SPRINT 4

PROJECT NAME	SMART FASHION RECOMMENDER APPLICATION
TEAM ID	PNT2022TMID08162

Server.js

```
// add product

App.get('/add-product', (req, res) => {

   Res.sendFile(path.join(staticPath, "addProduct.html"));

})
```

Add Product Page – Design

```
<head>

   // other head tags

   <link rel="stylesheet" href="css/signup.css">

   <link rel="stylesheet" href="css/addProduct.css">

</head>

<body>

   <img src="img/loader.gif" class="loader" alt="">


   <div class="alert-box">

      <img src="img/error.png" class="alert-img" alt="">

      <p class="alert-msg"></p>

   </div>


   <script src="js/token.js"></script>

   <script src="js/addProduct.js"></script>

</body>

<img src="img/dark-logo.png" class="logo" alt="">
```

```
<div class=”form”>
    <input type=”text” id=”product-name” placeholder=”product name”>
    <input type=”text” id=”short-des” placeholder=”short line about the product”>
    <textarea id=”des” placeholder=”detail description”></textarea>

    <!—product image →
    <div class=”product-info”>
        <div class=”product-image”><p class=”text”>product image</p></div>
        <div class=”upload-image-sec”>
            <!—upload inputs →
            <p class=”text”><img src=”img/camera.png” alt=””>upload image</p>
            <div class=”upload-catalouge”>
                <input type=”file” class=”fileupload” id=”first-file-upload-btn” hidden>
                <label for=”first-file-upload-btn” class=”upload-image”></label>
                <input type=”file” class=”fileupload” id=”second-file-upload-btn” hidden>
                <label for=”second-file-upload-btn” class=”upload-image”></label>
                <input type=”file” class=”fileupload” id=”third-file-upload-btn” hidden>
                <label for=”third-file-upload-btn” class=”upload-image”></label>
                <input type=”file” class=”fileupload” id=”fourth-file-upload-btn” hidden>
                <label for=”fourth-file-upload-btn” class=”upload-image”></label>
            </div>
        </div>
        <div class=”select-sizes”>
            <p class=”text”>size available</p>
            <div class=”sizes”>
                <input type=”checkbox” class=”size-checkbox” id=”xs” value=”xs”>
                <input type=”checkbox” class=”size-checkbox” id=”s” value=”s”>
                <input type=”checkbox” class=”size-checkbox” id=”m” value=”m”>
                <input type=”checkbox” class=”size-checkbox” id=”l” value=”l”>
                <input type=”checkbox” class=”size-checkbox” id=”xl” value=”xl”>
                <input type=”checkbox” class=”size-checkbox” id=”xxl” value=”xxl”>
            <input type=”checkbox” class=”size-checkbox” id=”xxxl” value=”xxxl”>
          </div>
        </div>
    </div>

    <div class=”product-price”>
        <input type=”number” id=”actual-price” placeholder=”actual price”>
        <input type=”number” id=”discount” placeholder=”discount percentage”>
        <input type=”number” id=”sell-price” placeholder=”selling price”>
    </div>

    <input type=”number” id=”stock” min=”20” placeholder=”item in sstocks (minimum 20)”>

    <textarea id=”tags” placeholder=”Enter categories here, for example – Men, Jeans, Blue Jeans,
Rough jeans (you sholud add men or women at start)”></textarea>

    <input type=”checkbox” class=”checkbox” id=”tac” checked>
    <label for=”tac”>clothing take 30% from your total sell</label>

    <div class=”buttons”>
        <button class=”btn” id=”add-btn”>add product</button>
        <button class=”btn” id=”save-btn”>save draft</button>
    </div>
</div>
```

Addproduct.css

```
Body{
    Display: block;
    Padding: 0 10vw;
}

.logo{
   Margin: 20px auto 50px;
}

Input, textarea{
   Font-weight: 500;
}

Input:not(input[type=”checkbox”]){
   Width: 100%;
}

Textarea{
   Width: 100%;
   Height: 270px;
   Resize: none;
   Padding: 10px 20px;
}

.product-info{
   Width: 100%;
   Height: 500px;
   Display: grid;
   Grid-template-columns: .75fr 1.25fr;
   Grid-template-rows: repeat(2, 1fr);
   Grid-gap: 20px;
   Margin-bottom: 20px;
}
```

```
.product-image{
   Display: flex;
   Justify-content: center;
   Align-items: center;
   Background: #fff;
   Background-size: cover;
   Border-radius: 10px;
   Grid-row: span 2;
   Text-shadow: 0 0 10px #fff;
}

.text{
   Text-transform: capitalize;
   Color: #383838;
   Font-size: 20px;
   Font-weight: 500;
}

.upload-image-sec, .select-sizes{
   Background: #fff;
   Border-radius: 10px;
   Padding: 20px;
}

.text img{
   Height: 20px;
   Margin-right: 10px;
}

.upload-catalouge{
   Width: 100%;
```

```
    Margin: 20px 0;

    Display: grid;

    Grid-template-columns: repeat(4, 100px);

    Grid-gap: 10px;

}


.upload-image{

    Width: 100%;

    Height: 100px;

    Background: #f5f5f5;

    Cursor: pointer;

    Background-size: cover;

}


.upload-image:hover{

    Background: rgba(0, 0, 0, 0.2);

    Background-size: cover;

}



.sizes{

    Margin-top: 30px;

}


.size-checkbox{

    -webkit-appearance: none;

    Width: 100px;

    Height: 40px;

    Border-radius: 5px;

    Border: 1px solid #383838;

    Cursor: pointer;
    Margin-bottom: 10px;

    Margin-right: 10px;

    Position: relative;

    Color: #383838;

}


.size-checkbox::after{

    Content: attr(value);

    Position: absolute;

    Top: 50%;

    Left: 50%;

    Transform: translate(-50%, -50%);

    Font-size: 16px;

    Text-transform: uppercase;

}


.size-checkbox:checked{

    Background: #383838;

    Color: #fff;

}


Input::-webkit-outer-spin-button,

Input::-webkit-inner-spin-button{

    -webkit-appearance: none;

    Margin: 0;

}


.product-price{

    Display: grid;

    Grid-template-columns: repeat(3, 1fr);

    Grid-gap: 20px;
```

```
}

.product-price input{
   Margin: 0;
}

.buttons{
   Margin: 20px 0 50px;
}

.btn{
   Padding: 10px 30px;
   Text-transform: capitalize;
   Color: #fff;
   Background: #383838;
   Border-radius: 5px;
   Border: none;
   Outline: none;
   Margin-right: 10px;
   Cursor: pointer;
}

#save-btn{
   Background: #a9a9a9;
}
Input::-webkit-outer-spin-button,
Input::-webkit-inner-spin-button{
   -webkit-appearance: none;
   Margin: 0;
}
```

Signup.css

```
Input[type=”text”],
Input[type=”password”],
Input[type=”email”],
Input[type=”number”], // add this new line
Textarea{
   // properties
}
```

Form submission

```
Let user = JSON.parse(sessionStorage.user || null);
Let loader = document.querySelector(‘.loader’);

// checknig user is logged in or not
Window.onload = () => {
   If(user){
      If(!compareToken(user.authToken, user.email)){
         Location.replace(‘/login’);
      }
   } else{
      Location.replace(‘/login’);
   }
}
```

Addproduct.js

```
// price inputs

Const actualPrice = document.querySelector('#actual-price');
Const discountPercentage = document.querySelector('#discount');
Const sellingPrice = document.querySelector('#sell-price');
discountPercentage.addEventListener('input', () => {
   if(discountPercentage.value > 100){
      discountPercentage.value = 90;
   } else{
      Let discount = actualPrice.value * discountPercentage.value / 100;
      sellingPrice.value = actualPrice.value – discount;
   }
})
sellingPrice.addEventListener('input', () => {
   let discount = (sellingPrice.value / actualPrice.value) * 100;
   discountPercentage.value = discount;
})
```

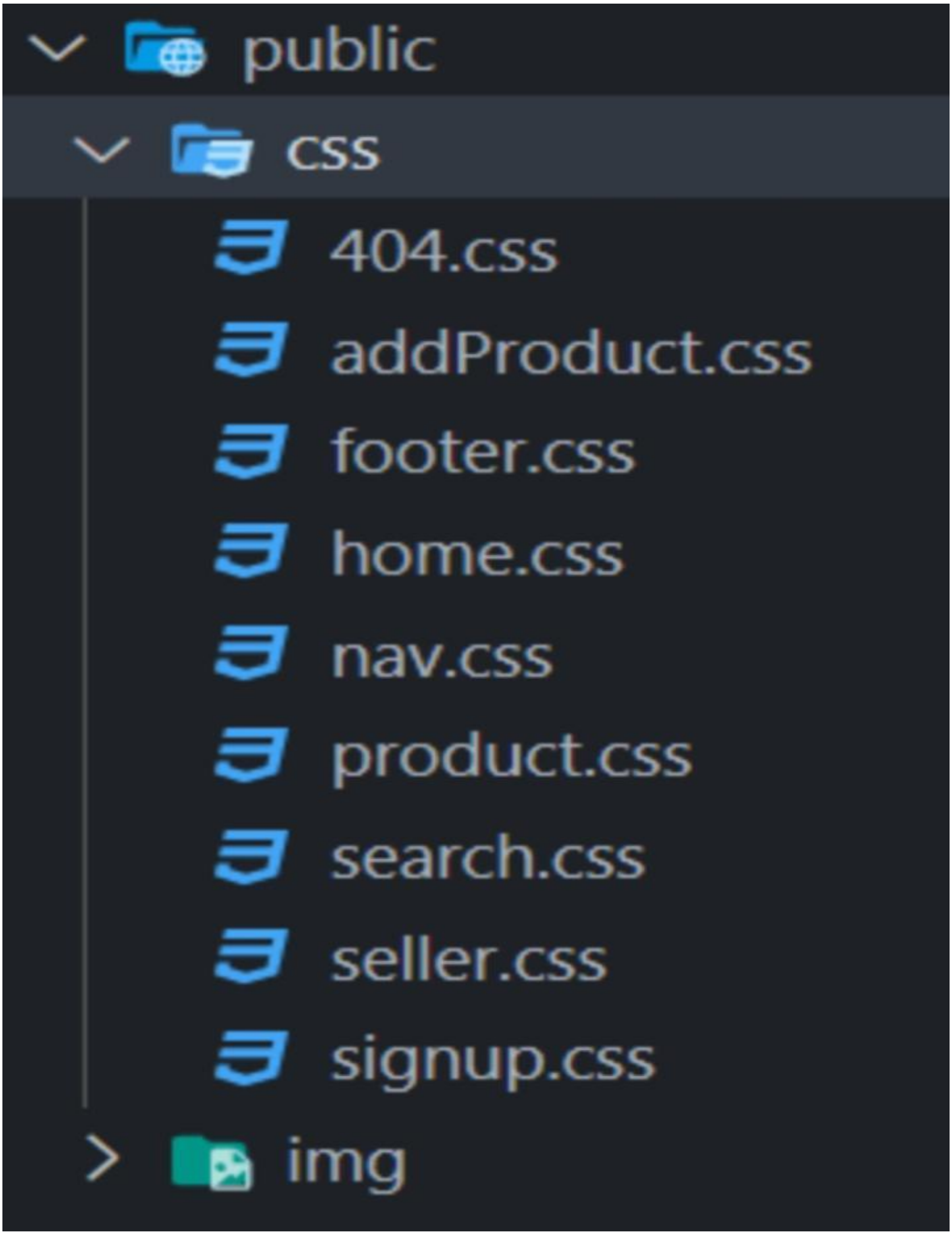
Server.js

AWS Config

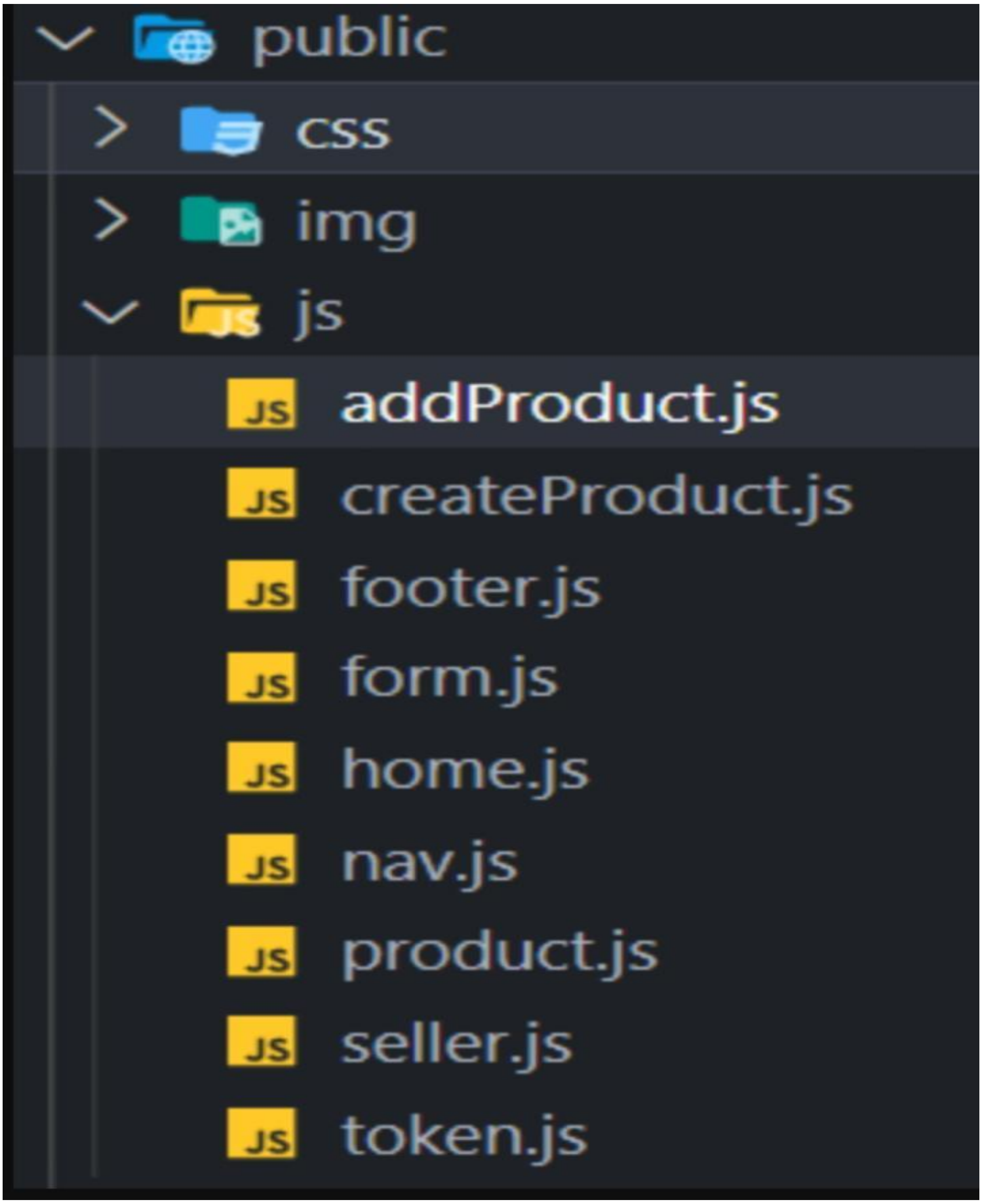
```
// aws config
Const aws = require('aws-sdk');
Const dotenv = require('dotenv');

Dotenv.config();

// aws parameters
```

```
Const region = “ap-south-1”;
Const bucketName = “ecom-website-tutorial-2”;
Const accessKeyId = process.env.AWS_ACCESS_KEY;
Const secretAccessKey = process.env.AWS_SECRET_KEY;

Aws.config.update({
  Region,
  accessKeyId,
  secretAccessKey
})

// init s3
Const s3 = new aws.S3();
// generate image upload link
Async function generateUrl(){
  Let date = new Date();
  Let id = parseInt(Math.random() * 10000000000);

  Const imageName = `${id}${date.getTime()}.jpg`;

  Const params = ({
    Bucket: bucketName,
    Key: imageName,
    Expires: 300, //300 ms
    ContentType: ‘image/jpeg’
  })
  Const uploadUrl = await s3.getSignedUrlPromise(‘putObject’, params);
  Return uploadUrl;
}

// generate image upload link
```

```
Async function generateUrl(){
    Let date = new Date();
    Let id = parseInt(Math.random() * 10000000000);

    Const imageName = `${id}${date.getTime()}.jpg`;

    Const params = ({
        Bucket: bucketName,
        Key: imageName,
        Expires: 300, //300 ms
        ContentType: ‘image/jpeg’
    })
    Const uploadUrl = await s3.getSignedUrlPromise(‘putObject’, params);
    Return uploadUrl;
}

// get the upload link
App.get(‘/s3url’, (req, res) => {
    generateUrl().then(url => res.json(url));
})
// upload image handle
Let uploadImages = document.querySelectorAll(‘.fileupload’);
Let imagePaths = []; // will store all uploaded images paths;
uploadImages.forEach((fileupload, index) => {
    fileupload.addEventListener(‘change’, () => {
        const file = fileupload.files[0];
        let imageUrl;

        if(file.type.includes(‘image’)){
            // means user uploaded an image
        } else{
            showAlert(‘upload image only’);
        }
    })
})
If(file.type.includes(‘image’)){
    // means user uploaded an image
    Fetch(‘/s3url’).then(res => res.json())
    .then(url => {
        Fetch(url,{
            Method: ‘PUT’,
            Headers: new Headers({‘Content-Type’: ‘multipart/form-data’}),
            Body: file
        }).then(res => {
            Console.log(url)
        })
    })
}
If(file.type.includes(‘image’)){
    // means user uploaded an image
    Fetch(‘/s3url’).then(res => res.json())
    .then(url => {
        Fetch(url,{
            Method: ‘PUT’,
            Headers: new Headers({‘Content-Type’: ‘multipart/form-data’}),
            Body: file
        }).then(res => {
            imageUrl = url.split(“?”)[0];
            imagePaths[index] = imageUrl;
            let label = document.querySelector(`label[for=${fileupload.id}]`);
            label.style.backgroundImage = `url(${imageUrl})`;
            let productImage = document.querySelector(‘.product-image’);
            productImage.style.backgroundImage = `url(${imageUrl})`;
        })
    })
}
```



```
public
  css
    404.css
    addProduct.css
    footer.css
    home.css
    nav.css
    product.css
    search.css
    seller.css
    signup.css
  img
```



```
public
  css
  img
  js
    addProduct.js
    createProduct.js
    footer.js
    form.js
    home.js
    nav.js
    product.js
    seller.js
    token.js
```