SPRINT 3

PROJECT NAME	SMART FASHION RECOMMENDER APPLICATION
TEAM ID	PNT2022TMID08162

Search.html

```
<head>
  <link rel=”stylesheet” href=”css/home.css”>
  <link rel=”stylesheet” href=”css/search.css”>
</head>
<body>
   <nav class=”navbar”></nav>
   <footer></footer>

   <script src=”js/nav.js”></script>
   <script src=”js/footer.js”></script>
</body>
```

```
<section class=”search-results”>
   <h2 class=”heading”>search results for <span>product</span></h2>
</section>
<img src=”img/404.png” class=”four-0-four-image” alt=””>
<p class=”four-0-four-msg”>look like you are lost. Head to beack to our <a
href=”#”>homepage</a></p>
```

Search.css

```
.search-results{
```

```
    Width: 100%;
    Padding: 60px 0;
}

.heading{
    Font-size: 20px;
    Text-transform: capitalize;
    Font-weight: 400;
    Margin-bottom: 40px;
    Padding: 0 10vw;
}

.heading span{
    Font-weight: 700;
}
```

Product.css

```
.product-container{
    Display: grid;
    Grid-template-columns: repeat(4, 1fr);
    Height: auto;
    Grid-row-gap: 40px;
}
```

Footer.css

```
.four-0-four-image{
    Display: block;
```

```
    Margin: 60px auto;
}

.four-0-four-msg{
    Text-align: center;
    Text-transform: capitalize;
    Color: #383838;
}

.four-0-four-msg a{
    Color: #383838;
}
```

Server page

Server.js

```
“scripts”: {
    “start”: “nodemon server.js”
}
Const express = require(‘express’);
Const admin = require(‘firebase-admin’);
Const bcrypt = require(‘bcrypt’);
Const path = require(‘path’);
// declare static path
Let staticPath = path.join(__dirname, “public”);
//intializing express.js
Const app = express();

//middlewares
App.use(express.static(staticPath));
```

```
App.listen(3000, () => {
   Console.log('listening on port 3000…….');
})
//routes
//home route
App.get("/", (req, res) => {
   Res.sendFile(path.join(staticPath, "index.html"));
})

App.get('/404', (req, res) => {
   Res.sendFile(path.join(staticPath, "404.html"));
})

App.use((req, res) => {
   Res.redirect('/404');
})
//signup route
App.get('/signup', (req, res) => {
   Res.sendFile(path.join(staticPath, "signup.html"));
})
// firebase admin setup
Let serviceAccount = require("path of key file");

Admin.initializeApp({
  Credential: admin.credential.cert(serviceAccount)
});

Let db = admin.firestore();
// store user in db
Db.collection('users').doc(email).get()
```

```
.then(user => {
   If(user.exists){
      Return res.json({‘alert’: ‘email already exists’});
   } else{
      // encrypt the password before storing it.
      Bcrypt.genSalt(10, (err, salt) => {
         Bcrypt.hash(password, salt, (err, hash) => {
            Req.body.password = hash;
            Db.collection(‘users’).doc(email).set(req.body)
            .then(data => {
               Res.json({
                  Name: req.body.name,
                  Email: req.body.email,
                  Seller: req.body.seller,
               })
            })
         })
      })
   }
})
Db.collection(‘users’).doc(email).get()
.then(…)
Const processData = (data) => {
   Loader.style.display = null;
   If(data.alert){
      showAlert(data.alert);
   } else if(data.name){
      // create authToken
      Data.authToken = generateToken(data.email);
      sessionStorage.user = JSON.stringify(data);
      location.replace(‘/’);
```

```
    }
}
```

Signup page

Form.html

```
<img src="img/loader.gif" class="loader" alt="">
<div class="container">
    <img src="img/dark-logo.png" class="logo" alt="">
    <div>
        <input type="text" autocomplete="off" id="name" placeholder="name">
        <input type="email" autocomplete="off" id="email" placeholder="email">
        <input type="password" autocomplete="off" id="password" placeholder="password">
        <input type="text" autocomplete="off" id="number" placeholder="number">
        <input type="checkbox" checked class="checkbox" id="terms-and-cond">
        <label for="terms-and-cond">agree to our <a href="">terms and conditions</a></label>
        <br>
        <input type="checkbox" class="checkbox" id="notification">
        <label for="notification">receive upcoming offers and events mails</a></label>
        <button class="submit-btn">create account</button>
    </div>
    <a href="/login" class="link">already have an account? Log in here</a>
</div>
<div class="alert-box">
    <img src="img/error.png" class="alert-img" alt="">
    <p class="alert-msg">Error message</p>
</div>
<script src="js/form.js"></script>
```

Form.css

```
*{
    Margin: 0;
    Padding: 0;
    Box-sizing: border-box;
}

Body{
    Width: 100%;
    Min-height: 100vh;
    Display: flex;
    Justify-content: center;
    Align-items: center;
    Background: #f5f5f5;
    Font-family: ‘roboto’, sans-serif;
}

.loader{
    Position: absolute;
    Top: 50%;
    Left: 50%;
    Transform: translate(-50%, -50%);
    Width: 100px;
}
.logo{
    Height: 80px;
    Display: block;
    Margin: 0 auto 50px;
}

Input[type=”text”],
```

```
Input[type=”password”],
Input[type=”email”],
Textarea{
    Display: block;
    Width: 300px;
    Height: 40px;
    Padding: 20px;
    Border-radius: 5px;
    Background: #fff;
    Border: none;
    Outline: none;
    Margin: 20px 0;
    Text-transform: capitalize;
    Color: #383838;
    Font-size: 14px;
    Box-shadow: 0 4px 10px rgba(0, 0, 0, 0.01);
    Font-family: ‘roboto’, sans-serif;
}

::placeholder{
    Color: #383838;
}

.submit-btn{
    Width: 300px;
    Height: 40px;
    Text-align: center;
    Line-height: 40px;
    Background: #383838;
    Color: #fff;
    Border-radius: 2px;
```

```
    Text-transform: capitalize;
    Border: none;
    Cursor: pointer;
    Display: block;
    Margin: 30px 0;
}

/* checkbox styles */

.checkbox{
    -webkit-appearance: none;
    Position: relative;
    Width: 15px;
    Height: 15px;
    Border-radius: 2px;
    Background: #fff;
    Border: 1px solid #383838;
    Cursor: pointer;
}

.checkbox:checked{
    Background: #383838;
}

.checkbox::after{
    Content: ‘’;
    Position: absolute;
    Top: 60%;
    Left: 50%;
    Transform: translate(-50%, -50%);
    Width: 80%;
```

```
    Height: 100%;
    Pointer-events: none;
    Background-image: url(../img/check.png);
    Background-size: contain;
    Background-repeat: no-repeat;
    Display: none;
}

.checkbox:checked::after{
    Display: block;
}

Label{
    Text-transform: capitalize;
    Display: inline-block;
    Margin-bottom: 10px;
    Font-size: 14px;
    Color: #383838;
}

Label a{
    Color: #383838;
}

.link{
    Color: #383838;
    Text-transform: capitalize;
    Text-align: center;
    Display: block;
}
/* alert */
```

```
.alert-box{
    Width: 300px;
    Min-height: 150px;
    Background: #fff;
    Border-radius: 10px;
    Box-shadow: 0 5px 100px rgba(0, 0, 0, 0.05);
    Position: absolute;
    Top: 60%;
    Left: 50%;
    Transform: translate(-50%, -50%);
    Padding: 20px;
    Opacity: 0;
    Pointer-events: none;
    Transition: 1s;
}

.alert-box.show{
    Opacity: 1;
    Pointer-events: all;
    Top: 50%;
}

.alert-img{
    Display: block;
    Margin: 10px auto 20px;
    Height: 60px;
}

.alert-msg{
    Color: #e24c4b;
    Font-size: 20px;
```

```
    Text-transform: capitalize;
    Text-align: center;
    Line-height: 30px;
    Font-weight: 500;
}
```

Form.js

```
Const loader = document.querySelector(‘.loader’);

// select inputs
Const submitBtn = document.querySelector(‘.submit-btn’);
Const name = document.querySelector(‘#name’);
Const email = document.querySelector(‘#email’);
Const password = document.querySelector(‘#password’);
Const number = document.querySelector(‘#number’);
Const tac = document.querySelector(‘#terms-and-cond’);
Const notification = document.querySelector(‘#notification’);
submitBtn.addEventListener(‘click’, () => {
      if(name.value.length < 3){
         showAlert(‘name must be 3 letters long’);
      } else if(!email.value.length){
         showAlert(‘enter your email’);
      } else if(password.value.length < 8){
         showAlert(‘password should be 8 letters long’);
      } else if(!number.value.length){
         showAlert(‘enter your phone number’);
      } else if(!Number(number.value) || number.value.length < 10){
         showAlert(‘invalid number, please enter valid one’);
      } else if(!tac.checked){
```

```
            showAlert('you must agree to our terms and conditions');
        } else{
            // submit form
        }
})
If(name.value.length < 3){
    showAlert('name must be 3 letters long');
}
showAlert('name must be 3 letters long');
// alert function
Const showAlert = (msg) => {
    Let alertBox = document.querySelector('.alert-box');
    Let alertMsg = document.querySelector('.alert-msg');
    alertMsg.innerHTML = msg;
    alertBox.classList.add('show');
    setTimeout(() => {
        alertBox.classList.remove('show');
    }, 3000);
}
// send data function
Const sendData = (path, data) => {
    Fetch(path, {
        Method: 'post',
        Headers: new Headers({'Content-Type': 'application/json'}),
        Body: JSON.stringify(data)
    }).then((res) => res.json())
    .then(response => {
        processData(response);
    })
}
Else{
```

```
    // submit form
    Loader.style.display = ‘block’;
    sendData(‘/signup’, {
        name: name.value,
        email: email.value,
        password: password.value,
        number: number.value,
        tac: tac.checked,
        notification: notification.checked,
        seller: false
    })
}
Const compareToken = (token, key) => {
    Let string = ‘’;
    For(let I = 0; I < token.length; i=i+2){
        Let index1 = char.indexOf(token[i]);
        Let index2 = char.indexOf(token[i+1]);
        String += char[index1 + index2];
    }
    If(string === key){
        Return true;
    }
    Return false;
}
```

Signup-post

```
App.use(express.json());
App.post(‘/signup’, (req, res) => {
```

```
    Let { name, email, password, number, tac, notification } = req.body;


    // form validations
    If(name.length < 3){
        Return res.json({‘alert’: ‘name must be 3 letters long’});
    } else if(!email.length){
        Return res.json({‘alert’: ‘enter your email’});
    } else if(password.length < 8){
        Return res.json({‘alert’: ‘password should be 8 letters long’});
    } else if(!number.length){
        Return res.json({‘alert’: ‘enter your phone number’});
    } else if(!Number(number) || number.length < 10){
        Return res.json({‘alert’: ‘invalid number, please enter valid one’});
    } else if(!tac){
        Return res.json({‘alert’: ‘you must agree to our terms and conditions’});
    }
})
Let { name, email, password, number, tac, notification } = req.body;
If(name.length < 3){
    Return res.json({‘alert’: ‘name must be 3 letters long’});
} else if …..
JSON = {
   ‘key’: ‘value’
}
Const processData = (data) => {
    Loader.style.display = null;
    If(data.alert){
        showAlert(data.alert);
    }
}
```

Token.js

```
<script src=”js/token.js”></script>
Const compareToken = (token, key) => {
   Let string = ‘’;
   For(let I = 0; I < token.length; i=i+2){
      Let index1 = char.indexOf(token[i]);
      Let index2 = char.indexOf(token[i+1]);
      String += char[index1 + index2];
   }
   If(string === key){
      Return true;
   }
   Return false;
}
```

Chatbot

```
<script>
                    Window.watsonAssistantChatOptions = {
                      integrationID: “c4969e38-901a-4c1b-a659-d054daf___”, // The ID of this integration.
                      Region: “eu-gb”, // The region your integration is hosted in.
                      serviceInstanceID: “9f76390e-b4b0-4b50-85d1-d22c61d___”, // The ID of your service instance.
                      onLoad: function(instance) {
                            instance.render();  // Renders the web chat widget on your page.
                            }
              };
                    setTimeout(function(){
                      const t=document.createElement(‘script’);
```

```
t.src=https://web-chat.global.assistant.watson.appdomain.cloud/loadWatsonAssistantChat.js;
document.head.appendChild(t);
});
</script>
```