

SPRINT-4

DATE	28 October 2022
Team ID	PNT2022TMID00855
Project Name	Plasma Donor Application

The following user stories are implemented in this sprint.

- **USN-1: As a user, I can register for the application by entering my email, password, and confirming my password.**
- **USN-2: As a user, I can log into the application using my registered email & password.**

CODE:

logout.html:

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Plasma Donor Application</title>

    <link

      rel="stylesheet"

      href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css"

    />

    <style>

      body {

        background-color: #ffffcc;

      }

    </style>

  </head>

  <body>

    <h1>You Have Been Successfully Logged Out □□□□</h1>

    <form action="/home" method="POST">

      <input type="submit" value="HOME" />

    </form>

  </body>

</html>
```

AccessForm.svelte:

<script>

```
import TextInput from '../components/TextInput.svelte'  
// import { isEmpty } from '../helpers/validation.js'  
// import { isValidEmail } from '../helpers/validation.js'  
// import { isValidAge } from '../helpers/validation.js'  
// import { isValidPassword } from '../helpers/validation.js'  
// import { isValidConfirmPassword } from '../helpers/validation.js'
```

```
let userName = ""  
let userEmail = ""  
let userDob = ""  
let userGender  
let phoneNumber = ""  
let userAge = ""  
let userAddress = ""  
let userPassword = ""  
let userConfirmPassword = ""
```

```
let validUserName = 'initial'  
let validUserEmail = 'initial'  
let validUserDob = 'initial'  
let validUserGender = 'initial'  
let validPhoneNumber = 'initial'  
let validUserAge = 'initial'  
let validUserAddress = 'initial'  
let validUserPassword = 'initial'  
let validUserConfirmPassword = 'initial'
```

```
$: validUserName = isEmpty(userName)

$: validUserDob = isEmpty(userDob)

$: validUserGender = isEmpty(userDob)

$: validPhoneNumber = isEmpty(phoneNumber)

$: validUserAddress = isEmpty(userAddress)

$: validUserEmail = isValidEmail(userEmail)

$: validUserAge = isValidAge(userAge)

$: validUserPassword = isValidPassword(userPassword)

$: validUserConfirmPassword = isValidConfirmPassword(userConfirmPassword)
```

```
function isEmpty(val) {
  if (val.trim().length === 0) {
    return 'invalid'
  } else {
    return 'valid'
  }
}
```

```
function isValidEmail(val) {
  let state = new RegExp(
    "[a-z0-9!#$%&'*/+=?^_`{|}~-]+(?:[a-z0-9!#$%&'*/+=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?"
  ).test(val)

  if (state) {
    return 'valid'
  } else {
    return 'invalid'
  }
}
```

```
}
```

```
function isValidAge(val) {
```

```
  if (val < 18) {
```

```
    // validUserAge = "Age is below 18,Not Eligible"
```

```
    return 'invalid'
```

```
  } else if (val > 50) {
```

```
    // validUserAge = "Age is above 50,Not Eligible"
```

```
    return 'invalid'
```

```
  }
```

```
  return 'valid'
```

```
}
```

```
function isValidPassword(val) {
```

```
  let state = new RegExp(
```

```
    '^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&*])(?={8,})'
```

```
  ).test(val)
```

```
  if (state) {
```

```
    return 'valid'
```

```
  } else {
```

```
    return 'invalid'
```

```
  }
```

```
}
```

```
function isValidConfirmPassword(val) {
```

```
  if (val === userPassword) {
```

```
    return 'valid'
```

```
  } else {
```

```

        return 'invalid'
    }
}
</script>
<form action="" class="container relative modal-box" on:submit|preventDefault>
    <h1 class="mb-4 mr-auto mt-16 text-3xl uppercase md:mt-0 md:mb-8">
        Access Form
    </h1>
    <label
        for="accessform-modal"
        class="absolute btn btn-sm btn-circle right-2 top-2"
    >✕
    </label>
    <!-- Username -->
    <TextInput
        type="text"
        label="Name:"
        id="name"
        name="username"
        placeholder="UserName"
        inputIcon="fa-solid fa-user"
        valid={ validUserName }
        validityMessage="Please Fill in Your Name"
        value={ userName }
        on:input={ (event) =>
            {
                userName = event.target.value
            }
        }
    />

```

<!-- Email ID -->

```
<TextInput
  type="email"
  label="Email ID:"
  id="email"
  name="email"
  placeholder="example@123.com"
  inputIcon="fa-solid fa-envelope"
  valid={ validUserEmail }
  validityMessage="Please Fill in Your Email ID"
  value={ userEmail }
  on:input={ (event) =>
    {
      userEmail = event.target.value
    }
  }
/>
```

<!-- Mobile number -->

```
<TextInput
  type="number"
  label="Mobile:"
  id="mobile no"
  name="mobile"
  placeholder="Phone number"
  inputIcon="fa-solid fa-phone"
  valid={ validPhoneNumber }
  validityMessage="Please Fill in Your Mobile number"
  value={ phoneNumber }
  on:input={ (event) =>
    {
```

```
phoneNumber = event.target.value
}}
/>
<!-- Address -->
<TextInput
  contentType="descriptiveType"
  name="address"
  label="Address:"
  id="Address:"
  placeholder="Address"
  inputIcon="null"
  valid={ validUserAddress }
  validityMessage="Please mention your address"
  value={ userAddress }
  on:input={ (event) =>
    {
      userAddress = event.target.value
    }
  }
/>
<!-- Password -->
<TextInput
  type="password"
  label="Password:"
  id="createPassword"
  name="password"
  placeholder="Password@123"
  inputIcon="fa-solid fa-lock"
  valid={ validUserPassword }
  validityMessage="Use a strong password"
```



```
        value={userPassword}
        on:input={ (event) =>
        {
        userPassword = event.target.value
        }}
    />
    <div class="buttons">
        <button type="submit">Submit</button>
        <button
            type="reset"
            on:click={() => {
            validUserName =
            validUserEmail =
            validUserDob =
            validPhoneNumber =
            validUserAge =
            validUserPassword =
            validUserConfirmPassword =
            'initial'
            } }>Reset</button>
        >
    </div>
</form>
<style>
.container {
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
```

```
width: 90%;
max-width: 700px;
padding: 3em 1.5em 1.5em;
font-size: 1.25rem;
border-radius: 1rem;
border: 2px solid var(--clr-primary-400);
}

.buttons {
display: flex;
justify-content: space-between;
width: 75%;
max-width: 400px;
}

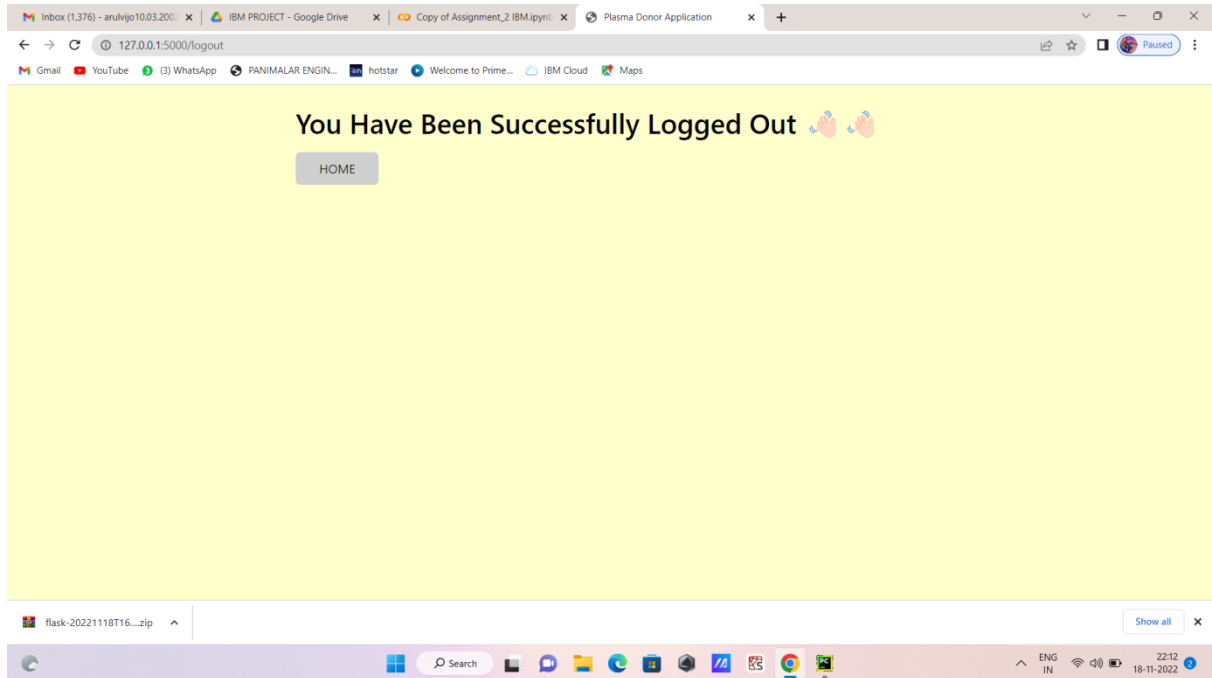
.buttons * {
padding: 0.3em 1.2em;
color: var(--clr-primary);
background-color: var(--clr-primary-400);
font-size: 1.175rem;
font-weight: 600;
text-transform: uppercase;
border-radius: 100vmax;
box-shadow: 0 5px 10px rgba(0, 0, 0, 0.5);
}

@media (max-width: 50em) {
.modal-box {
max-height: max-content;
padding: 1em;
top: 2em;
}
}
```

```
.modal-box h1 {  
margin: 0 auto 0.5em 0;  
}  
.buttons {  
width: 100%;  
}  
}  
</style>
```

SCREENSHOTS:

Logout page:



AccessForm:

