

PROJECT DEVELOPMENT PHASE

SPRINT 1

Date	18 Nov 2022
Team ID	PNT2022TMID44584
Project Name	Project – Inventory Management System For Retailers

FUNCTIONAL REQUIREMENTS

Login page

Registration page

CODE

Login.html

```
<html>
  <head>
    <title>login</title>
    <link rel="stylesheet" href="color.css">
  </head>
  <style>
    .border{
      height: 30rem;
      width: 25rem;
      border-radius: .5rem;
      margin: 2rem;
      cursor: pointer;
      overflow: hidden;
      position: relative;
      box-shadow: 0 0 .5rem #333;
      left: 300px;
      top: 45px;
    }
  </style>
</html>
```



```
</form>
</div></center>
</body>
</html>
```

Register.html

```
<html>
  <head>
    <link rel="stylesheet" href="co.css">
  </head><body>
    <div class="container">
<form id="contact">
    <h1>Registration Form</h1>
    <h3>Fill the form below and press the submit button!</h3>
    <div class="row">
      <!-- first column -->
      <div class="column">
        <fieldset>
          <input type="text" placeholder="Full Name *"
name="name" >
        </fieldset>
        <fieldset>
          <input type="email" placeholder="Email Id *"
name="email">
        </fieldset>
        <fieldset>
          <input type="password" placeholder="password *"
name="password" >
        </fieldset>
        <!-- adding all country code list -->
        <fieldset>
          <input type="text" placeholder="Phone number *"
name="phone" id="phone">
        </fieldset>
      </div>
      <!-- second column -->
      <div class="column">
        <fieldset>
          <input type="text" placeholder="Address *"
name="address" id="address" >
        </fieldset>
      </div>
    </div>
  </form>
</div>
</body>
</html>
```

```

        <input type="text" placeholder="Land Mark *"
name="Land Mark" id="land mark" >
    </fieldset>
    <fieldset>
        <input type="text" placeholder="District *"
name="district" id="address" ></fieldset>
        <fieldset>
            <input type="text" placeholder="Pin code *"
name="pincode" id="pincode" >
        </fieldset>
    </div>
</div>
<!-- submit button -->
<fieldset>
    <button type="submit" id="button"><a href="lo.html" >submit
now</a></button>
</fieldset></div>
</form>
</body>
</html>

```

Common.css

```

* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}
body {
    font-family: 'Open Sans', sans-serif, helvetica, Arial;
    font-weight: 400;
    font-size: 14px;
    color: black;
    /* body background image */
    background-image: linear-gradient(to bottom, rgba(128, 128, 128, 0.541),
    rgba(171, 37, 238, 0.637)), url(image.jpg);
    background-attachment: fixed;
    background-size: cover;
}
.container {
    max-width: 800px;
    width: 100%;
    margin: 0 auto;
}
#contact {

```

```
background-color: rgba(25, 23, 25, 0.7);
padding: 20px;
margin: 50px 0;
}
#contact input, button {
  font: 400 15px 'Open Sans', sans-serif, helvetica, Arial;
}
#contact h1 {
  font-size: 35px;
  font-weight: bold;
  text-align: center;
  color: rgb(246, 246, 250);
}
#contact h3 {
  margin: 5px 0px 15px;
  text-align: center;
  color: rgb(150, 150, 150)
}
.row {
  display: flex;
  width: 100% !important;
  padding: 20px 0px;
}
.row .column {
  margin: 0px 20px;
  width: 50%;
}
fieldset {
  border: medium none !important;
  margin: 0 0 10px;
  min-width: 100%;
  width: 100%;
}
#contact input {
  width: 100%;
  border: 1px solid rgb(150, 150, 150);
  background-color: white;
  padding: 10px;
  margin: 5px 0;
}
input[type = "radio"] {
  width: 10% !important;
}
#contact .row .radio {
  border: 1px solid rgb(150, 150, 150);
  background-color: white;
  margin: 7px 0 10px;
  padding: 5px;
```

```

}

#contact input:hover {
  transition: border-color 0.3s ease-in-out;
  border: 1px solid rgb(68, 68, 68);
}
#contact button {
  outline: none;
  border: none;
  background-color: rgb(176, 176, 185);
  color: rgb(25, 24, 24);
  margin: 0 0 5px 40%;
  padding: 10px;
  font-size: 17px;
  width: 150px;
}

#contact button:hover {
  background-color: rgba(0, 0, 255, 0.8);
}

```

Db

```

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);
}
};
};

```

OUTPUT



