera, Safari 10.1+ */ color:
#8dacd1; opacity: 1; /* Firefox */ font-
size: 14px !important;
}
.form-control:-ms-input-placeholder {
/* Internet Explorer 10-11 */ color:
#8dacd1; font-size: 14px !important;
}
.form-control::-ms-input-placeholder {
/* Microsoft Edge */ color: #8dacd1;
font-size: 14px !important;
}
.form-control select {
color: #8dacd1; font-size:
14px !important;
}
.theme-form select { color:
#8dacd1 !important; font-
size: 14px !important;
}
.theme-form select option {
color: #8dacd1 !important; font-
size: 14px !important;
}
.theme-form select option:first-child {
color: #8dacd1 !important; font-size:
14px !important;
}
.btn { padding: 0.375rem
0.95rem; font-size: 14px
!important; min-height:
38px; font-weight: 500;
}
.btn-sm { padding: 0.25rem
0.75rem; font-size: 13px
!important; min-height:
28px; font-weight: 500;
}
.lftBdr-radius { border-
radius: 5px 0 0 5px;
}
.rgtBdr-radius { border-
radius: 0 5px 5px 0;
}
/*--Text Color--*/ .blue-text {
color: #006be7 !important;
}

```

```
}
.green-text { color:
#7cb83b !important;
}
.red-text { color: #eb5463
!important;
}
.purple-text { color:
#8d44ad !important;
}
.orange-text { color:
#fa9151 !important;
}
.aqua-text { color: #45c8e9
!important;
}

/*--BG Color--*/ .blue-bg { background-
color: #006be7 !important;
}
.green-bg { background-color: #7cb83b
!important;
}
.red-bg { background-color: #eb5463
!important;
}
.purple-bg { background-color: #8d44ad
!important;
}
.orange-bg { background-color:
#fa9151 !important;
}
.aqua-bg { background-color: #45c8e9
!important;
}

.bg-info-light { background-color:
#e4f6f9 !important;
}
.bg-orange-light { background-color:
#fff1e8 !important;
}
.bg-red-light { background-color:
#ffe9eb !important;
}
}
/*-----*/
/*--Login Page--*/ .login-
page { background:
#000; padding: 75px 0px;
```



```

}
.login-page .card {
border: 0;
}
.login-page .card-header {
text-align: center;
background: #333; padding:
15px 0;
}
.login-page .form-group { margin-
bottom: 1.5rem;
}
.login-page .form-control {
font-size: 14px; padding:
1.25rem 0.75rem;
}
/*----*/
/*--Admin Page--*/
/*--HEADER--*/ #my-header .bg-dark {
background-color: #0276fd !important;
}
#my-header .navbar-brand {
margin-right: 15px; width:
235px;
}
#my-header .navbar {
padding: 0px;
}
#my-header .nav-item {
padding: 15px 12px;
}
#my-header .nav-item .nav-link {
color: #eee !important; padding:
10px 0px;
}
#my-header .nav-item .nav-link:hover {
color: #0276fd !important;
}
.nav-user { padding: calc(31px / 2) 20px calc(31px / 2) 57px
!important; text-align: left !important; position: relative;
background-color: rgb(18, 119, 202); border-width: 0 1px;
min-width: 180px;
}
.nav-user .account-user-avatar {
position: absolute; top:
calc(34px / 2); left: 15px;
}
.nav-user .account-user-avatar img {
height: 36px; width: 36px;
}

```

```

.nav-user .account-user-name {
display: block; font-weight:
500; color: #fff; font-size:
14px !important;
}
.nav-user .account-position {
display: block; font-size:
11px; margin-top: 0px;
color: #fff;
}
.dropdown-menu-right { min-
width: 180px;
}
.dropdown-menu-right .dropdown-item {
font-size: 14px; padding: 0.35rem
0.75rem; font-weight: 500; color:
#394449;
}
.dropdown-toggle::after {
display: none;
}
.dropdown-menu-animated {
-webkit-animation-name: DropDownSlide;
animation-name: DropDownSlide; -webkit-
animation-duration: 0.3s; animation-
duration: 0.3s; -webkit-animation-fill-
mode: both; animation-fill-mode: both;
margin: 0; position: absolute; z-index:
1000;
}
.dropdown-menu-animated.show {
top: 100% !important;
}
.dropdown-menu-animated i {
display: inline-block; margin-right:
3px;
}
.dropdown-menu-animated.dropdown-menu-right {
right: 0 !important; left: auto !important;
}
.dropdown-menu-animated.dropdown-menu[x-placement^="left"],
.dropdown-menu-animated.dropdown-menu[x-placement^="right"],
.dropdown-menu-animated.dropdown-menu[x-placement^="top"] { top:
auto !important;
-webkit-animation: none !important;
animation: none !important;
}
@-webkit-keyframes DropDownSlide {
100% {

```

```

    -webkit-transform: translateY(0);
transform: translateY(0);
}
0% {
    -webkit-transform: translateY(20px);
transform: translateY(20px);
}
}
@keyframes DropDownSlide {
    100% {
        -webkit-transform: translateY(0);
transform: translateY(0);
    }
    0% {
        -webkit-transform: translateY(20px);
transform: translateY(20px);
    }
}
/*--MAIN-MENU--*/ .main-menu {
background: #111; padding:
10px; border-bottom: 1px solid
#222;
}
.main-menu .nav-item {
margin: 8px;
}
.main-menu .nav-link {
border: 1px solid #007bff;
color: #007bff; text-
transform: uppercase; font-
size: 13px; background:
#ffffff; font-weight: 600;
}
.main-menu .nav-link:hover {
color: #fff;
background-color: #007bff;
}
.main-menu .nav-link i {
font-size: 1.15rem;
display: block; margin:
0 auto 5px auto; text-
align: center;
}
.green-link { border: 1px solid #7cb83b
!important; color: #7cb83b !important;
}
.green-link.active { color:
#fff !important;
background-color:
#7cb83b !important;

```

```
}  
.green-link:hover { color: #fff  
!important; background-color: #7cb83b  
!important;  
}  
.red-link { border: 1px solid #eb5463  
!important; color: #eb5463 !important;  
}  
.red-link.active { color: #fff !important;  
background-color: #eb5463 !important;  
}  
.red-link:hover { color: #fff !important;  
background-color: #eb5463 !important;  
}  
.purple-link { border: 1px solid  
#8d44ad !important; color: #8d44ad  
!important;  
}  
.purple-link.active { color: #fff  
!important; background-color: #8d44ad  
!important;  
}  
.purple-link:hover { color: #fff  
!important; background-color: #8d44ad  
!important;  
}  
.mint-link { border: 1px solid #46ceac  
!important; color: #46ceac !important;  
}  
.mint-link.active { color: #fff !important;  
background-color: #46ceac !important;  
}  
.mint-link:hover { color: #fff !important;  
background-color: #46ceac !important;  
}  
.orange-link { border: 1px solid #fa9151  
!important; color: #fa9151 !important;  
}  
.orange-link.active { color:  
#fff !important;  
background-color:  
#fa9151 !important;  
}  
.orange-link:hover { color: #fff  
!important; background-color: #fa9151  
!important;  
}  
.aqua-link { border: 1px solid #45c8e9  
!important; color: #45c8e9 !important;  
}
```

```

.aqua-link.active { color: #fff !important;
background-color: #45c8e9 !important;
}
.aqua-link:hover { color: #fff !important;
background-color: #45c8e9 !important;
}
.yellow-link { border: 1px solid #c29c18
!important; color: #c29c18 !important;
}
.yellow-link.active { color: #fff
!important; background-color: #c29c18
!important;
}
.yellow-link:hover { color: #fff
!important; background-color: #c29c18
!important;
}
.pink-link { border: 1px solid #f5799c
!important; color: #f5799c !important;
}
.pink-link.active { color: #fff !important;
background-color: #f5799c !important;
}
.pink-link:hover { color: #fff !important;
background-color: #f5799c !important;
}
/*--MENU-PAGE--*/
/*--Sub Menu--*/ .sub-menu
.nav-item { margin: 0px
1px;
}
.sub-menu .nav-item .nav-link { background: #fff; border-radius: 5px 5px 0px 0px; font-
weight: 500; padding: 0.65rem 1rem;
}
.sub-menu .nav-item .nav-link:hover {
background: #0276fd;
color: #fff;
}
.sub-menu .nav-item .active {
background: #0276fd;
color: #fff;
}
}
.page-header { background:
#0276fd; margin: 0 1px;
padding: 0; border-radius: 0px
5px 0px 0px;
}
.ph-radius { border-radius: 5px
5px 0px 0px;
}

```

```

.page-header h3 {
margin: 0; color:
#fff; font-size:
1rem; padding:
13px;
}
.page-header .breadcrumb {
margin-bottom: 0; list-style:
none; background-color:
transparent; border-radius: 0;
font-size: 12px; padding:
0.9rem;
}
.page-header .breadcrumb .breadcrumb-item a {
color: #fff;
}
.page-header .breadcrumb .breadcrumb-item.active { color:
#fff;
}
.page-header .breadcrumb-item + .breadcrumb-item::before {
display: inline-block; padding-right: 0.5rem; color: #fff;
content: "|";
}
.page-action {
display: -ms-flexbox;
display: flex;
-ms-flex-wrap: wrap; flex-
wrap: wrap; margin-
bottom: 0; list-style: none;
padding-left: 0;
}
.page-action li { border-right:
1px solid #2d92ff; border-left:
1px solid #0166d3; padding:
0.7rem 0.9rem;
}
.page-action li:last-child {
border-right: 0; border-radius:
0px 5px 0px 0px;
}
.page-action li:hover {
background: #0073ef;
}
.page-action li a i {
vertical-align: middle;
color: #fff; font-size:
20px;
}
/*--WRAPPER--*/ .content-
wrapper { padding: 1.5rem 0

```

```

1.5rem 0; background: #b2bfdb;
/* height: calc(100vh - 230px); */
overflow: hidden; margin: 0
1px;
}
.table-striped tbody tr:nth-of-type(odd) { background-color:
#fcfdff;
}
.table-striped thead {
background: #dee5f1;
}
.table-striped > tbody td {
border: 1px solid #dee5f1;
}
.table-striped thead th { border-
bottom: 2px solid #dee5f1;
}
.table-container { margin-bottom:
1rem; margin-top: 1rem;
padding: 0;
/* border: 2px solid #dee5f1; */
-webkit-border-radius: 3px;
-moz-border-radius: 3px; border-
radius: 0;
}
.custom-table { border:
1px solid #dee5f1;
background: #fff;
}
.custom-table thead {
background: #dee5f1;
}
.custom-table thead th {
border: 0; white-
space: nowrap; font-
size: 13px; color:
#000;
}
/* .custom-table > tbody tr:hover { background: #eef1f8; } */
.custom-table > tbody tr:nth-of-type(even) { background-
color: #fcfdff;
}
.custom-table > tbody td {
border: 1px solid #dee5f1; white-
space: nowrap;
}
.table th { padding: 0.8rem
1rem; font-weight: 600;
text-transform: uppercase;

```

```

border: 1px solid #000;
background: #fff;
}
.table td { border-top: 1px
solid #dee5f1; vertical-align:
middle; padding: 0.65rem
1rem;
}
.td-notes { max-width:
185px; overflow:
hidden; text-overflow:
ellipsis; white-space:
nowrap;
}
#sort-table_length { padding-left:
15px; padding-top: 15px;
padding-bottom: 15px; display:
inline-block; width: 50%;
}

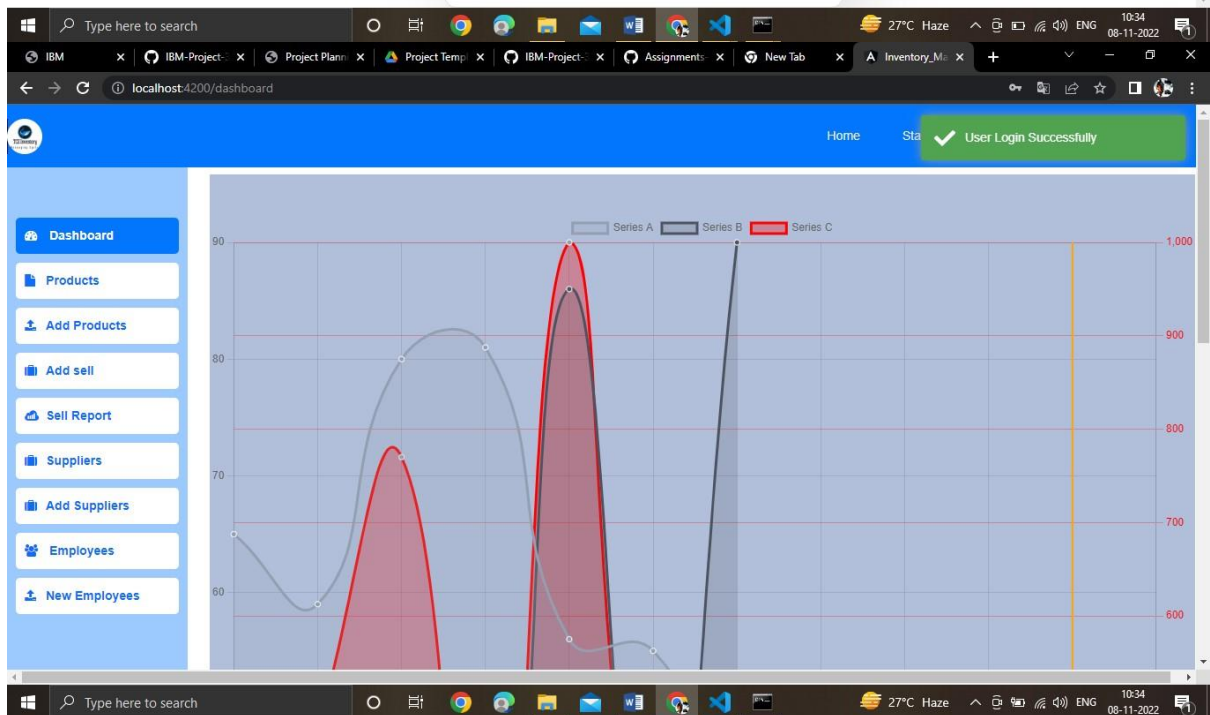
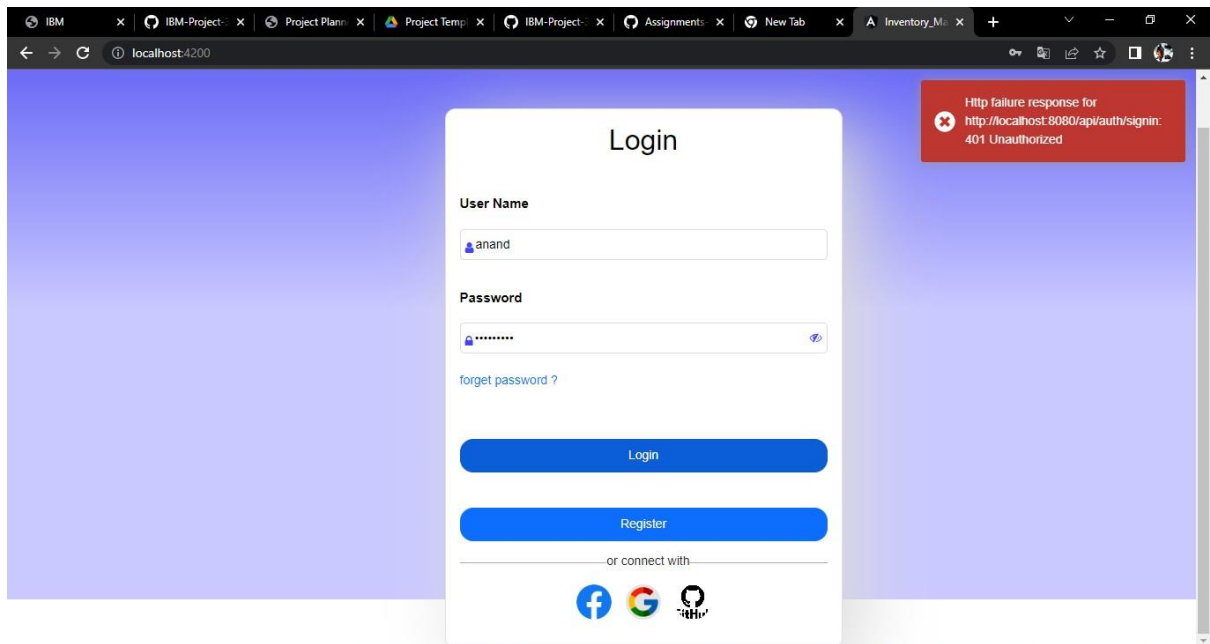
#sort-table_paginate {
padding-right: 15px;
padding-top: 15px;
padding-bottom: 15px;
display: inline-block; width:
50%;
}
.dataTables_info {
padding-left: 15px;
font-weight: 500;
}
.dataTables_paginate { padding-
right: 15px;
}
.theme-form label { margin-bottom:
0.5rem !important; font-size: 15px
!important; font-weight: bolder
!important; color: #000;
}
.theme-form .form-control {
background: #fcfdff; padding:
0.375rem 0.75rem; height:
calc(1.95em + 0.75rem + 2px); border-
color: #dbe4f1; font-size: 14px
!important;
}
.theme-form .form-group { margin-
bottom: 1.75rem;
}

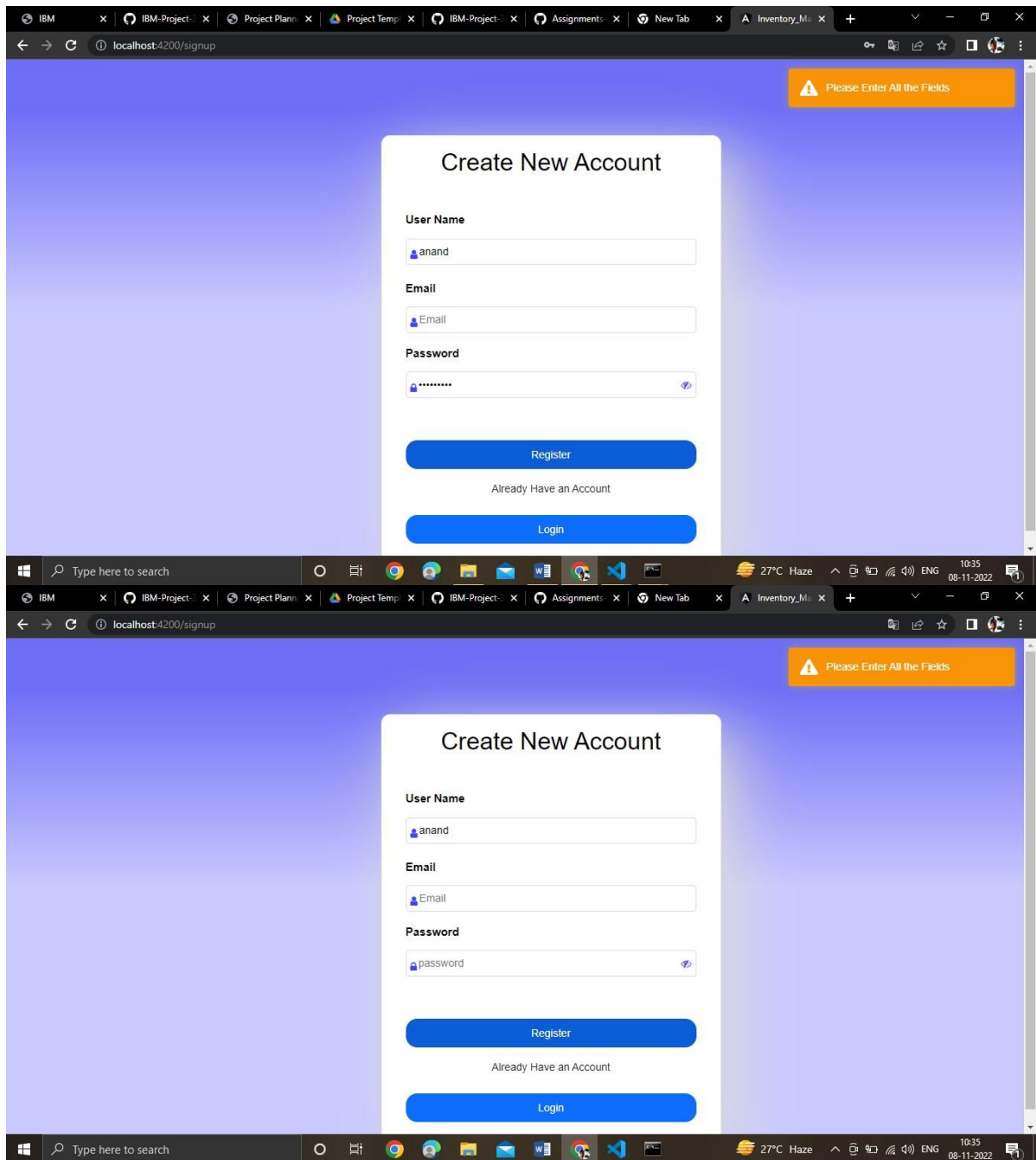
```



```
/*--Theme Modal--*/ .modal-  
header { background:  
#0276fd;  
}  
.modal-title {  
color: #fff;  
}  
.modal-content {  
border: 0;  
}  
.close { float:  
right; font-  
size:  
1.85rem;  
font-weight:  
500; line-  
height: 1;  
color: #fff;  
text-shadow: none;  
opacity: 0.75;  
}
```

Outputs:





MongoDB (not SQL): Controller

```
const config =
require("../config/auth.config"); const db =
require("../models"); const User = db.user;
const Role = db.role;
var jwt =
require("jsonwebtoken"); var
bcrypt = require("bcryptjs");
```



```

    exports.signup = (req, res) => {    const user =
new User({    username: req.body.username,
email: req.body.email,    password:
bcrypt.hashSync(req.body.password, 8),
    });    user.save((err, user) => {    if
(err) {    res.status(500).send({
message: err });    return;
    }
    if (req.body.roles) {
        Role.find(
            {
name: { $in: req.body.roles },
            },
            (err, roles) => {    if (err) {
res.status(500).send({ message: err });
return;
            }
            user.roles = roles.map((role) =>
role._id);    user.save((err) => {
if (err) {    res.status(500).send({
message: err });    return;
            }

            res.send({ message: "User was registered successfully!" });
        });
    }
    });
    } else {
        Role.findOne({ name: "user" }, (err, role) => {
if (err) {
            res.status(500).send({ message: err });
return;
        }

        user.roles = [role._id];
user.save((err) => {    if (err) {
res.status(500).send({ message: err });
return;

```



```

    }
    res.send({ message: "User was registered successfully!"
});
    });
    });
    }
    });
};
exports.signin = (req, res) =>
{
    User.findOne({
username: req.body.username,
    })
    .populate("roles", "-__v") .exec((err,
user) => {
        if (err) {
res.status(500).send({ message: err });
return;
        }
        if (!user) {
            return res.status(404).send({
message: "User Not found." });
        }
        var passwordIsValid =
bcrypt.compareSync(
            req.body.password,
user.password
        );
        if (!passwordIsValid) {
            return res.status(401).send({
message: "Invalid Password!" });
        }
        var token = jwt.sign({ id: user.id }, config.secret, {
expiresIn: 86400, // 24 hours
        });
        var authorities =
[];

        for (let i = 0; i < user.roles.length; i++) {
            authorities.push("ROLE_" +
user.roles[i].name.toUpperCase());
        }
        req.session.token = token;
        res.status(200).send({
id: user._id,
            username:
user.username,

```

```
        email: user.email,
roles: authorities,
    });
});
}; exports.signout = async (req, res) => {    try {        req.session =
null;        return res.status(200).send({ message: "You've been signed
out!" });
    } catch (err) {
this.next(err);
    }
};
```