

## Assignment2

|               |   |
|---------------|---|
| NAME          | HARSAVARTHNI.S.K                          |
| TEAM ID       | PNT2022TMID43190                          |
| ROLL NO       | 714219106002                              |
| Project Title | Inventory Management System for Retailers |

### TASK 4 : Signup & Login With Data base

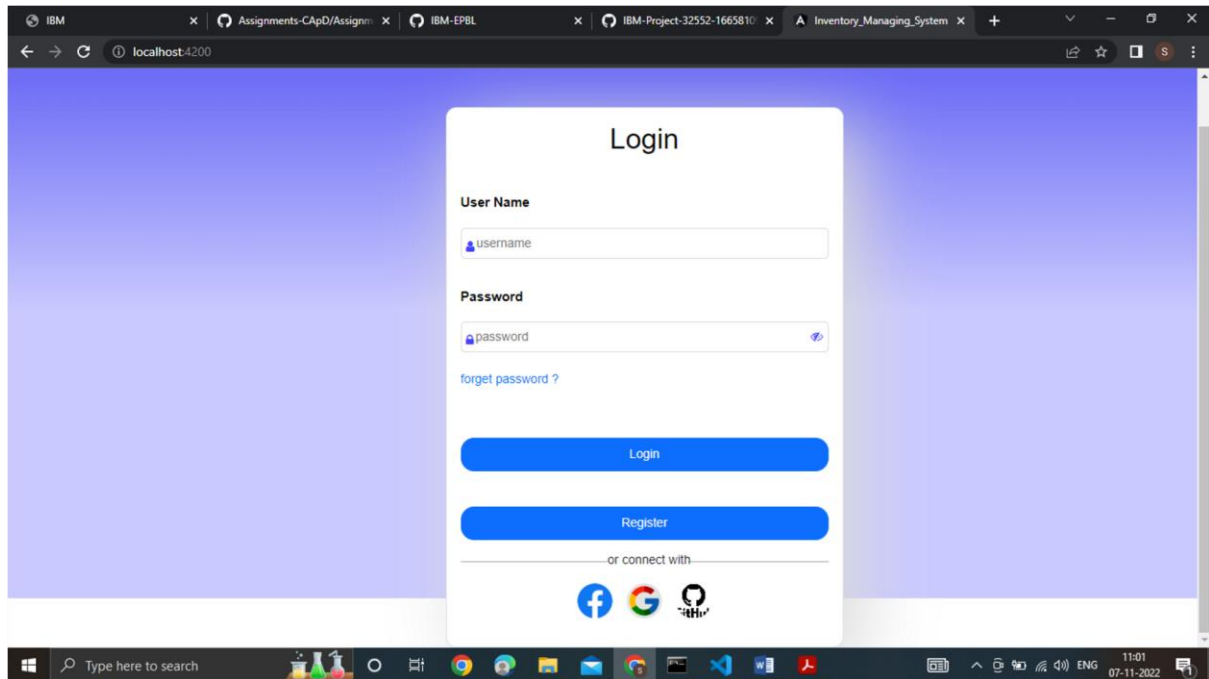
#### Login page

```
<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 {{eyeIcon}}"></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>

```



## Signup Page

```

<app-sidenav></app-sidenav>
<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>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

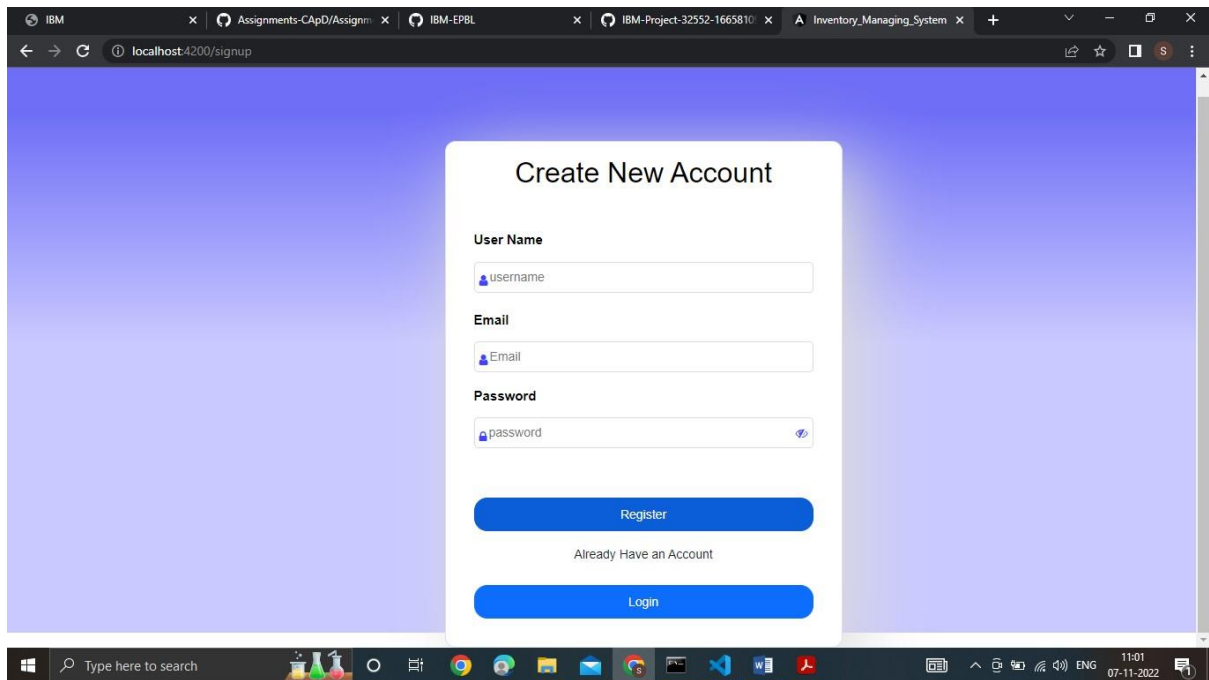
```

    </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 {{eyeIcon}}"></span>
          </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>

```



## 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");

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;
}

```

## Backend(not SQL)

```

const db = require("../models");
const ROLES = db.ROLES; const
User = db.user;

checkDuplicateUsernameOrEmail = (req, res, next) => {
  // Username  User.findOne({
username: req.body.username
}).exec((err, user) => {
  if (err) {
    res.status(500).send({ message: err });    return;

```

```

    }

    if (user) {
      res.status(400).send({ message: "Failed! Username is already in use!" });
      return;
    }

    // Email
    User.findOne({ email:
req.body.email
    }).exec((err, user) => {
      if (err) {
        res.status(500).send({ message: err });      return;
      }

      if (user) {
        res.status(400).send({ message: "Failed! Email is already in use!" });      return;
      }

      next();
    });
  });
};

checkRolesExisted = (req, res, next) => { if
(req.body.roles) {
  for (let i = 0; i < req.body.roles.length; i++) { if
(!ROLES.includes(req.body.roles[i])) {
    res.status(400).send({
      message: `Failed! Role ${req.body.roles[i]} does not exist!`
    });
    return;
  }
}
}

  next();
};

const verifySignUp = { checkDuplicateUsernameOrEmail,
checkRolesExisted
};

module.exports = verifySignUp;

```

