

APT1030-B: FUNDAMENTALS OF PROGRAMMING LANGUAGES

Instructor: Edward Ombui PhD.

Login Interface

The following group assignment is due at 8:00 a.m. (Kenya time) on 20th May,2024:

1. Develop a login application (See figure 1) that can validate both input fields
 - The fields should not be empty on submission
 - The username field should be a valid email address
 - The password should be more than 8 characters
 - Validation should employ three alternatives:
 - Functional,
 - OOP,
 - Procedural programming

Procedural programming

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, ini"/>
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Programming Paradigms</title>
    <link rel="stylesheet" href="app.css"/>
    <script src="procedural.js" defer></script>
  </head>
  <body>
    <form id="user-input">
      <div class="form-control">
        <label for="username">Username</label>
        <input type="text" id="username" />
      </div> <br>
      <div class="form-control">
        <label for="password">Password</label>
        <input type="text" id="password" />
        <p>*password must be at least 8 characters and contain figures
and symbols</p>
      </div>
      <button type="submit">Create User</button>
    </form>
  </body>
```

```
</html>
```

procedural.js

```
const form = document.getElementById('user-input');

function signupHandler(event) {
  event.preventDefault();

  const userNameInput = document.getElementById('username');
  const enteredUserName = userNameInput.value;

  const passwordInput = document.getElementById('password');
  const enteredPassword = passwordInput.value;

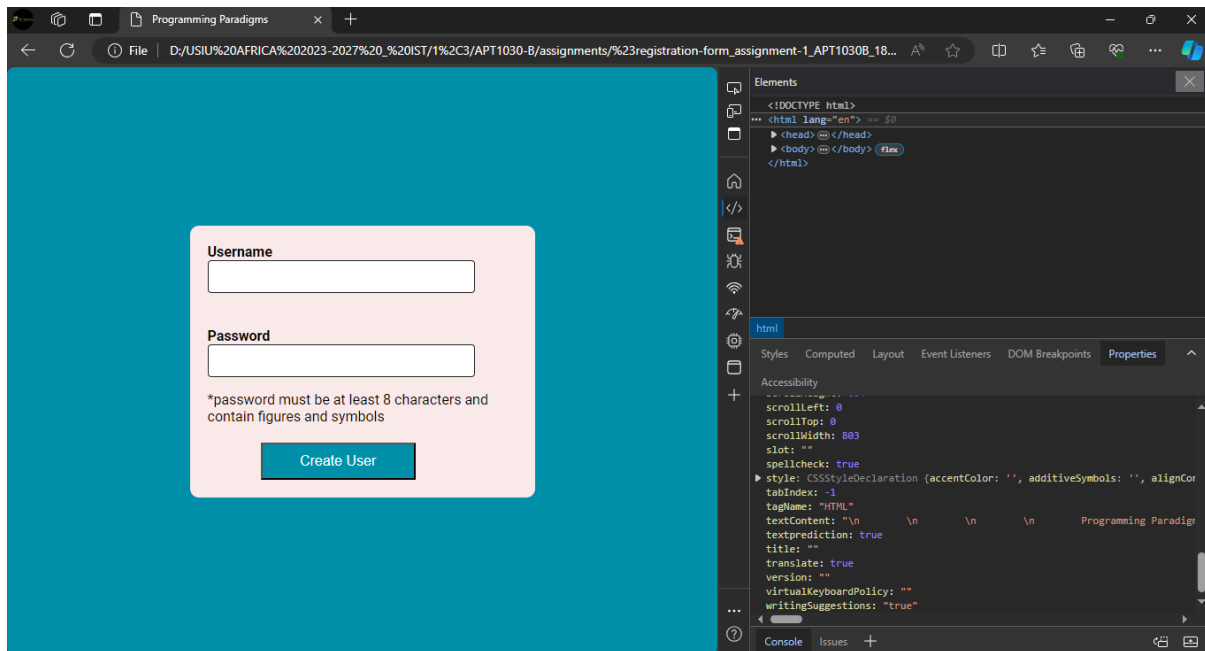
  if (enteredUsername.trim().length === 0) {
    alert('Invalid input - username must not be empty!');
    return;
  }
  if (enteredPassword.trim().length <= 5) {
    alert('Invalid input - password must be six characters or longer. |');
    return;
  }

  const user = {
    userName: enteredUsername,
    password: enteredPassword
  };

  console.log(user);
  console.log('Hi, I am ' + user.userName);
}

form.addEventListener('submit', signupHandler);
```

Output



Functional programming

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, ini"/>
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Programming Paradigms</title>
    <link rel="stylesheet" href="app.css"/>
    <script src="functional.js" defer></script>
  </head>
  <body>
    <form id="user-input">
      <div class="form-control">
        <label for="username">Username</label>
        <input type="text" id="username" />
      </div> <br>
      <div class="form-control">
        <label for="password">Password</label>
        <input type="text" id="password" />
        <p>*password must be at least 8 characters and contain figures
and symbols</p>
      </div>
      <button type="submit">Create User</button>
    </form>
  </body>
</html>
```

functional.js

```
const REQUIRED = 'REQUIRED';
const MIN_LENGTH = 'MIN_LENGTH';

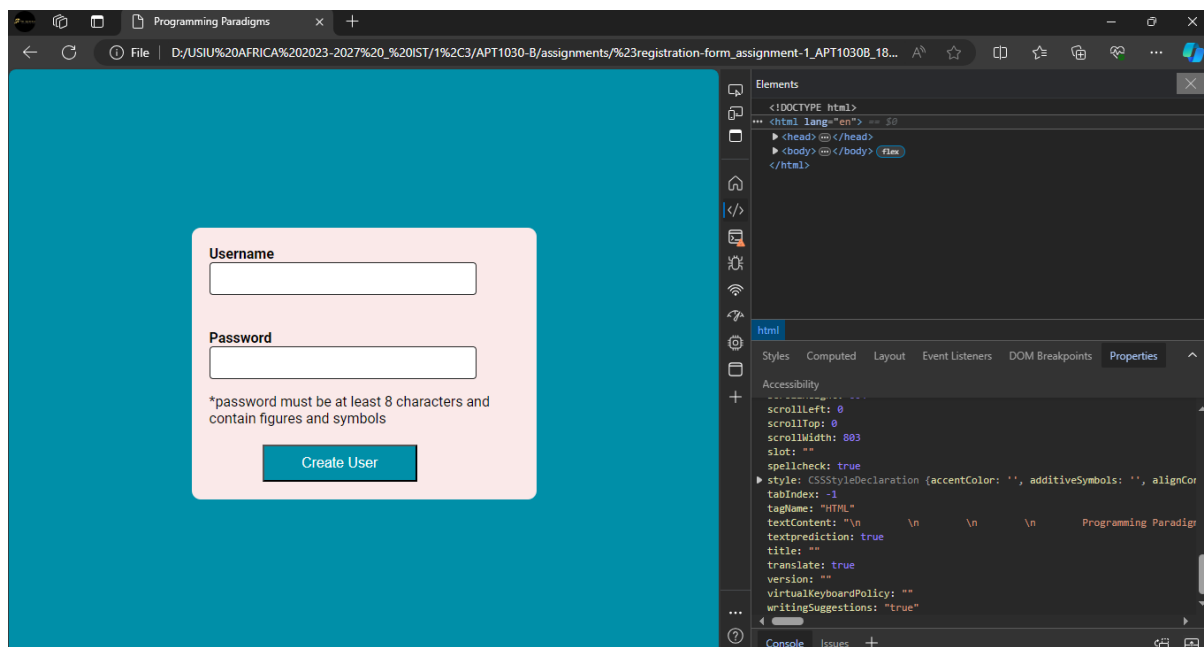
function validate(value, flag, validatorValue) {
  if (flag === REQUIRED) {
    return value.trim().length > 0;
  }
  if (flag === MIN_LENGTH) {
    return value.trim().length > validatorValue;
  }
}

function getUserInput(inputElementId) {
  return document.getElementById(inputElementId).value;
}

function signupHandler(event) {
  event.preventDefault();

  const enteredUsername = getUserInput('username');
  const enteredPassword = getUserInput('password');
}
```

Output



Object oriented programming

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, ini"/>
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Programming Paradigms</title>
    <link rel="stylesheet" href="app.css"/>
    <script src="oop.js" defer></script>
  </head>
  <body>
    <form id="user-input">
      <div class="form-control">
        <label for="username">Username</label>
        <input type="text" id="username" />
      </div> <br>
      <div class="form-control">
        <label for="password">Password</label>
        <input type="text" id="password" />
        <p>*password must be at least 8 characters and contain figures
and symbols</p>
      </div>
      <button type="submit">Create User</button>
    </form>
  </body>
</html>
```

oop.js

```
class Validator {
  static REQUIRED = "REQUIRED";
  static MIN_LENGTH = "MIN_LENGTH";

  static validate(value, flag, validatorValue) {
    if (flag === this.REQUIRED) {
      return value.trim().length > 0;
    }

    return false;
  }
}

class User {
  constructor(uName, uPassword) {
    this.userName = uName;
```

```
        this.password = uPassword;
    }

    greet() {
        console.log("Hello my name is" + this.userName);
    }
}

class UserInputForm {
    constructor() {
        this.form = document.getElementById('user-input');
        this.userNameInput = document.getElementById('username');
        this.passwordInput = document.getElementById('password');

        this.form.addEventListener('submit', (event) =>
this.signupHandler(event));
    }

    signupHandler(event) {
        event.preventDefault();
        const enteredUsername = this.userNameInput.value;
        const enteredPassword = this.passwordInput.value;

        if (
            !Validator.validate(enteredUsername, Validator.REQUIRED) ||
            Validator.validate(enteredPassword, Validator.REQUIRED) ||
            Validator.validate(enteredPassword, Validator.MIN_LENGTH, 6)
        ) {
            alert('Invalid input - username or password is wrong (password
should be at least 8 characters or longer).');
            return;
        }

        const newUser = new User(enteredUsername, enteredPassword);
        newUser.greet();
        console.log(newUser);
    }
}
```

Output

