

Java Advanced

28/08/2025

Prof. Dr. Marcel Stefan Wagner

Aula 08 – Spring Security, Login e Sign In

FIAP

Usando Spring Security e mudando usuário e senha

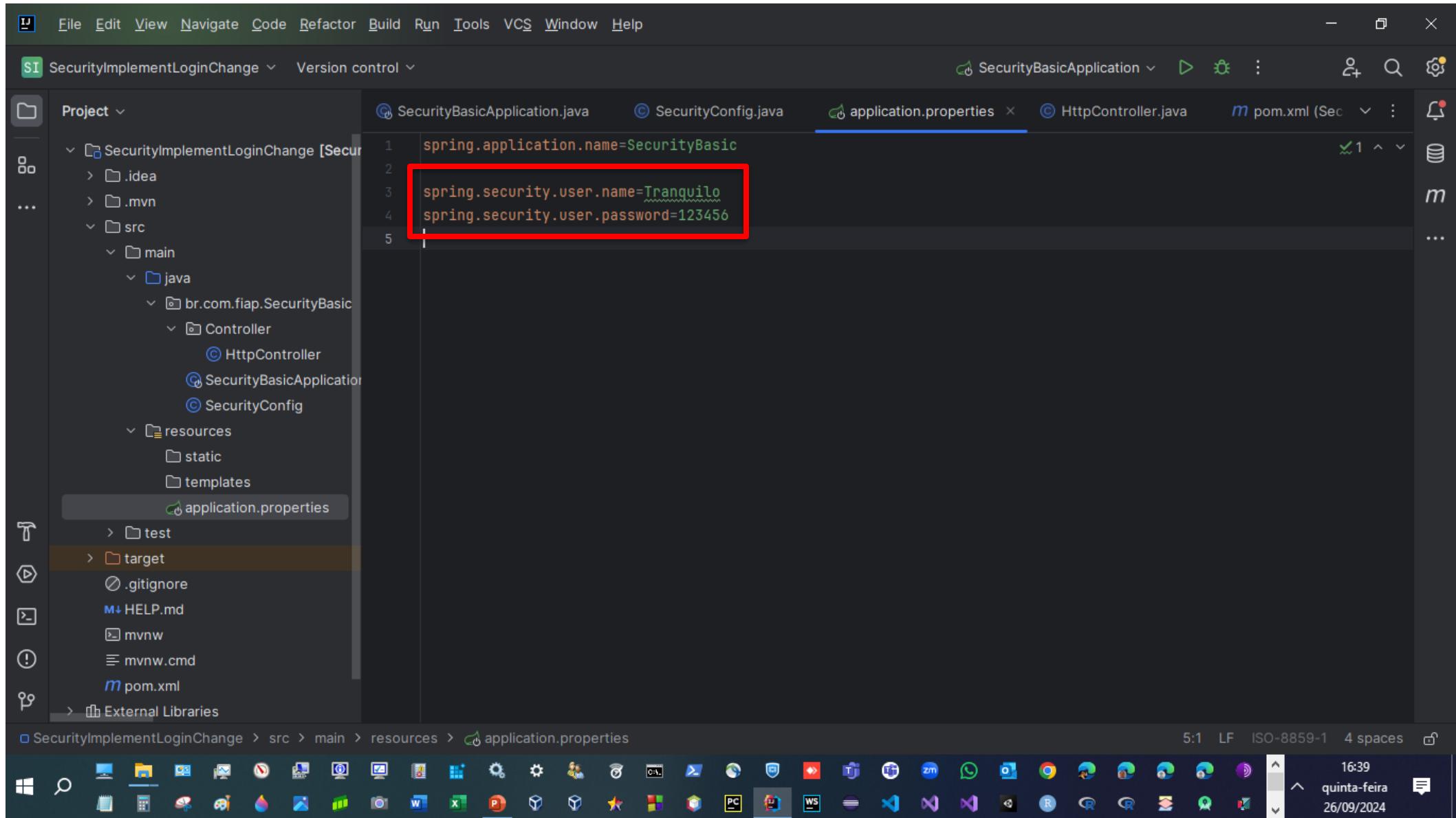
Conceitos

Spring Security

- Spring Security

Usando o *Spring Security*, mas querendo alterar o usuário e a senha para personalizar a aplicação na interface de *login* que o *framework* fornece, basta alterar o *application.properties* e acrescentar essas informações:

```
spring.security.user.name=<Seu_Usuário>
spring.security.user.password=<Sua_Senha>
```



The screenshot shows a Java development environment with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** SecurityImplementLoginChange, Version control.
- Central Area:** Editor showing `application.properties`. A red box highlights the following configuration:

```
spring.application.name=SecurityBasic
spring.security.user.name=Tranquilo
spring.security.user.password=123456
```
- Left Sidebar (Project Explorer):** Shows the project structure:
 - Project: SecurityImplementLoginChange [Secure]
 - src
 - main
 - java
 - br.com.fiap.SecurityBasic
 - HttpController
 - SecurityBasicApplication
 - SecurityConfig
 - resources
 - static
 - templates
 - test
 - target
 - .gitignore
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - External Libraries
- Bottom Status Bar:** SecurityImplementLoginChange > src > main > resources > application.properties, 5:1 LF ISO-8859-1 4 spaces, 16:39 quinta-feira 26/09/2024.
- Taskbar:** Shows various application icons.

The screenshot shows an IDE interface with the following components:

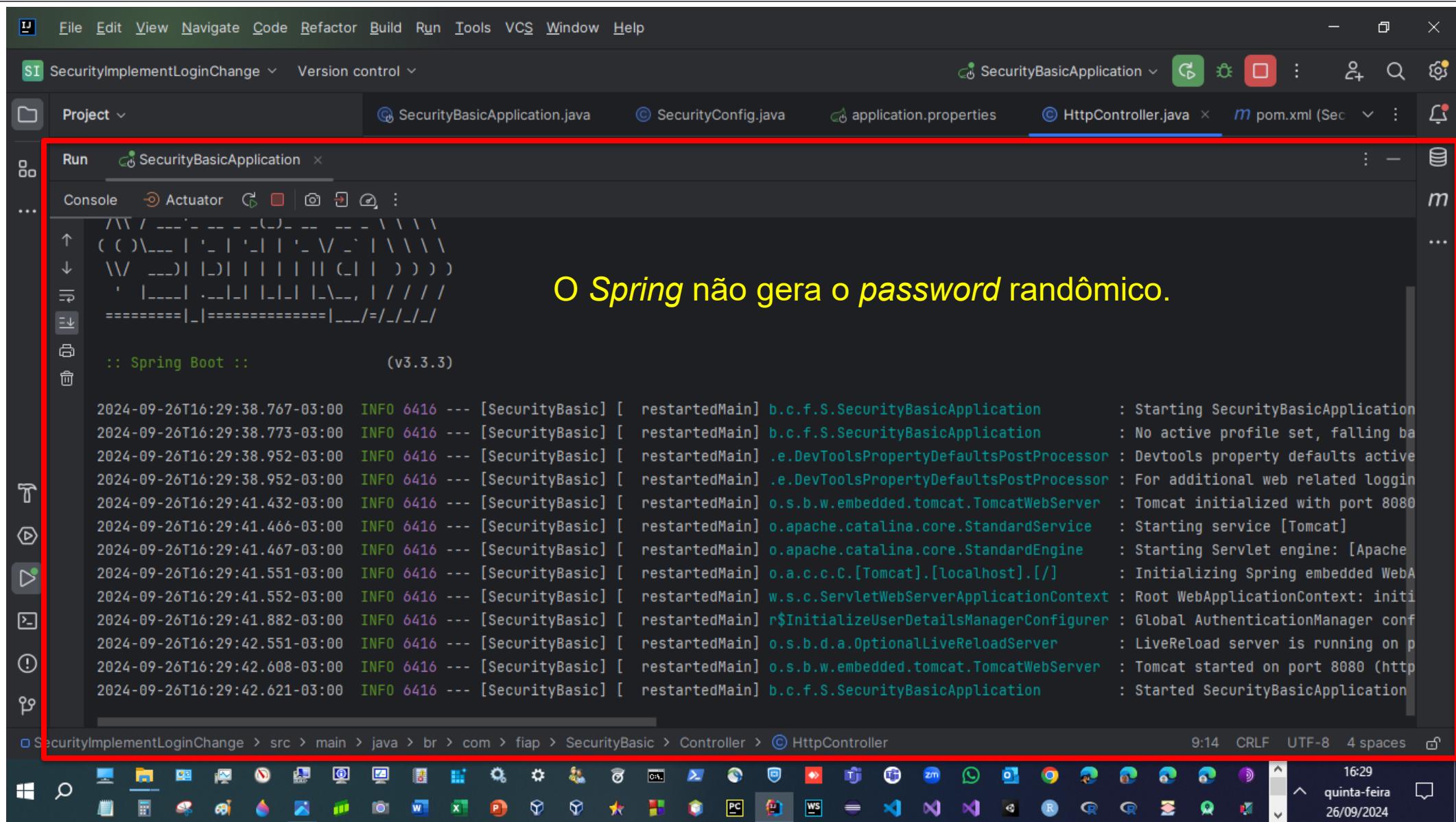
- Top Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Explorer:** Shows the project structure for "SecurityImplementLoginChange".
- Code Editor:** Displays `HttpController.java` with the following code:package br.com.fiap.SecurityBasic.Controller;

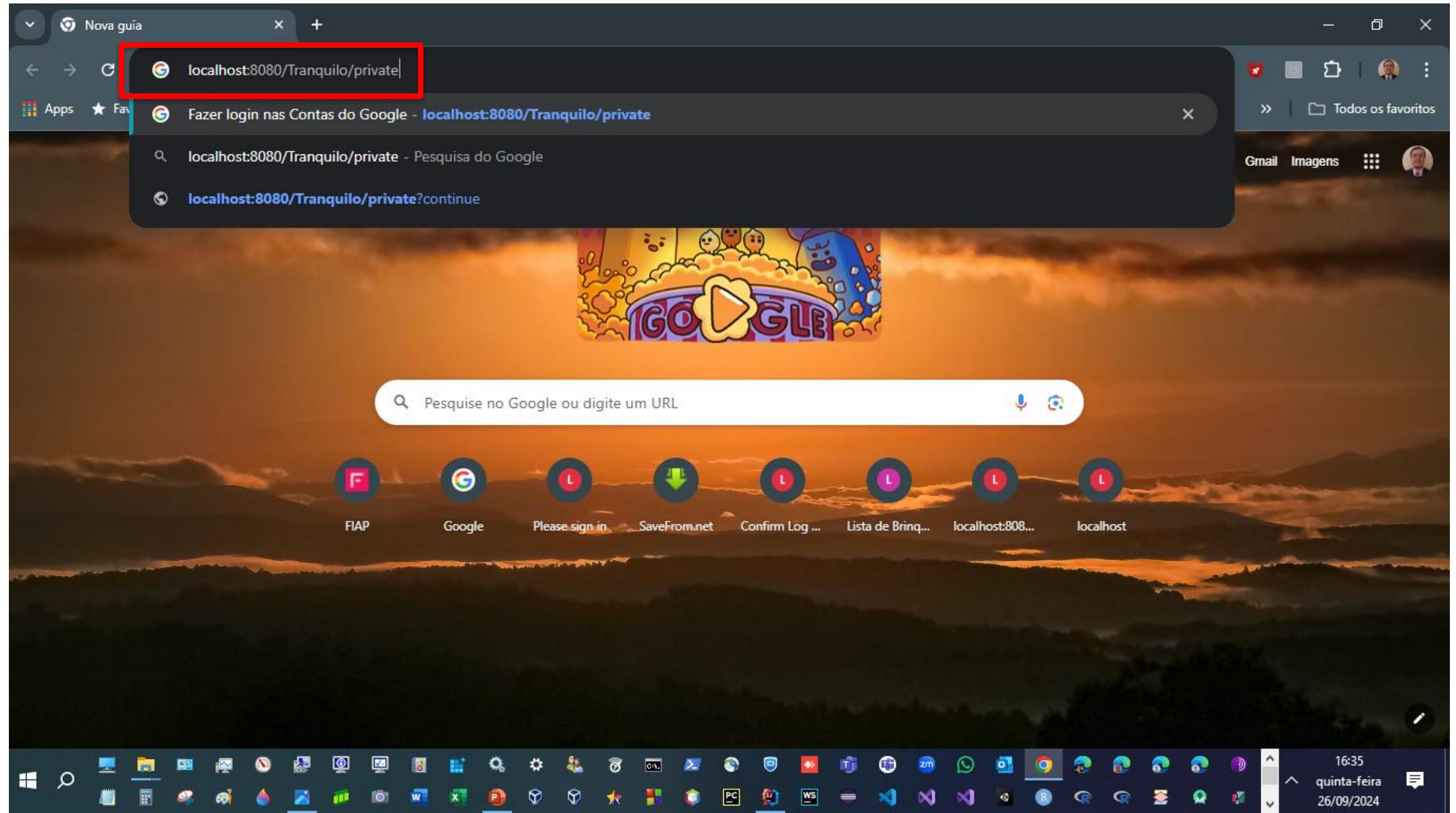
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

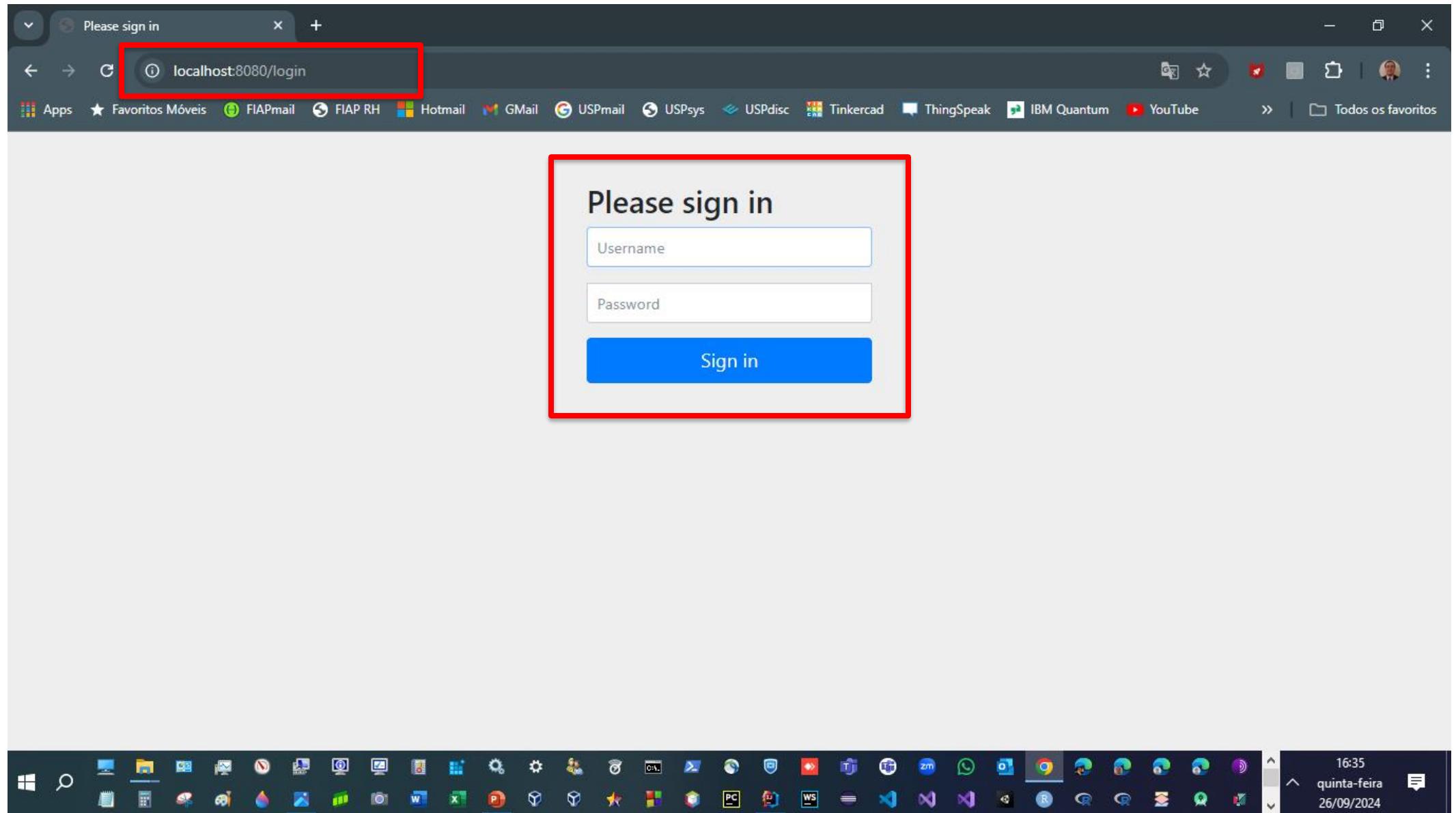
@RestController
@RequestMapping("/Tranquilo")
public class HttpController {

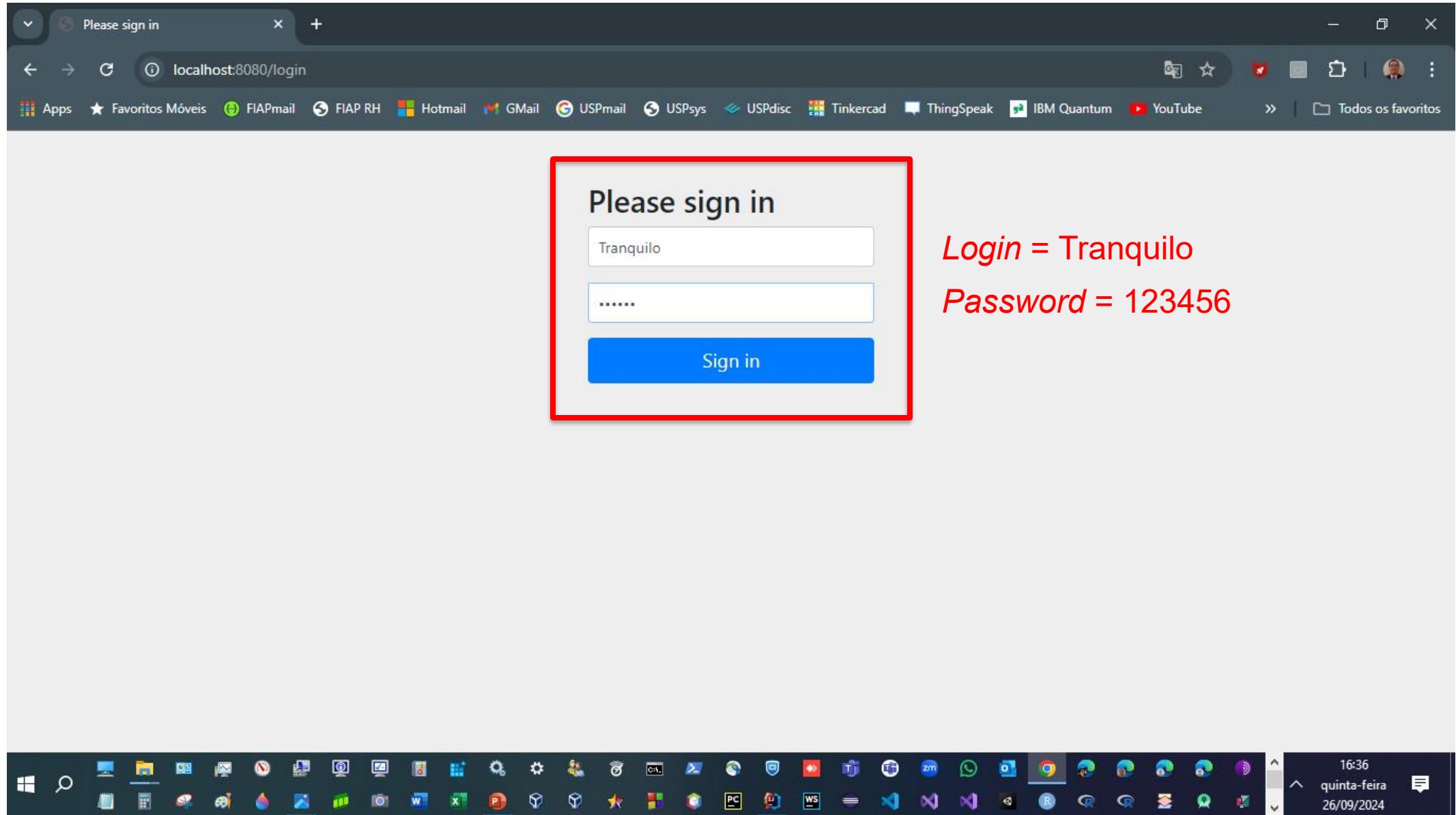
 @GetMapping("/public")
 String publicRoute() { return "<h1>Calma! É uma rota Pública. Fique Tranquilo! </h1>"; }

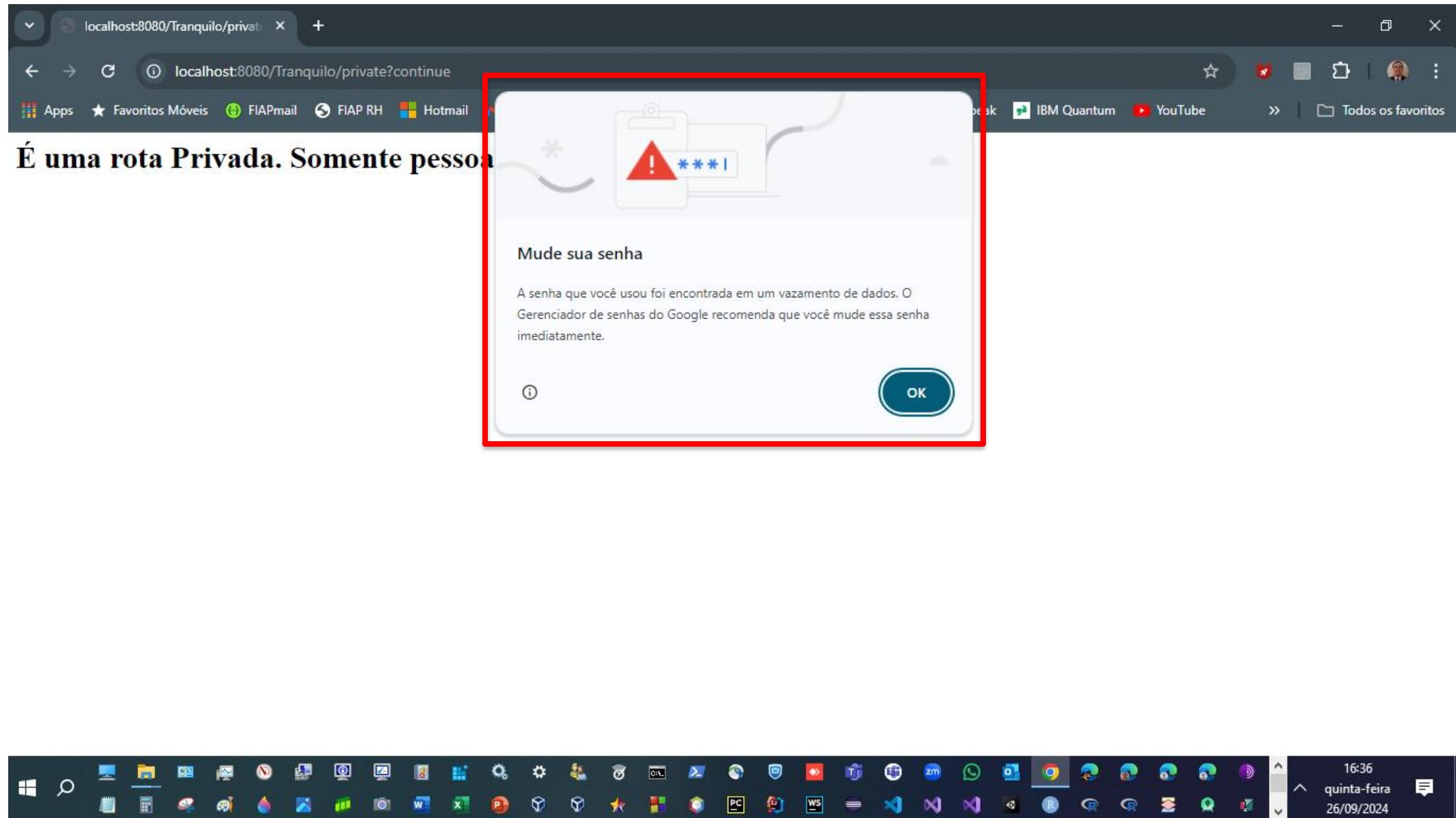
 @GetMapping("/private")
 String privateRoute() { return "<h1>É uma rota Privada. Somente pessoas autorizadas pelo Tranquilo! </h1>"; }
}
- Run Tab:** Shows the application is running ("SecurityBasicApplication").
- Console Tab:** Displays the application's startup logs and standard output. The logs show the application starting up on port 8080, Tomcat service starting, and the servlet engine starting. The log entries are:2024-09-26T16:29:41.452-03:00 INFO 6410 --- [SECU... 1 RESTARTING [Tomcat] [localhost] [Tomcat] [localhost] : Starting service [Tomcat]
2024-09-26T16:29:41.466-03:00 INFO 6410 --- [SecurityBasic] [restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-26T16:29:41.467-03:00 INFO 6410 --- [SecurityBasic] [restartedMain] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache
2024-09-26T16:29:41.551-03:00 INFO 6410 --- [SecurityBasic] [restartedMain] o.a.c.u.Tomcat [localhost] [/] : Initializing Spring embedded Web
SecurityImplementLoginChange > src > main > java > br > com > fiap > SecurityBasic > Controller > HttpController
- Bottom Taskbar:** Shows various system icons for file operations, network, and system status.

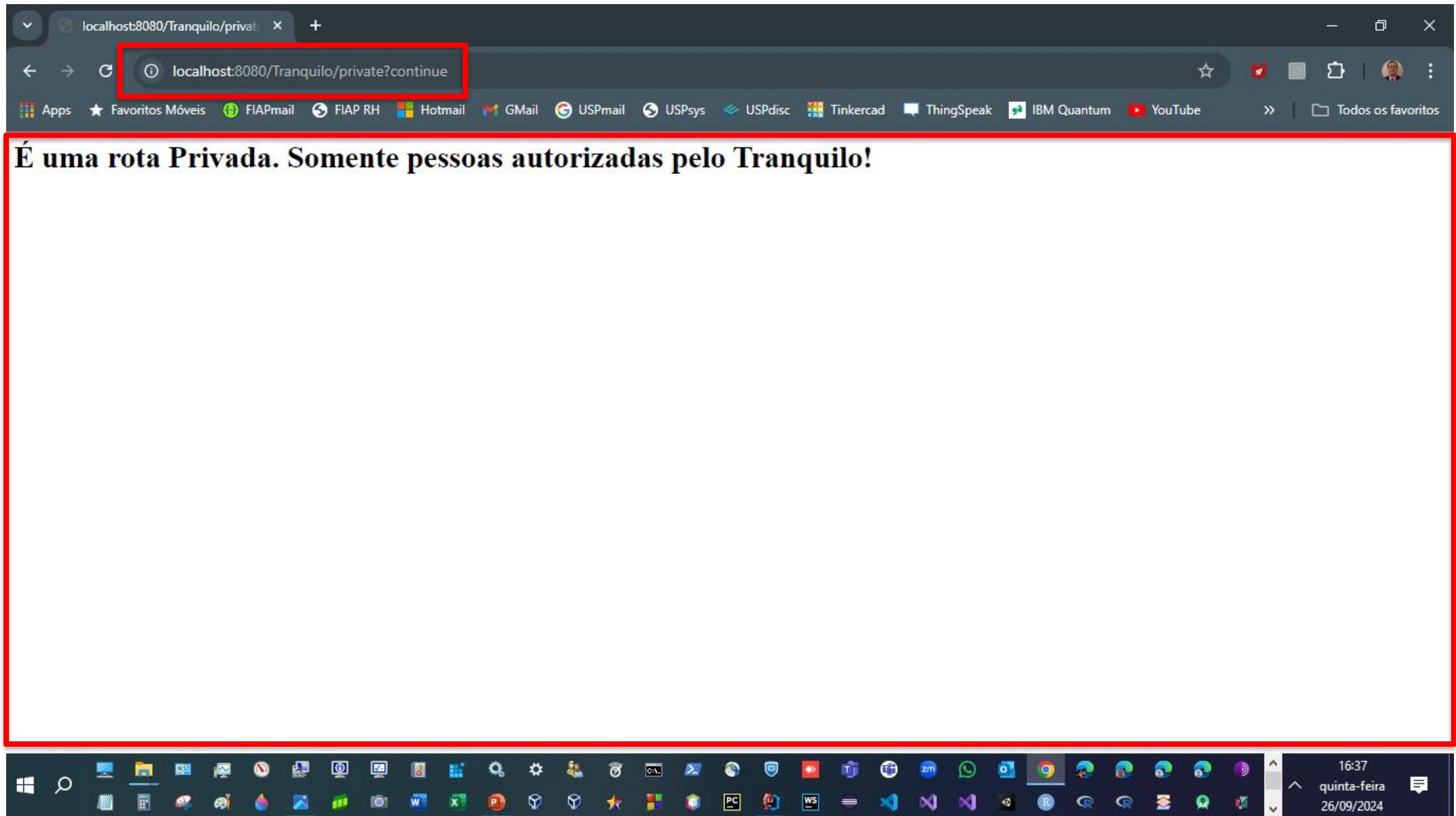


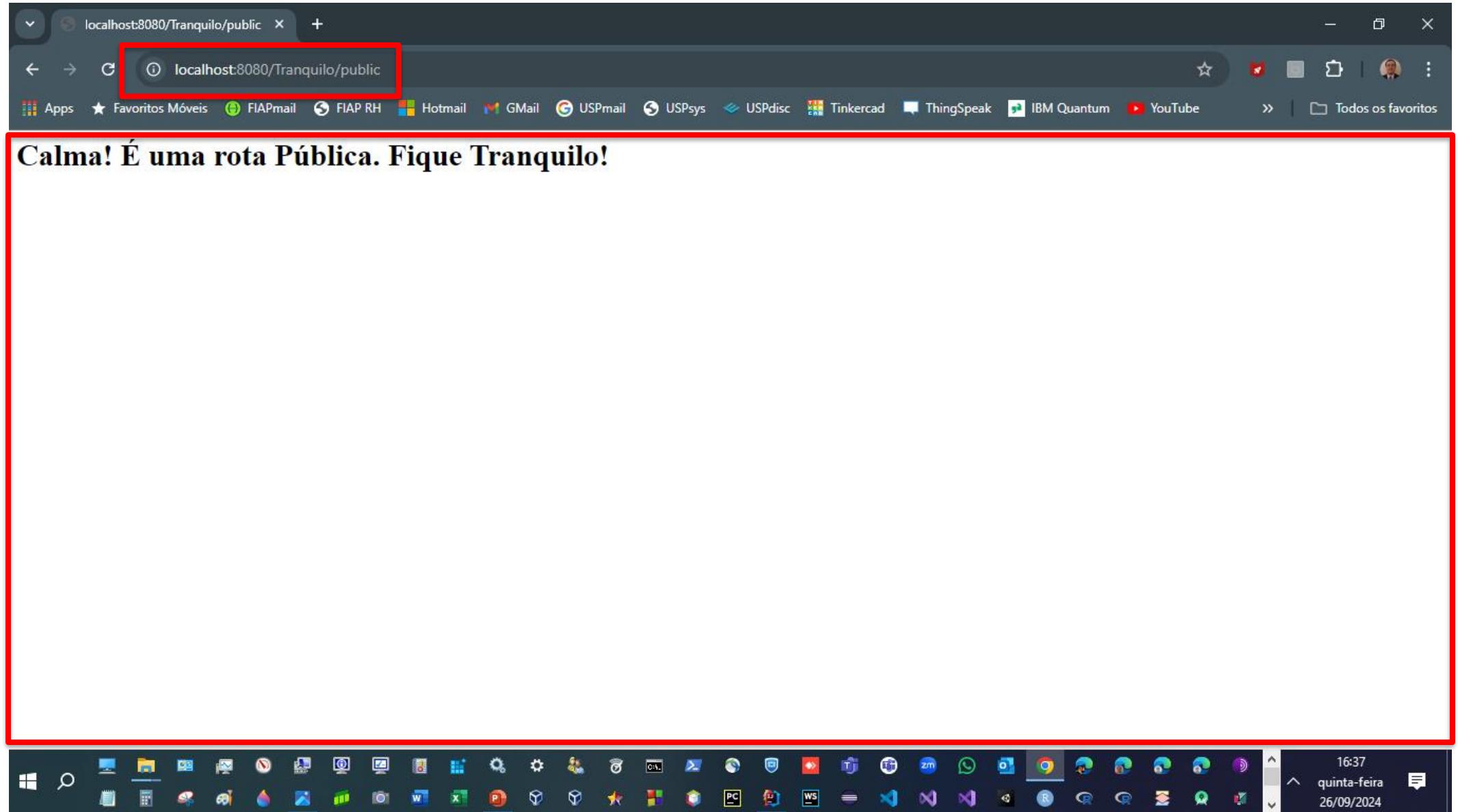


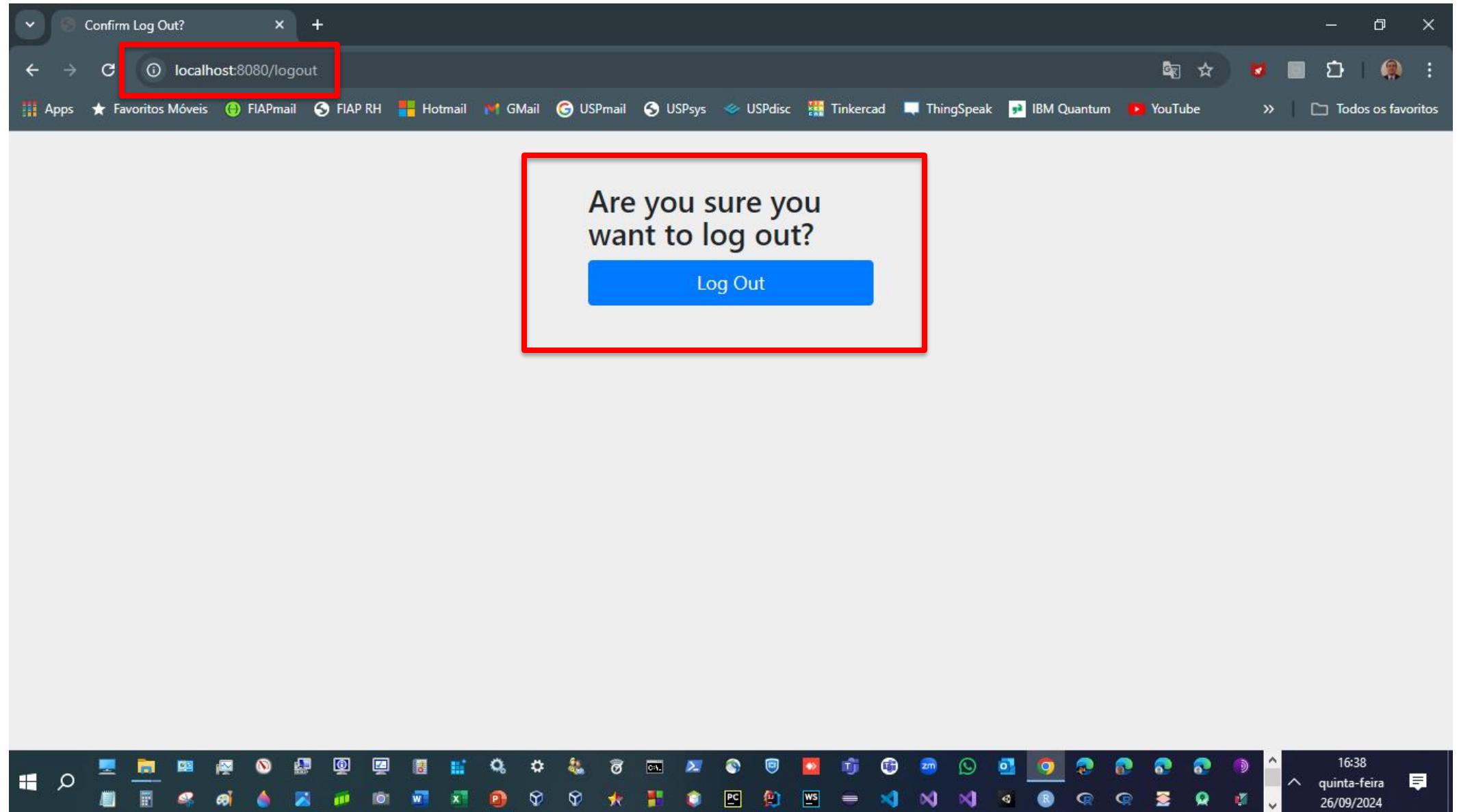


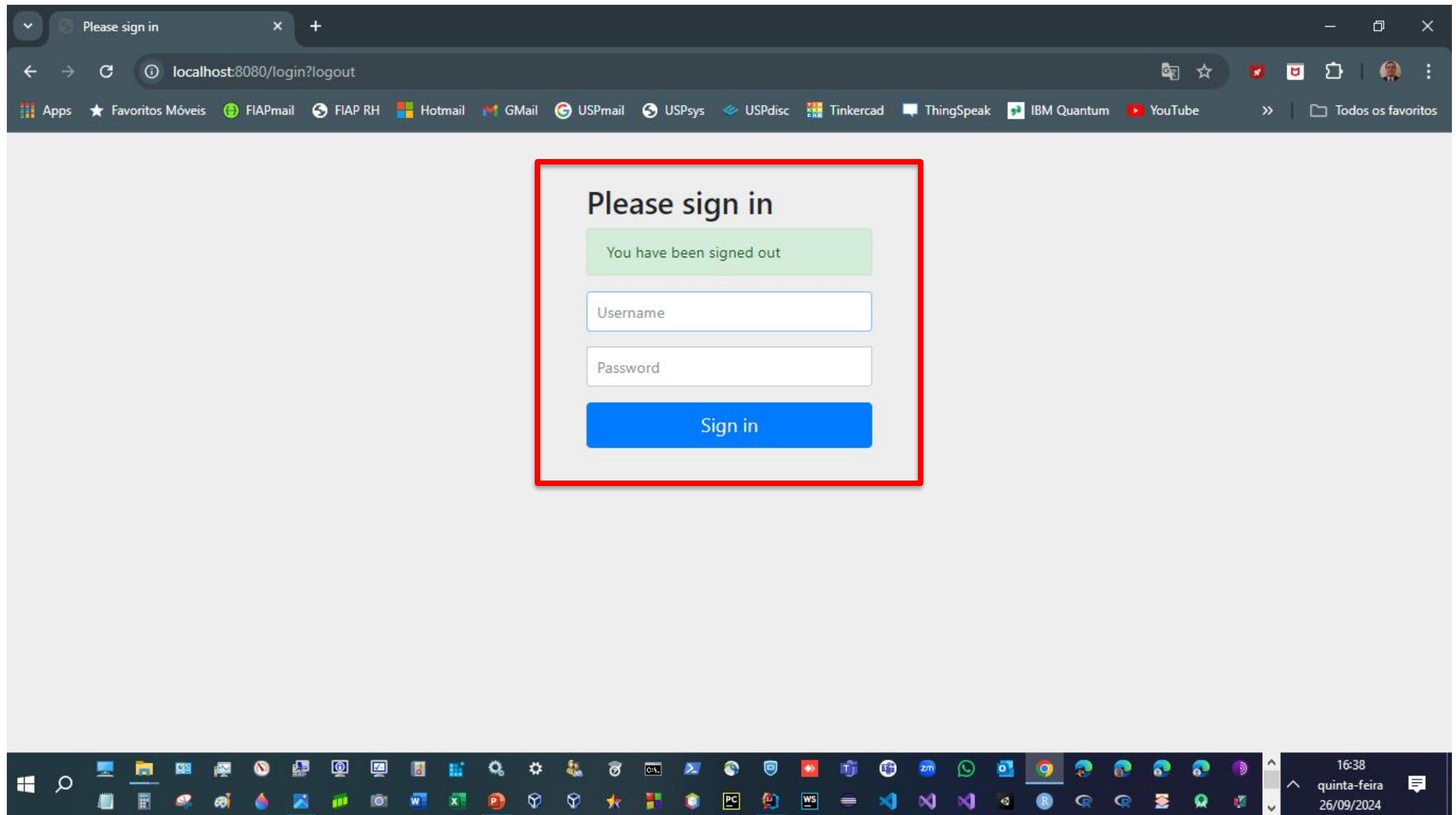










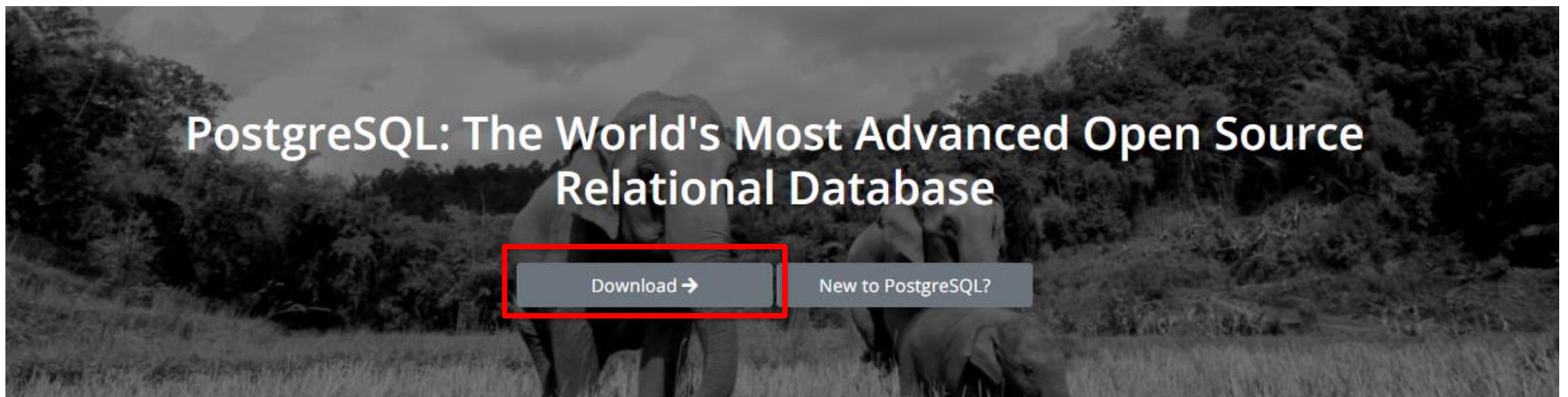


Criando uma interface de *Sign In* e *Login* com o *Spring Security* e usando BD PostgreSQL

Conceitos

PostgreSQL

- Trabalhando com outro Banco de Dados relacional

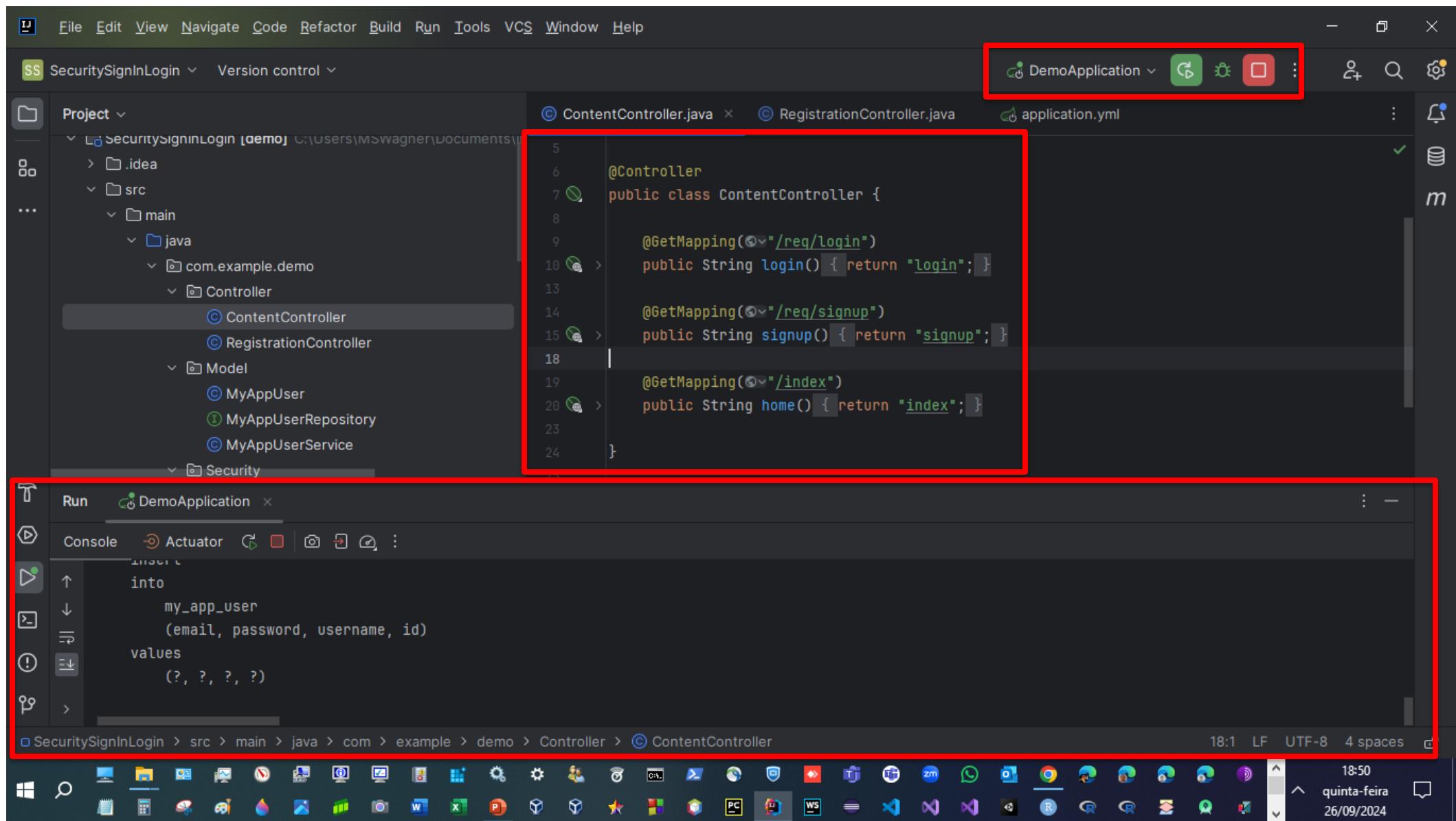


Conceitos

Spring Security

- *Sign In* e *Login*

Para realizar a parte de *Sign In* ou *Sign Up*, deve-se implementar algumas funcionalidades ao projeto com *Spring Security*.



pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Properties SQL Statistics Dependencies Dependents Processes public.my_app_user/securityLogin/postgres@PostgreSQL 17

Domains FTS Configurations FTS Dictionaries FTS Parsers FTS Templates Foreign Tables Functions Materialized Views Operators Procedures Sequences Tables (1) my_app_user Columns (4) id email password username Constraints Indexes RLS Policies Rules Triggers Trigger Functions Types

No limit

Query History

Query Scratch Pad

```
1 SELECT * FROM public.my_app_user
2 ORDER BY id ASC
```

O interessante é que a sua senha não fica visível no BD e nem na comunicação de dados pela Web (codificado).

Data Output Messages Notifications

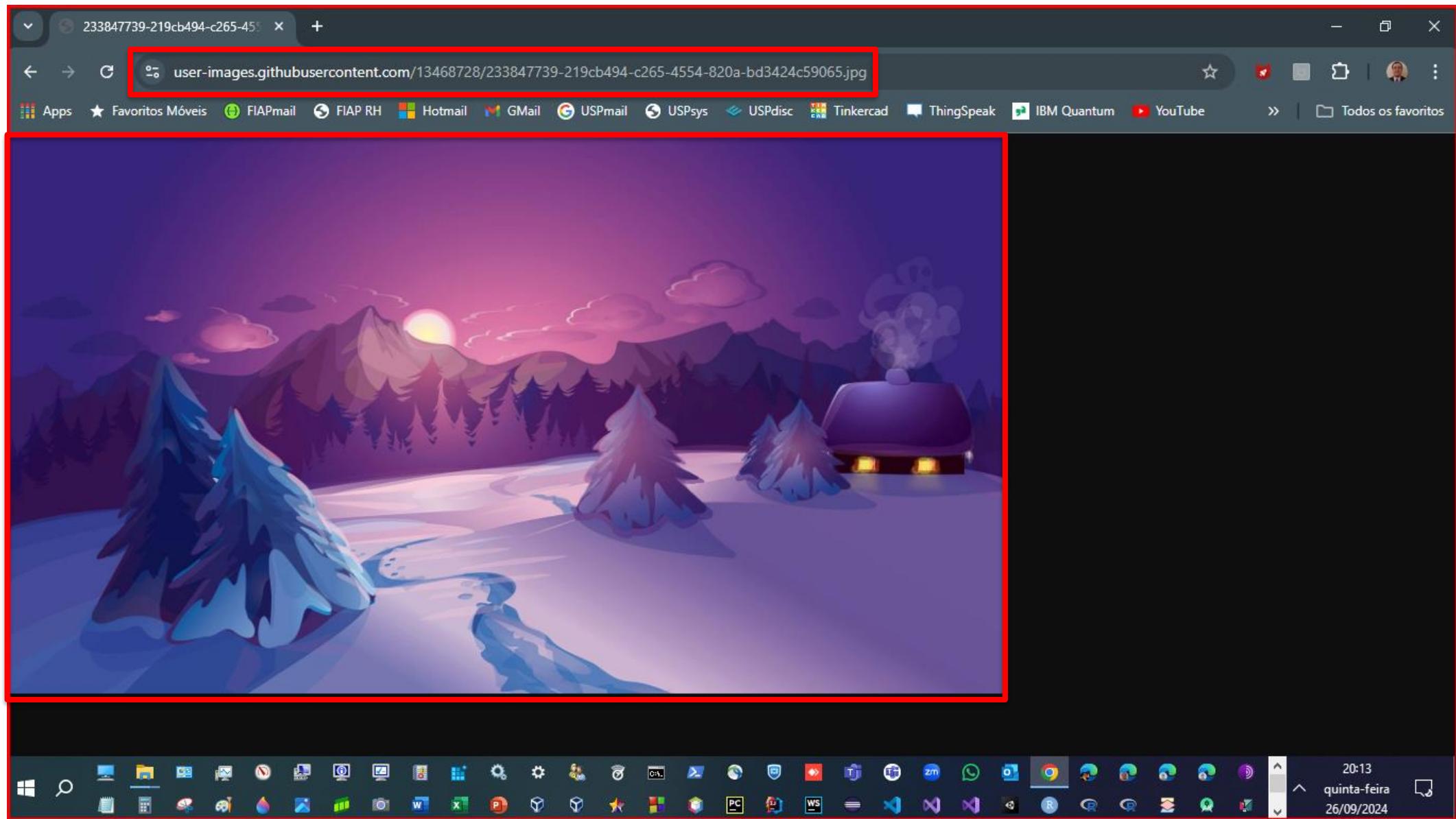
	id [PK] bigint	email character varying (255)	password character varying (255)	username character varying (255)
1	1	profmarcel.wagner@fiap.com.br	\$2a\$10\$LYBEgodAQJhs89SLEyMize93L590bY04ae4iWSF7/s6LGo7luQ4...	Marcel
2	2	tranquilo@yahoo.com	\$2a\$10\$txVnffU6XLvdYzNxh88MT.wvRKxX6/SvhQR4zZwbjEhiovM23X...	Ricardo

Servers > PostgreSQL 17 > Databases > securityLogin > Schemas > public > Tables > my_app_user

18:49 quinta-feira 26/09/2024

The screenshot shows a code editor interface with a dark theme. On the left is a project tree titled "SecuritySignInLogin". It contains several packages: "Controller" (with "ContentController" and "RegistrationController"), "Model" (with "MyAppUser", "MyAppUserRepository", and "MyAppUserService"), "Security" (with "SecurityConfig" and "DemoApplication"), and "resources" (with "public.error" containing "404.html"). Inside "public.error" is a "static" folder, which contains a "css" folder with a "style.css" file. This "style.css" file is highlighted with a red box. The main editor area shows the content of "style.css". A second red box highlights the "body" section of the CSS code, specifically the background image definition.

```
import url('https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap') format('woff2'), url('https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap') format('woff'), url('https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap') format('normal');  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
    font-family: 'Poppins', sans-serif;  
}  
  
body {  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    min-height: 100vh;  
    background-image: url(https://user-images.githubusercontent.com/13468728/233847739-219cb494-  
    background-repeat: no-repeat;  
    background-position: center;  
    background-size: cover;  
}  
  
section {  
    position: relative;  
    max-width: 500px; /* Adjust the max-width as needed */  
    background-color: transparent;  
}
```



The screenshot shows a code editor interface with a dark theme. The left sidebar displays a project structure:

- Project
- Controller
 - ContentController
 - RegistrationController
- Model
 - MyAppUser
 - MyAppUserRepository
 - MyAppUserService
- Security
 - SecurityConfig
 - DemoApplication
- resources
 - public.error
 - 404.html
 - static
 - css
 - style.css
- js
 - signup.js
- templates
 - index.html
 - login.html
 - signup.html
- application.yml

The right pane shows the content of the `signup.js` file. A red box highlights the validation logic for the sign-up button click event:

```
document.addEventListener('DOMContentLoaded', () => {
  const signupForm : HTMLElement = document.querySelector('section');
  signupForm.style.opacity = 0;

  setTimeout(() => {
    signupForm.style.transition = 'opacity 1s ease-in-out';
    signupForm.style.opacity = 1;
  }, 500);

  const signupButton : HTMLButtonElement = document.querySelector('button');
  signupButton.addEventListener('click', () => {
    const emailInput : Element = document.querySelector('input[type="email"]');
    const passwordInput : Element = document.querySelector('input[type="password"]');
    const confirmPasswordInput : Element = document.querySelector('input[type="password"] [name="confirmPassword"]');

    // Check for a valid email and password (you can add your validation logic here)
    const isValid = emailInput.checkValidity() && passwordInput.checkValidity() && confirmPasswordInput.checkValidity();

    if (!isValid) {
      signupForm.classList.add('shake');

      setTimeout(() => {
        signupForm.classList.remove('shake');
      }, 1000);
    }
  });
});
```

The status bar at the bottom indicates the file path: `SecuritySignInLogin > src > main > resources > static > js > signup.js`, encoding: `UTF-8`, and other details like file count (1:1), line endings (LF), and date/time (20:22 quint-feira 26/09/2024).

The screenshot shows a Java IDE interface with a dark theme. The left sidebar displays a project structure:

- Project: SecuritySignInLogin
- Controller: ContentController, RegistrationController
- Model: MyAppUser, MyAppUserRepository, MyAppUserService
- Security: SecurityConfig, DemoApplication
- resources:
 - public.error: 404.html
 - static:
 - css: style.css
 - js: signup.js
 - templates:
 - index.html
 - login.html
 - signup.html
- application.yml

The right side shows the code editor with the file `index.html` open. The code contains HTML and CSS, with some parts highlighted by red boxes:

```
</head>
<body>
    <header>
        <h1>Welcome to My Home Page</h1>
    </header>
    <nav>
        <a href="#home">Home</a>
        <a href="#about">About</a>
        <a href="#services">Services</a>
        <a href="#contact">Contact</a>
    </nav>
    <main>
        <h2>About Me</h2>
        <p>This is a paragraph about me. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
        <h2>Services</h2>
        <p>These are the services I offer. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    </main>
    <footer>
        <p>Footer Content © 2024</p>
    </footer>
</body>
</html>
```

The file path in the bottom navigation bar is: SecuritySignInLogin > src > main > resources > templates > index.html

The status bar at the bottom shows: 63:11 LF UTF-8 4 spaces, 20:20, quint-feira, 26/09/2024.

The screenshot shows a Java IDE interface with a dark theme. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The title bar displays "SecuritySignInLogin" and "Version control". The right side of the title bar shows "DemoApplication" with various status icons and a build summary: 2 errors, 12 warnings, 1 info, and 5 checkmarks.

The left sidebar shows the project structure:

- Project (selected)
- Controller
 - ContentController
 - RegistrationController
- Model
 - MyAppUser
 - MyAppUserRepository
 - MyAppUserService
- Security
 - SecurityConfig
 - DemoApplication
- resources
 - public.error (404.html)
 - static
 - css (style.css)
 - js (signup.js)
 - templates
 - index.html
 - login.html (highlighted with a red box)
 - signup.html

The main editor area displays the content of the "login.html" file:

```
<!DOCTYPE html>
<html lang="en" >
<head>
<meta charset="UTF-8">
<title>Login</title>
<script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
<link rel="stylesheet" href="/css/style.css">
<script src="/js/signup.js" defer></script>
</head>
<body>
<body>
<section>
<form action="/req/login" method="post">
<h1>Login</h1>
<div class="dialog-row">
<label th:if="${param.error}" th:text="${session['SPRING_SECURITY_LAST_EXCEPTION'].message}"></label>
</div>
<div class="inputbox">
<ion-icon name="email-outline"></ion-icon>
<input name="username" id="username" type="name" required>
<label for="username">username</label>
</div>
```

The "login.html" file is highlighted with a red box in the project tree. The "login.html" code in the editor is also highlighted with a red box. The status bar at the bottom shows the file path "SecuritySignInLogin > src > main > resources > templates > login.html", encoding "UTF-8", and a timestamp "20:25 quinta-feira 26/09/2024".

The screenshot shows a code editor interface with a dark theme. At the top, there's a navigation bar with File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. Below the navigation bar is a toolbar with icons for file operations like Open, Save, Find, and Run.

The main area has tabs for Project, Controller, Model, Security, resources, static, and DemoApplication. The DemoApplication tab is active, showing Java code for a controller:

```
16 <h1>Login</h1>
17 <div class="dialog-row">
18   <label th:if="${param.error}" th:text="${session['SPRING_SECURITY_LAST_EXCEPTION'].message}">
19 </div>
20 <div class="inputbox">
21   <ion-icon name="email-outline"></ion-icon>
22   <input name="username" id="username" type="name" required>
23   <label for="username">username</label>
24 </div>
25 <div class="inputbox">
26   <ion-icon name="lock-closed-outline"></ion-icon>
27   <input name="password" type="password" id="password" required>
28   <label for="password">Password</label>
29 </div>
30 <div class="register">
31   <p><a href="#">Forgot Password ?</a></p>
32 </div>
33 <button>Log in</button>
34 <div class="register">
35   <p>Don't have a account ?<a href="/req/signup"> Register!!</a></p>
36 </div>
37 </form>
38 </section>
39 </body>
```

The left sidebar shows a project structure with a 'templates' folder containing index.html, login.html, and signup.html. The 'login.html' file is currently selected and highlighted with a red box around its path in the sidebar. The code for 'login.html' is displayed in the main editor area, with the line 'Don't have a account ? Register!!' also highlighted with a red box.

The screenshot shows a Java IDE interface with a dark theme. The left sidebar displays the project structure:

- Project: SecuritySignInLogin
- src:
 - Controller:
 - ContentController
 - RegistrationController
 - Model:
 - MyAppUser
 - MyAppUserRepository
 - MyAppUserService
 - Security:
 - SecurityConfig
 - DemoApplication
- resources:
 - public.error:
 - 404.html
 - static:
 - css:
 - style.css
 - js:
 - signup.js
 - templates:
 - index.html
 - login.html
 - signup.html

- application.yml

The right pane shows the content of the `signup.html` file:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Signup Form</title>
    <script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
    <script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
    <link rel="stylesheet" href="/css/style.css">
    <script src="/js/signup.js" defer></script>
</head>
<body>
    <section>
        <form action="/req/signup" method="post">
            <h1>Sign Up</h1>
            <div class="inputbox">
                <ion-icon name="person-outline"></ion-icon>
                <input type="text" id="username" name="username" required>
                <label for="username">Name</label>
            </div>
            <div class="inputbox">
                <ion-icon name="mail-outline"></ion-icon>
                <input type="email" id="email" name="email" required>
                <label for="email">Email</label>
            </div>
    </section>
</body>

```

Two specific sections of the code are highlighted with red boxes:

- A red box surrounds the `<script src="/js/signup.js" defer></script>` line.
- A red box surrounds the `<form action="/req/signup" method="post">` line.

The status bar at the bottom shows the file path: `SecuritySignInLogin > src > main > resources > templates > <> signup.html`, and the bottom right corner shows the date and time: `20:29 quinta-feira 26/09/2024`.

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbars:** SecuritySignInLogin, Version control, DemoApplication, style.css, signup.js, login.html.
- Project Structure:** Shows the project structure with packages like SecuritySignInLogin [demo], .idea, src, main, java, com.example.demo, Controller, ContentController, RegistrationController, Model, MyAppUser, MyAppUserRepository, MyAppUserService, Security, SecurityConfig, DemoApplication, resources, public.error, 404.html, static, css, style.css, and js.
- Content Controller Class:** ContentController.java is selected in the Project Structure and is displayed in the main editor window.
- Code Editor:** The ContentController.java code is shown, annotated with a red box highlighting the controller class definition and its three methods: login(), signup(), and home().

```
package com.example.demo.Controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class ContentController {

    @GetMapping("/req/login")
    public String login() { return "login"; }

    @GetMapping("/req/signup")
    public String signup() { return "signup"; }

    @GetMapping("/index")
    public String home() { return "index"; }
}
```
- Bottom Status Bar:** Shows the file path (SecuritySignInLogin > src > main > java > com > example > demo > Controller > ContentController), encoding (UTF-8), character count (7:14), and system status (LF, 4 spaces).
- Taskbar:** Shows various application icons including Windows, Task View, File Explorer, Task Manager, and several browser and utility icons.
- System Tray:** Shows the date (26/09/2024), time (20:31), and day (quinta-feira).

The screenshot shows a Java code editor in an IDE. The project structure on the left includes a .idea folder, a src folder with main, java, com.example.demo, Controller, ContentController, RegistrationController, Model, MyAppUser, MyAppUserRepository, MyAppUserService, Security, SecurityConfig, DemoApplication, resources, public.error, 404.html, static, css, and style.css. The RegistrationController.java file is open in the editor, and its code is highlighted with a red box.

```
import org.springframework.web.bind.annotation.RestController;
import com.example.demo.Model.MyAppUser;
import com.example.demo.Model.MyAppUserRepository;

@RestController
public class RegistrationController {

    @Autowired
    private MyAppUserRepository myAppUserRepository;

    @Autowired
    private PasswordEncoder passwordEncoder;

    @PostMapping(value = "/req/signup", consumes = "application/json")
    public MyAppUser createUser(@RequestBody MyAppUser user){
        user.setPassword(passwordEncoder.encode(user.getPassword()));
        return myAppUserRepository.save(user);
    }
}
```

The status bar at the bottom shows the file path: SecuritySignInLogin > src > main > java > com > example > demo > Controller > RegistrationController. It also shows the file is 28:1 LF, encoded in UTF-8 with 4 spaces, and was last modified at 20:34 on quinta-feira, 26/09/2024.

The screenshot shows a Java application structure and its corresponding code in an IDE. The project tree on the left lists files and folders under 'SecuritySignInLogin [demo]'. A red box highlights the 'MyAppUserService' file in the 'Model' package. The code editor on the right displays the implementation of the 'MyAppUserService' class, which implements the 'UserDetailsService' interface. The code uses Lombok annotations (@Service, @AllArgsConstructor) and the Spring framework (@Autowired). It retrieves a user by username from a repository and returns a UserDetails object.

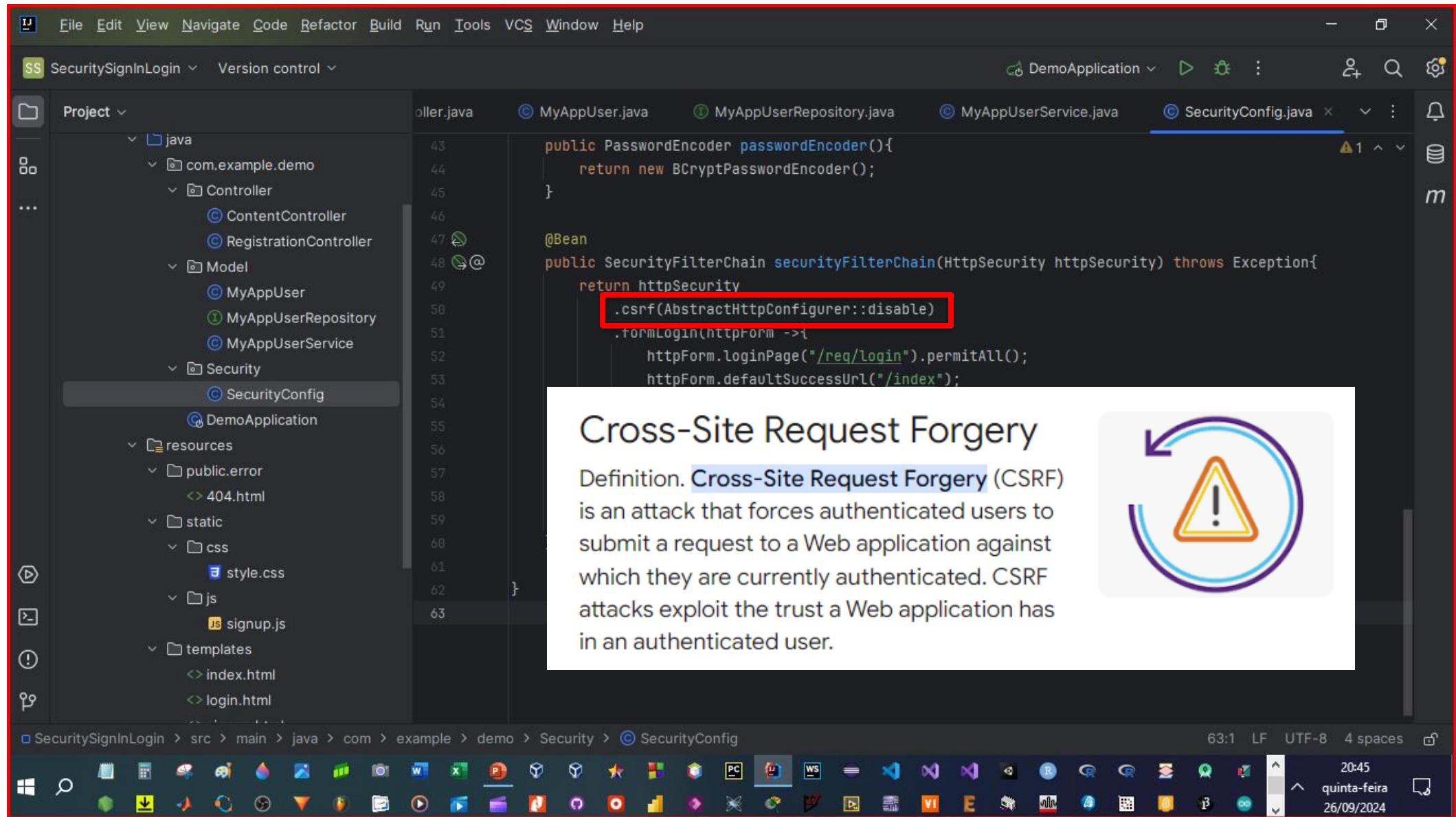
```
import lombok.AllArgsConstructor;  
  
@Service  
@AllArgsConstructor  
public class MyAppUserService implements UserDetailsService {  
  
    @Autowired  
    private MyAppUserRepository repository;  
  
    no usages  
    @Override  
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {  
  
        Optional<MyAppUser> user = repository.findByUsername(username);  
        if (user.isPresent()) {  
            var userObj = user.get();  
            return User.builder()  
                .username(userObj.getUsername())  
                .password(userObj.getPassword())  
                .build();  
        } else {  
            throw new UsernameNotFoundException(username);  
        }  
    }  
}
```

The screenshot shows a Java IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Bar:** SecuritySignInLogin, Version control.
- Toolbars:** DemoApplication, various icons for file operations.
- Code Editor:** The file `SecurityConfig.java` is open. A red box highlights the entire code block.

```
public PasswordEncoder passwordEncoder(){
    return new BCryptPasswordEncoder();
}

@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws Exception{
    return httpSecurity
        .csrf(AbstractHttpConfigurer::disable)
        .formLogin(httpForm ->{
            httpForm.loginPage("/req/login").permitAll();
            httpForm.defaultSuccessUrl("/index");
        })
        .authorizeHttpRequests(registry ->{
            registry.requestMatchers("/req/signup", "/css/**", "/js/**").permitAll();
            registry.anyRequest().authenticated();
        })
        .build();
}
```
- Project Explorer:** Shows the project structure with files like `Controller.java`, `MyAppUser.java`, `MyAppUserRepository.java`, `MyAppUserService.java`, and `SecurityConfig.java`. `SecurityConfig.java` is selected and highlighted with a red box.
- Bottom Status Bar:** SecuritySignInLogin > src > main > java > com > example > demo > Security > SecurityConfig. It also shows file encoding (63:1 LF), character set (UTF-8), and line width (4 spaces).
- Taskbar:** Shows various application icons.
- System Tray:** Shows the date (26/09/2024), time (20:45), and day (quinta-feira).

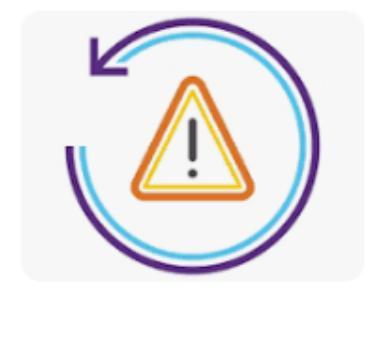


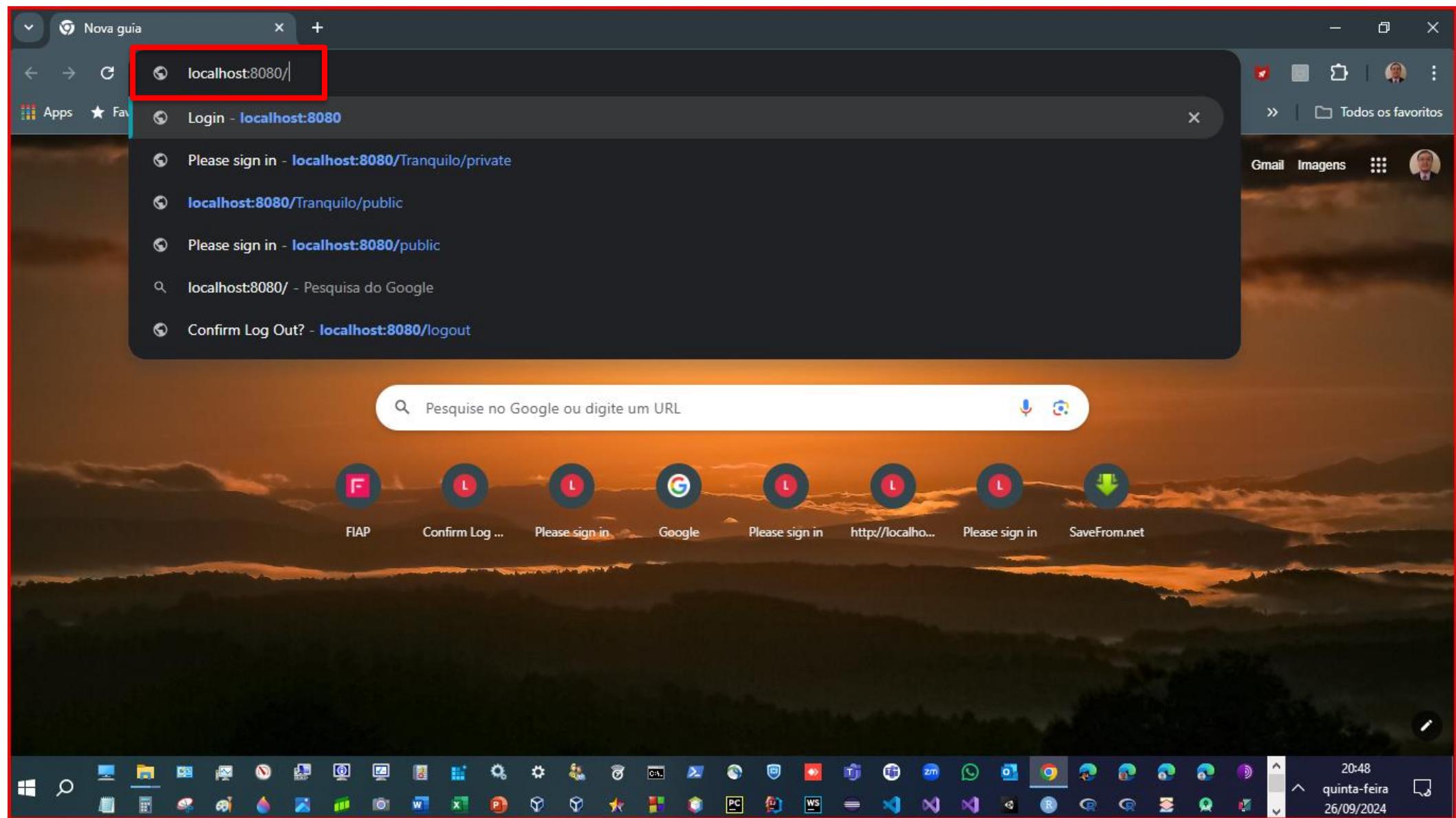
```
public PasswordEncoder passwordEncoder(){
    return new BCryptPasswordEncoder();
}

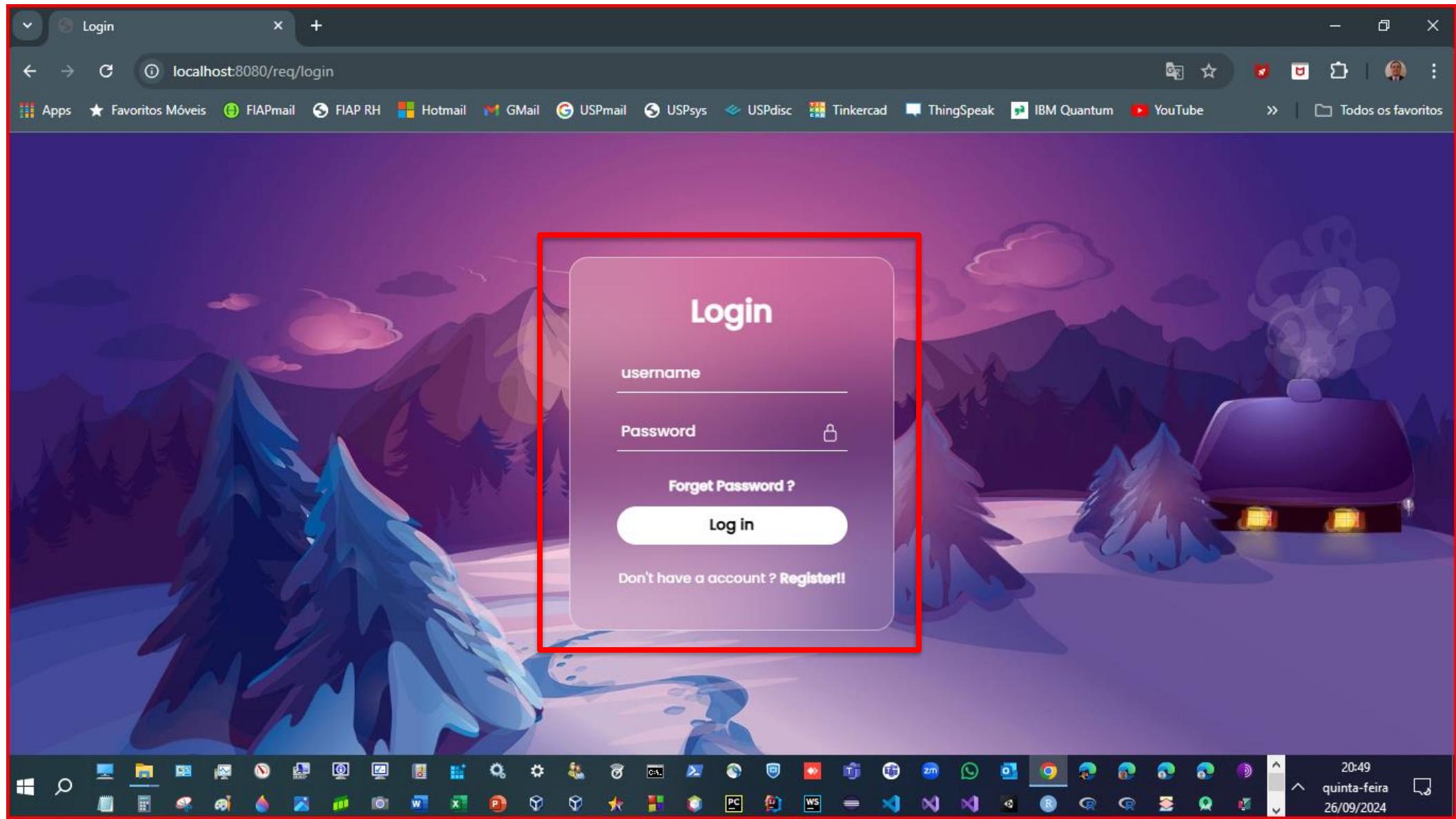
@Bean
public SecurityFilterChain securityFilterChain(HttpSecurity httpSecurity) throws Exception{
    return httpSecurity
        .csrf(AbstractHttpConfigurer::disable)
        .formLogin(httpForm ->
            httpForm.loginPage("/req/login").permitAll()
            httpForm.defaultSuccessUrl("/index");
}
```

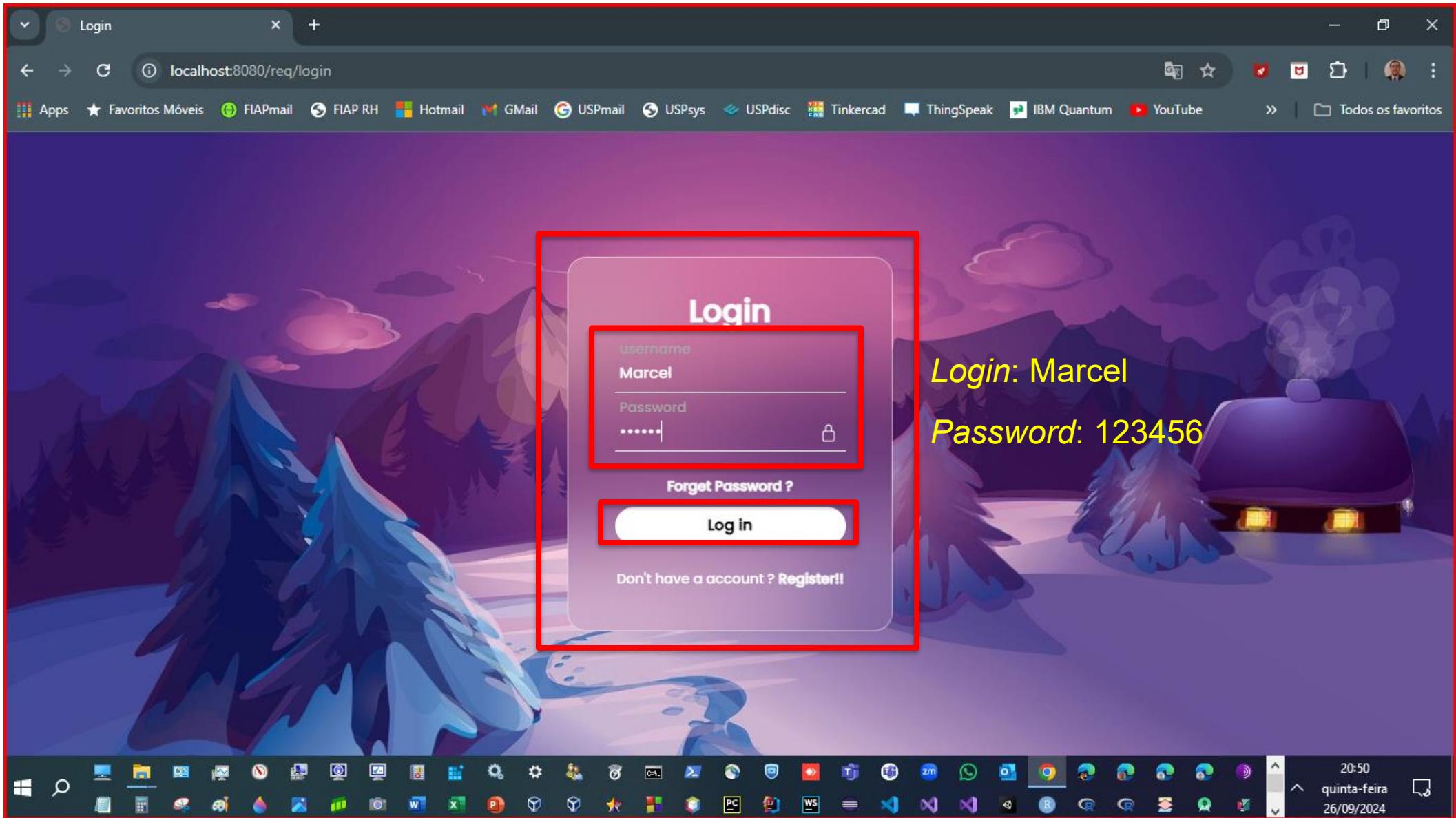
Cross-Site Request Forgery

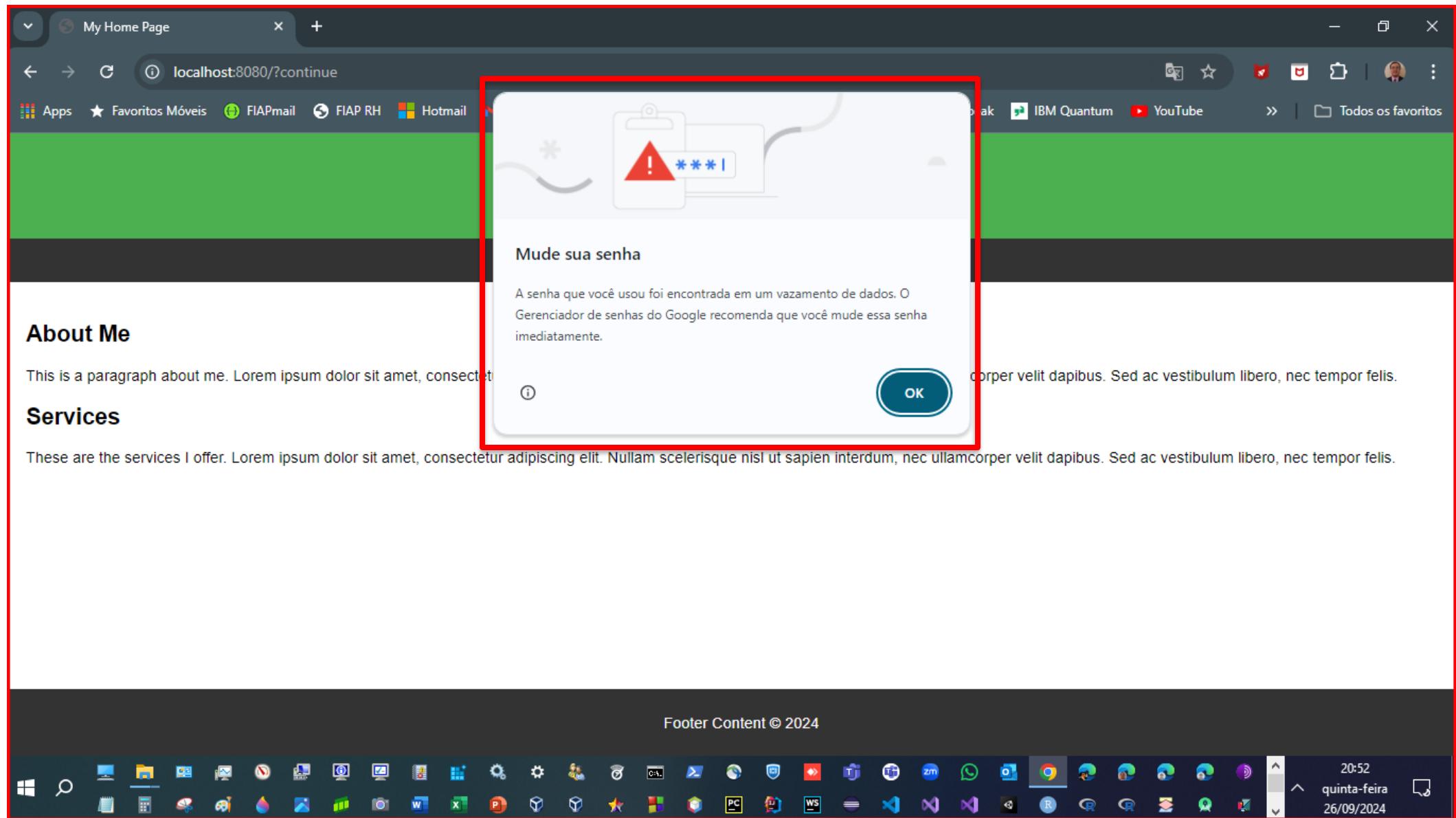
Definition. **Cross-Site Request Forgery** (CSRF) is an attack that forces authenticated users to submit a request to a Web application against which they are currently authenticated. CSRF attacks exploit the trust a Web application has in an authenticated user.











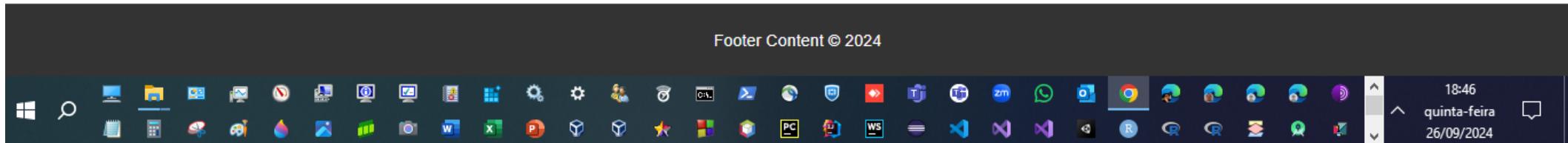
A screenshot of a web browser window titled "My Home Page". The address bar shows "localhost:8080/index#home". The page content includes a green header with the text "Welcome to My Home Page" and a black navigation bar below it with links for "Home", "About", "Services", and "Contact". The browser's toolbar and various tabs are visible at the top.

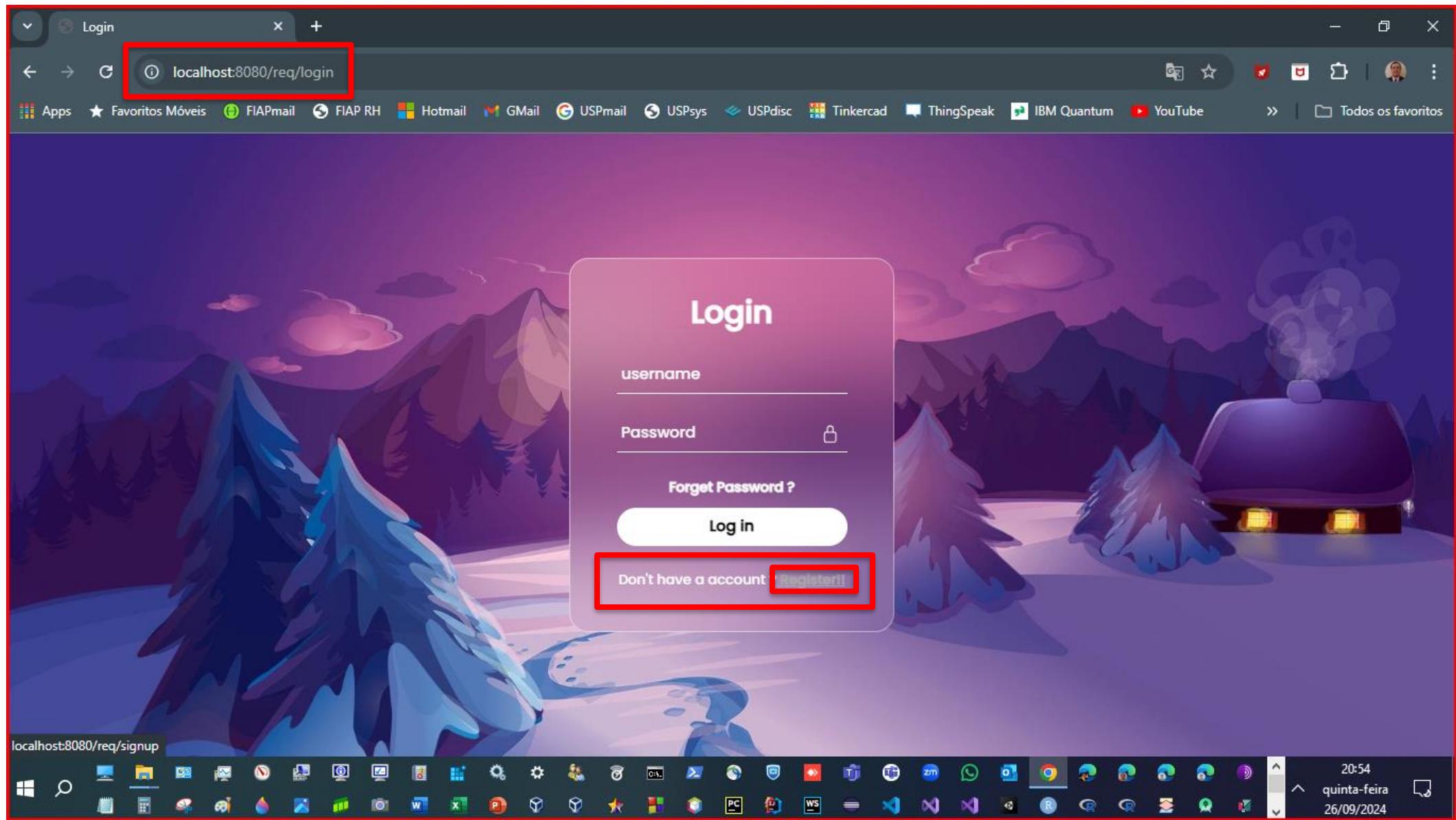
About Me

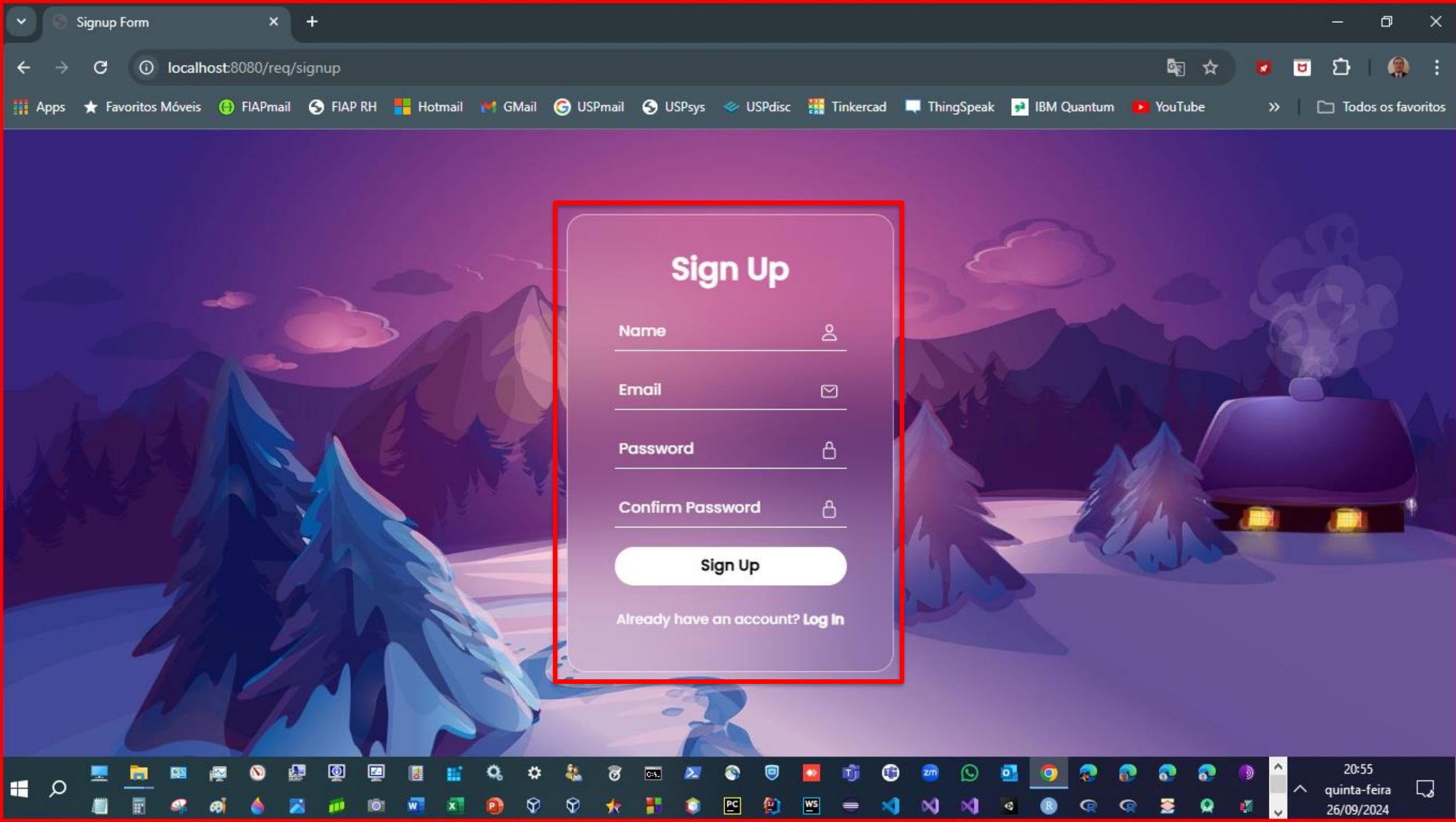
This is a paragraph about me. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam scelerisque nisl ut sapien interdum, nec ullamcorper velit dapibus. Sed ac vestibulum libero, nec tempor felis.

Services

These are the services I offer. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam scelerisque nisl ut sapien interdum, nec ullamcorper velit dapibus. Sed ac vestibulum libero, nec tempor felis.







Signup Form

localhost:8080/req/signup

Apps Favoritos Móveis FIAPmail FIAP RH Hotmail GMail USPmail USPsys USPdisc Tinkercad ThingSpeak IBM Quantum YouTube Todos os favoritos

Sign Up

Name

Email

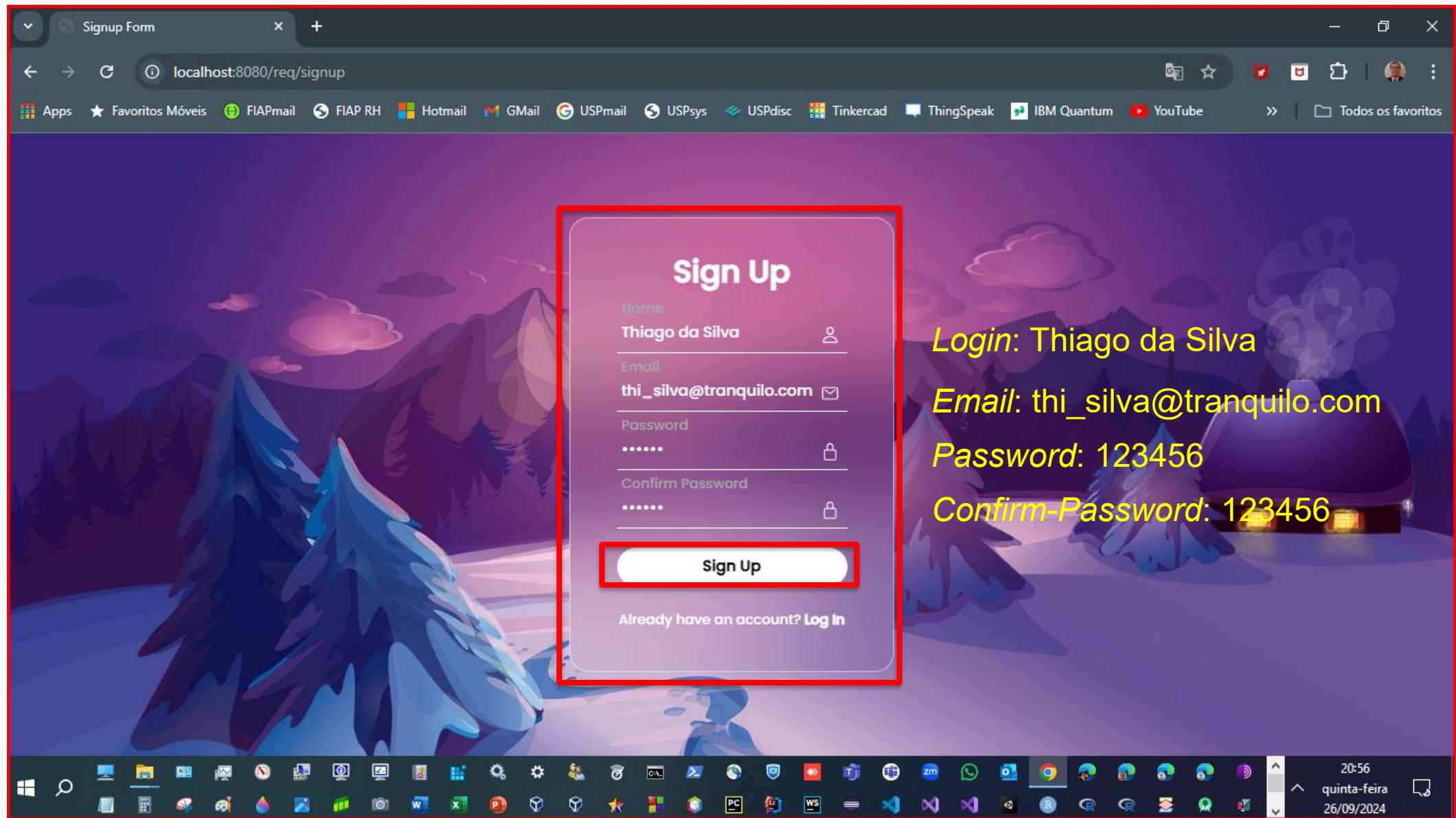
Password

Confirm Password

Sign Up

Already have an account? [Log In](#)

20:55 quinta-feira 26/09/2024



pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Properties SQL Statistics Dependencies Dependents Processes public.my_app_user/securityLogin/postgres@PostgreSQL 17

public.my_app_user/securityLogin/postgres@PostgreSQL 17

No limit

Query History Scratch Pad

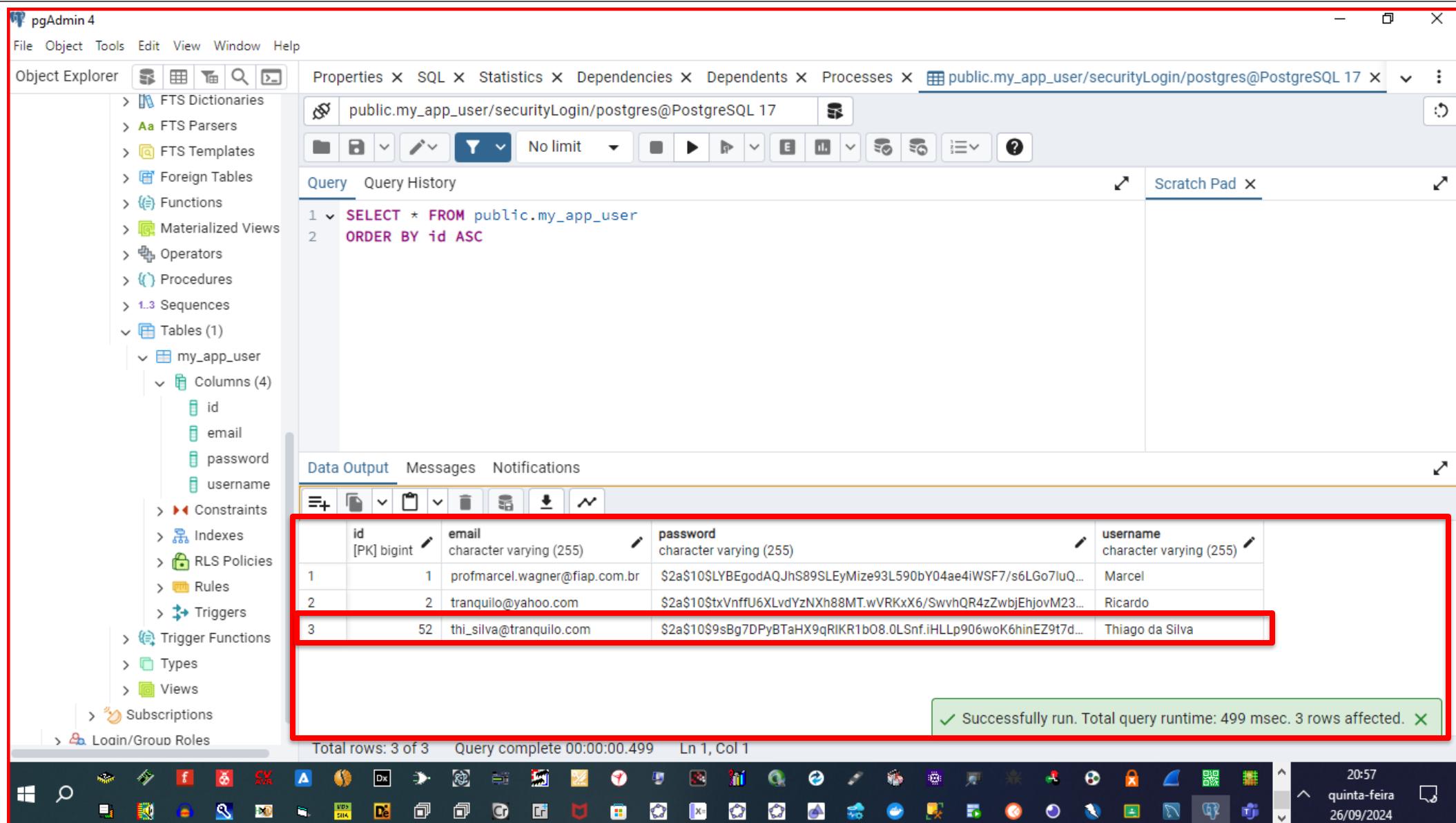
SELECT * FROM public.my_app_user
ORDER BY id ASC

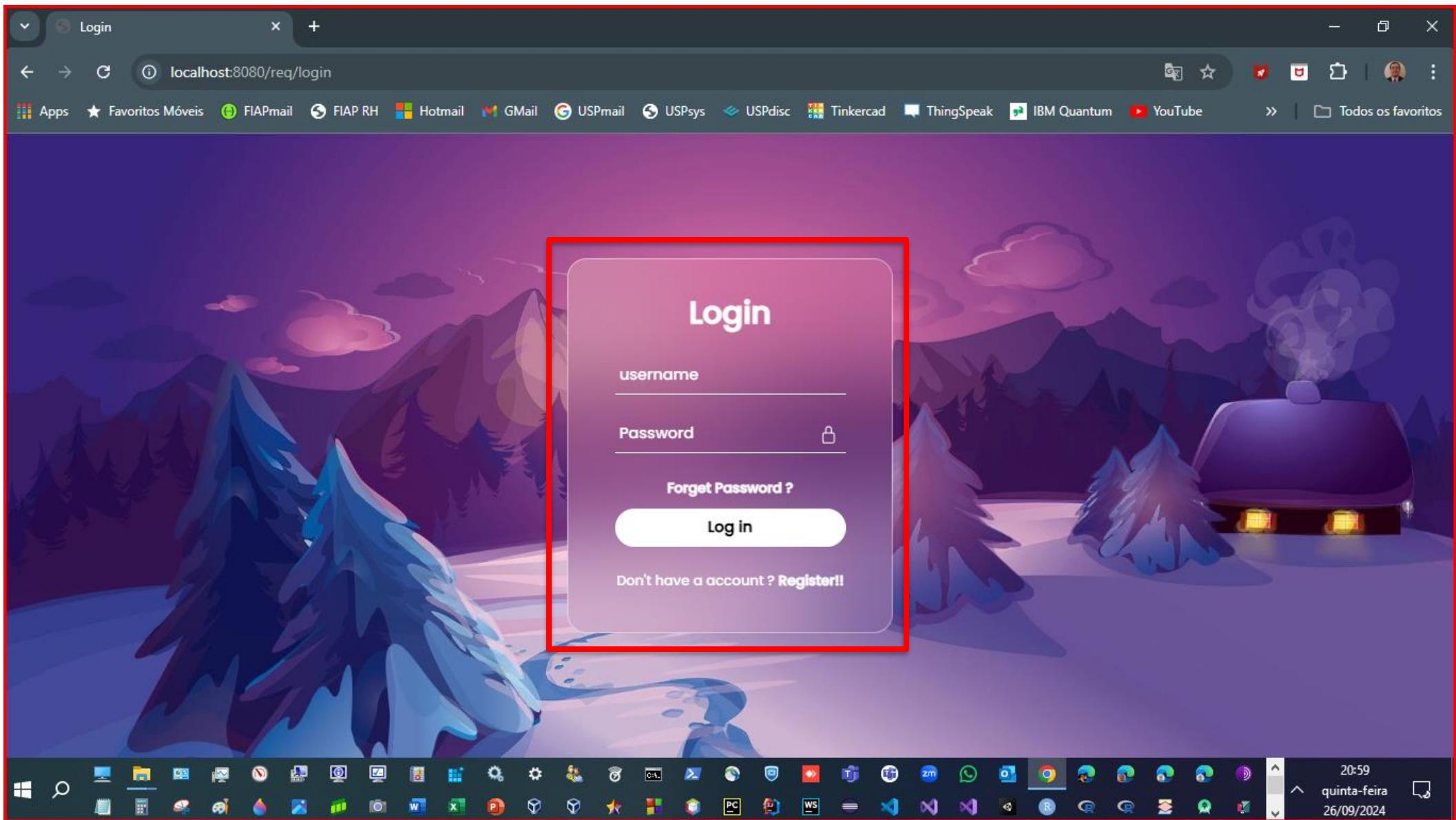
Data Output Messages Notifications

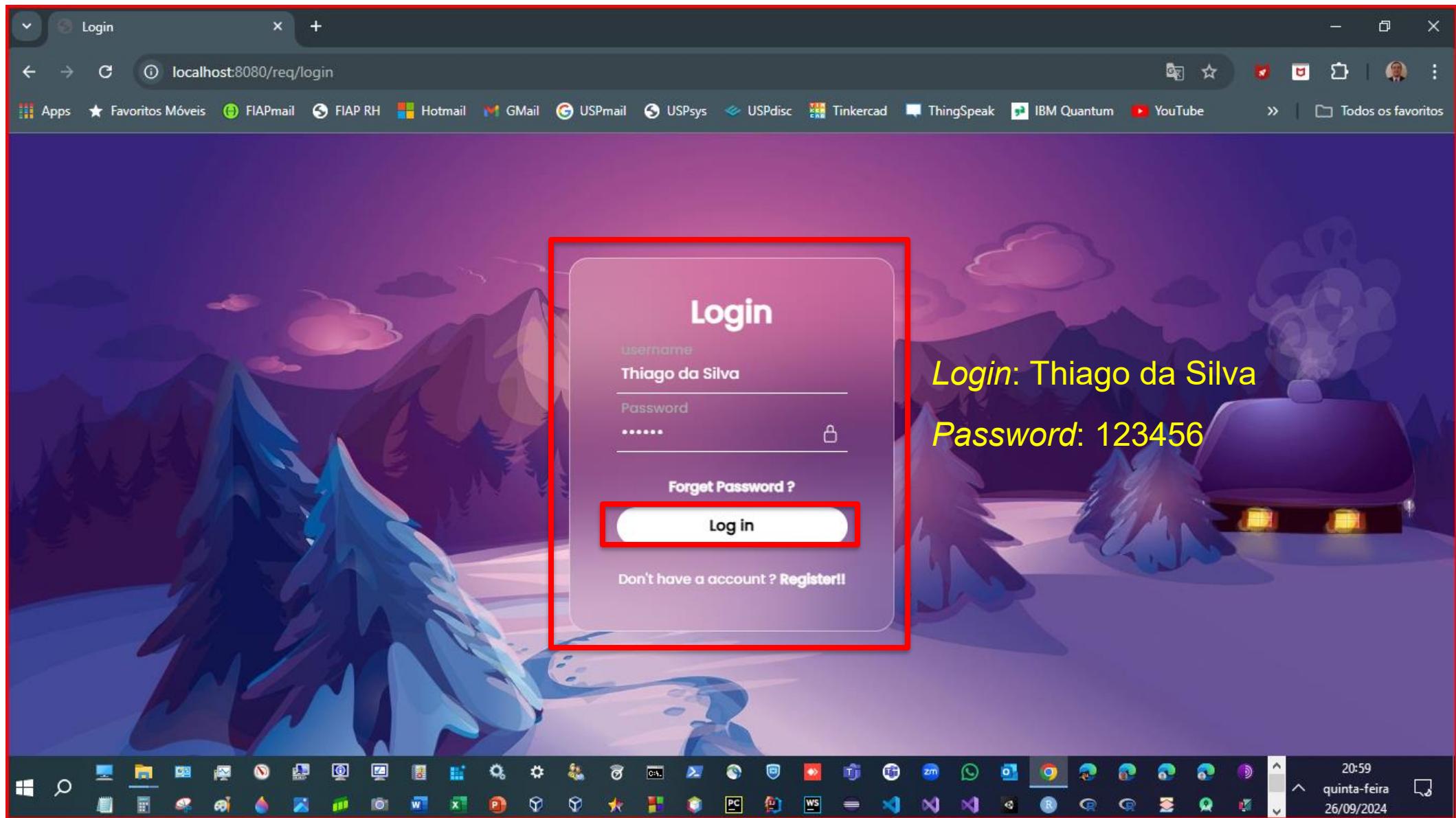
	id [PK] bigint	email character varying (255)	password character varying (255)	username character varying (255)
1	1	profmarcel.wagner@fiap.com.br	\$2a\$10\$LYBEgodAQJhs89SLEyMize93L590bY04ae4iWSF7/s6LGo7luQ4...	Marcel
2	2	tranquilo@yahoo.com	\$2a\$10\$txVnffU6XLvdYzNXh88MT.wvRKxX6/SvhQR4zZwbjEhjovM23X...	Ricardo

Total rows: 2 of 2 Query complete 00:00:00.200 Ln 1, Col 1

20:57 quinta-feira 26/09/2024







The screenshot shows a Microsoft Edge browser window with a red border around the main content area. The title bar says "My Home Page" and the address bar shows "localhost:8080/index". The page itself has a green header with the text "Welcome to My Home Page" and a black navigation bar below it with links for Home, About, Services, and Contact. The main content area contains two sections: "About Me" with placeholder text and "Services" with another placeholder text. A footer at the bottom contains the text "Footer Content © 2024" and a taskbar at the bottom with various pinned icons.

My Home Page

localhost:8080/index

Welcome to My Home Page

Home About Services Contact

About Me

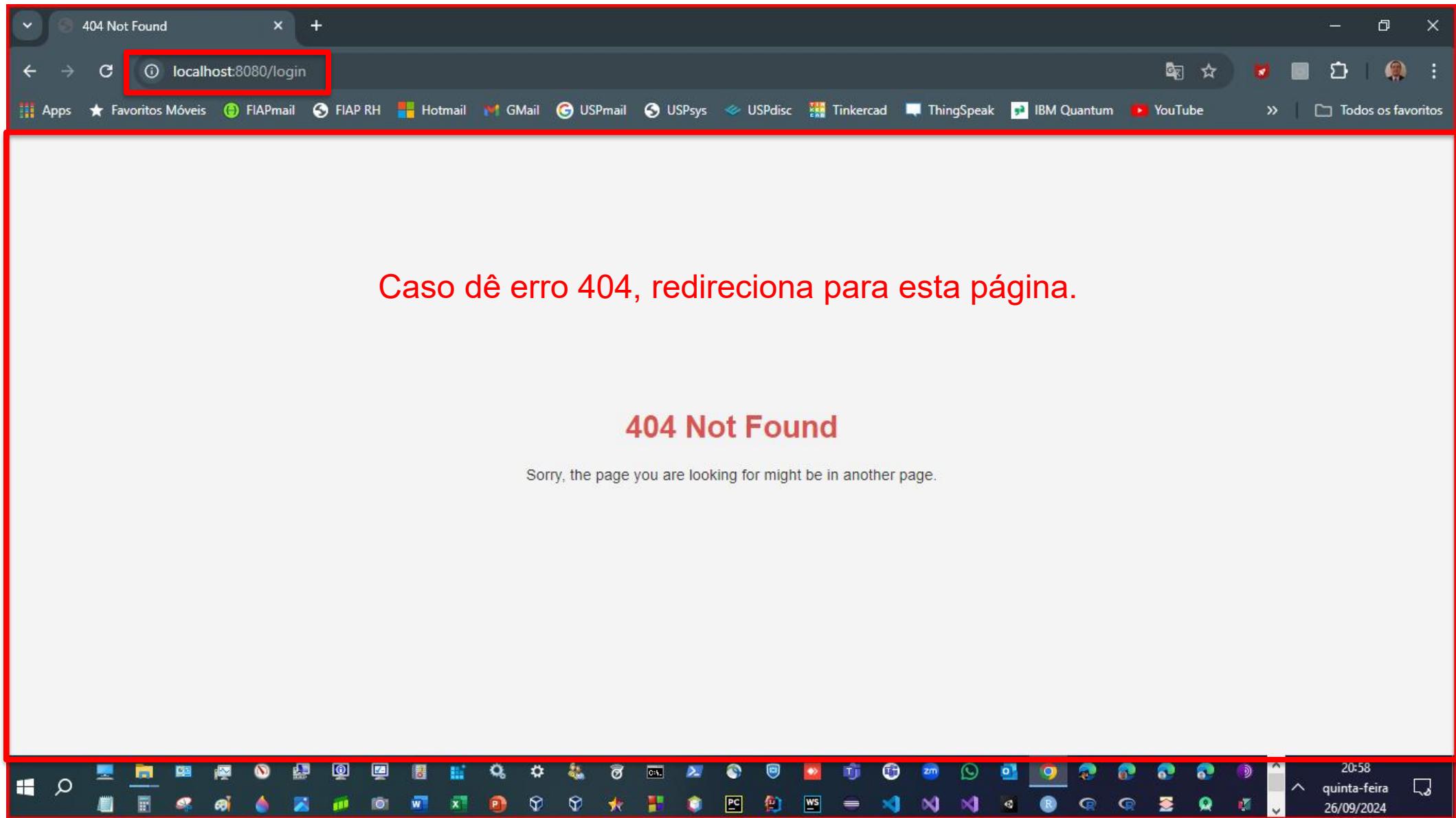
This is a paragraph about me. *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam scelerisque nisl ut sapien interdum, nec ullamcorper velit dapibus. Sed ac vestibulum libero, nec tempor felis.*

Services

These are the services I offer. *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam scelerisque nisl ut sapien interdum, nec ullamcorper velit dapibus. Sed ac vestibulum libero, nec tempor felis.*

Footer Content © 2024

21:00
quinta-feira
26/09/2024



Referências para a Apresentação

Referências

George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. **Sistemas Distribuídos: Conceitos e Projeto.** Bookman Editora, 5 edition, 2013.

Harvey M Deitel, Paul J Deitel, David R Choffnes, et al. **Sistemas Operacionais.** Pearson/Prentice Hall, 3 edition, 2005.

Maarten Van Steen and A Tanenbaum. **Sistemas Distribuídos: Princípios e Paradigmas.** Pearson/Prentice Hall, 2 edition, 2007.

Harvey M Deitel and Paul J Deitel. **Java, como programar.** Ed. Pearson/Prentice Hall, 8 edition, 2010.

StackOverflow. Disponível em: <<https://pt.stackoverflow.com/>>. Acesso em: abril de 2023.

GAMMA, Erich et al. **Elements of Reusable Object-Oriented Software.** Design Patterns, 1995.

COOPER, James William. **Java design patterns: a tutorial.** 2000.

GUERRA, Eduardo. **Design Patterns com Java: Projeto orientado a objetos guiado por padrões.** Editora Casa do Código, 2014.

Referências para a Apresentação

Referências

AWS Amazon – O que é RESTful. Disponível em: <[https://aws.amazon.com/pt/what-is/restful-api/#:~:text=Representational%20State%20Transfer%20\(REST\)%20is,complex%20network%20like%20the%20internet](https://aws.amazon.com/pt/what-is/restful-api/#:~:text=Representational%20State%20Transfer%20(REST)%20is,complex%20network%20like%20the%20internet)> . Acesso em: maio de 2023.

InfoQ. Disponível em: <<https://www.infoq.com/minibooks/emag-03-2010-rest>>. Acesso em: maio de 2024.

Michelli Brito. Disponível em: <<https://www.youtube.com/@MichelliBrito>>. Acesso em: abril de 2024.

Obeautifulcode. Disponível em: <<http://blogobeautifulcode.com/API/Learn-REST-In-18-Slides/>>. Acesso em: junho de 2022.

Berkeley. Disponível em: <<http://courses.ischool.berkeley.edu/i290-rmm/s12/slides/Lecture3%20REST.pdf>>. Acesso em: maio de 2023.

Trybe. Disponível em: <<https://blogtrybe.com/tecnologia/deploy/>>. Acesso em: julho de 2024.

Obrigado!

Contato: profmarcel.wagner@fiap.com.br

Cursos:

Tecnologia em Análise e Desenvolvimento de Sistemas (TDS)

Tecnologia em Defesa Cibernética (TDC)

Engenharia de Software (ES)

