****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

***Laborator nr.3***

***Disciplina: “Baze de date”***

**Efectuat:**

**Calancea Cătălin** MI-222

**Verificat:**

**Perebinos Mihail,** *a.u.*

**Chișinău – 20****23**

**Cuprins:**

[Tema: 5](#_Toc152173220)

[1.Proiectarea Input-Output: 5](#_Toc152173221)

[1.1 Modelul Conceptual 5](#_Toc152173222)

[1.2 Modelul Logic 6](#_Toc152173223)

[2.Proiectarea Input-Output: 7](#_Toc152173224)

[CREATE 12](#_Toc152173225)

[READ 16](#_Toc152173226)

[UPDATE 22](#_Toc152173263)

[DELETE 29](#_Toc152173264)

[3. Povestea modificata 33](#_Toc152173265)

[Concluzie 34](#_Toc152173266)

Tema: CRUD

# 1.Proiectarea Input-Output:

## 1.1 Modelul Conceptual

## 1.2 Modelul Logic



# 2.Proiectarea Input-Output:

* Crearea bazei de date

-- Creare bazei de date

CREATE DATABASE Lab3;

USE Lab3;

-- Creare tabela scenario

CREATE TABLE scenario (

scenario\_id INT PRIMARY KEY AUTO\_INCREMENT,

scenario\_name VARCHAR(255) NOT NULL

);

-- Creare tabela characters

CREATE TABLE characters (

character\_id INT PRIMARY KEY AUTO\_INCREMENT,

main\_character VARCHAR(255) NOT NULL,

second\_character VARCHAR(255) NOT NULL,

negative\_main\_character VARCHAR(255) NOT NULL,

negative\_second\_character VARCHAR(255) NOT NULL,

scenario\_id INT,

FOREIGN KEY (scenario\_id) REFERENCES scenario(scenario\_id)

);

-- Creare tabela circumstance

CREATE TABLE circumstance (

circumstance\_id INT PRIMARY KEY AUTO\_INCREMENT,

circumstance VARCHAR(255) NOT NULL,

season VARCHAR(255) NOT NULL,

scenario\_id INT,

FOREIGN KEY (scenario\_id) REFERENCES scenario(scenario\_id)

);

-- Creare tabela clothes

CREATE TABLE clothes (

clothes\_id INT PRIMARY KEY AUTO\_INCREMENT,

clothes\_for\_main\_character VARCHAR(255) NOT NULL,

scenario\_id INT,

FOREIGN KEY (scenario\_id) REFERENCES scenario(scenario\_id)

);

-- Creare tabela characteristics

CREATE TABLE characteristics (

characteristics\_id INT PRIMARY KEY AUTO\_INCREMENT,

characteristics\_for\_main\_character VARCHAR(255) NOT NULL,

characteristics\_for\_second\_character VARCHAR(255) NOT NULL,

scenario\_id INT,

FOREIGN KEY (scenario\_id) REFERENCES scenario(scenario\_id)

);

-- Creare tabela transport

CREATE TABLE transport (

transport\_id INT PRIMARY KEY AUTO\_INCREMENT,

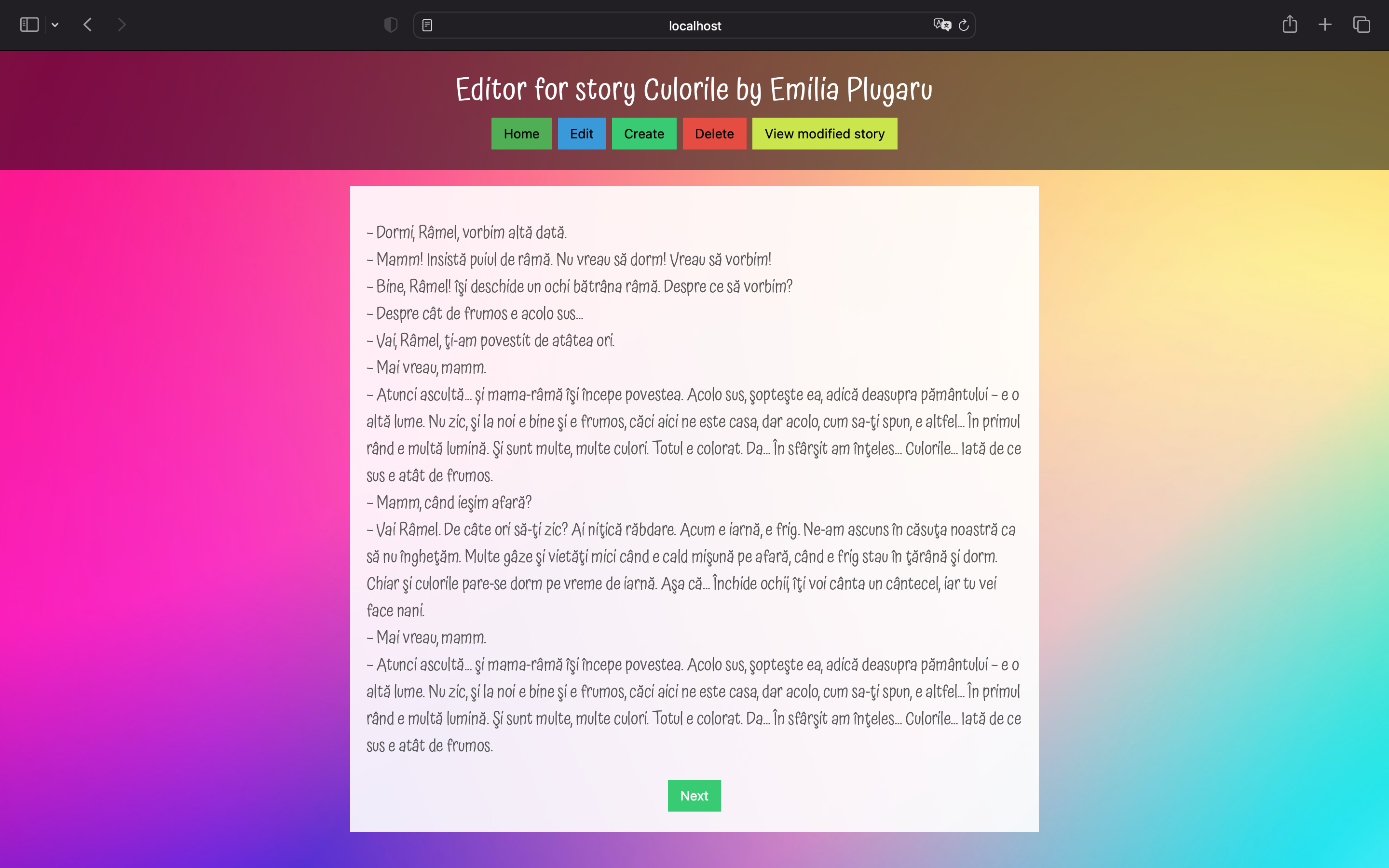
transport\_type VARCHAR(255) NOT NULL,

scenario\_id INT,

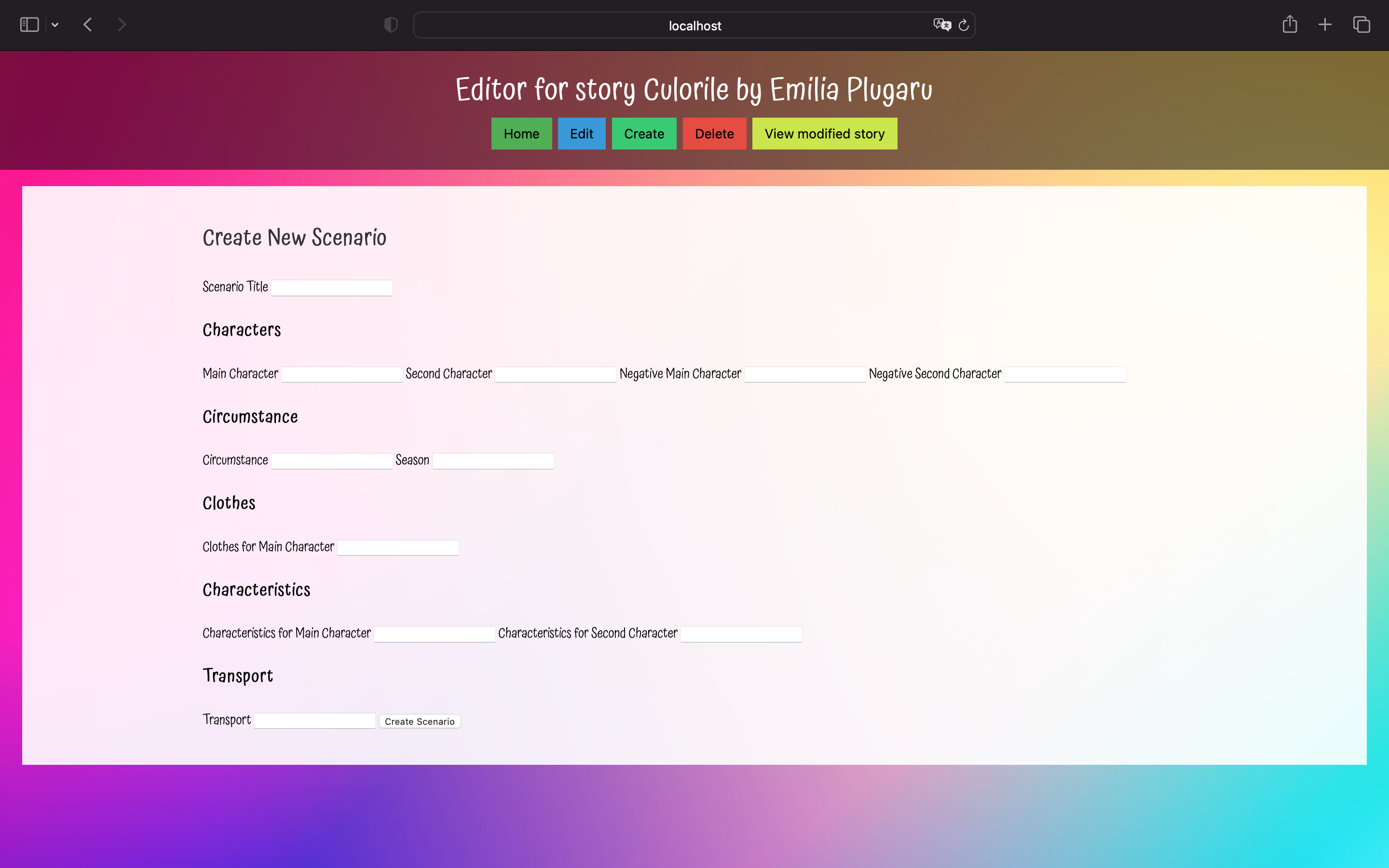
FOREIGN KEY (scenario\_id) REFERENCES scenario(scenario\_id)

);

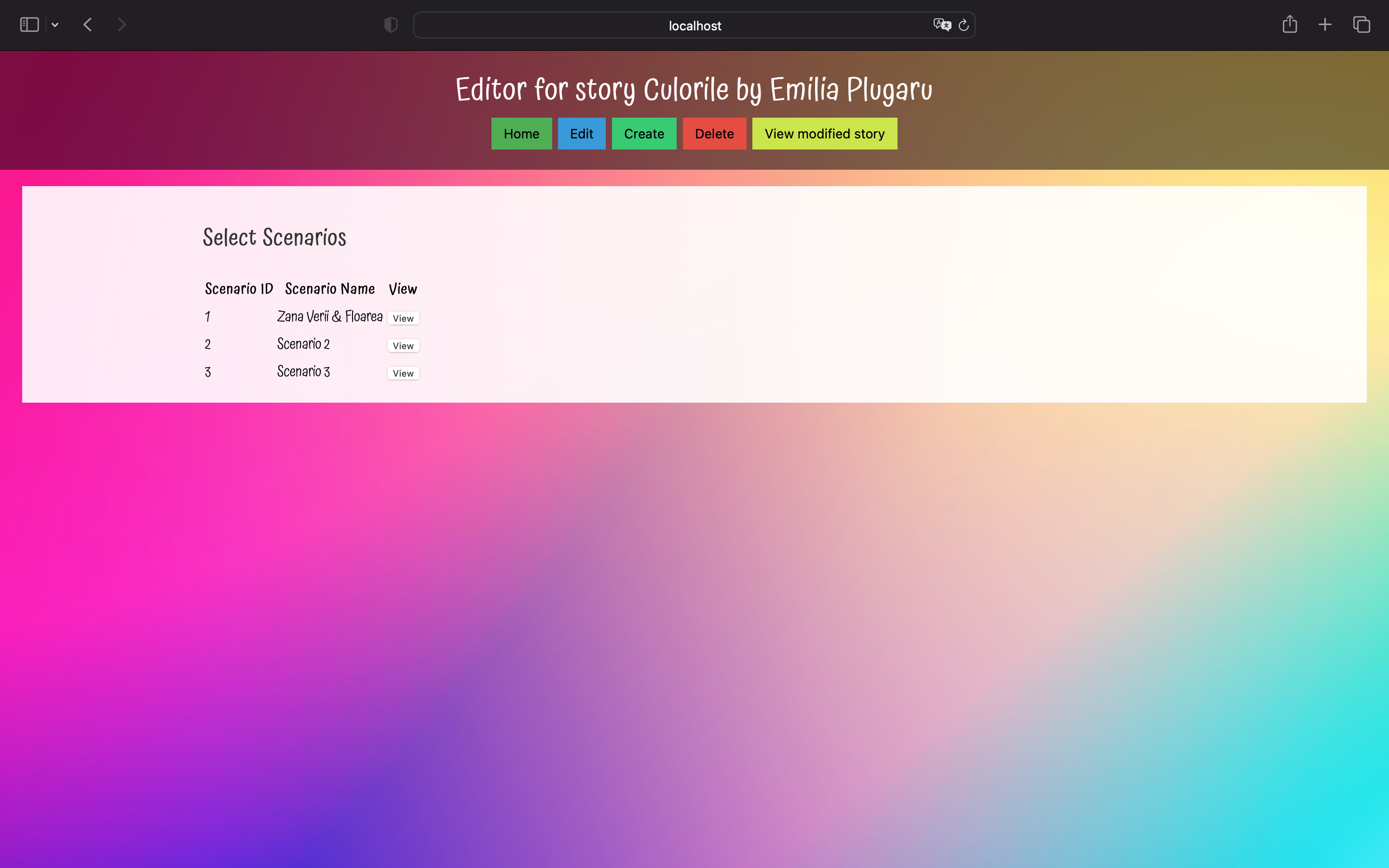


Pe pagina principala , avem povestea initiala , nemodificata.  


Dupa care avem forma de creare a scenariului :



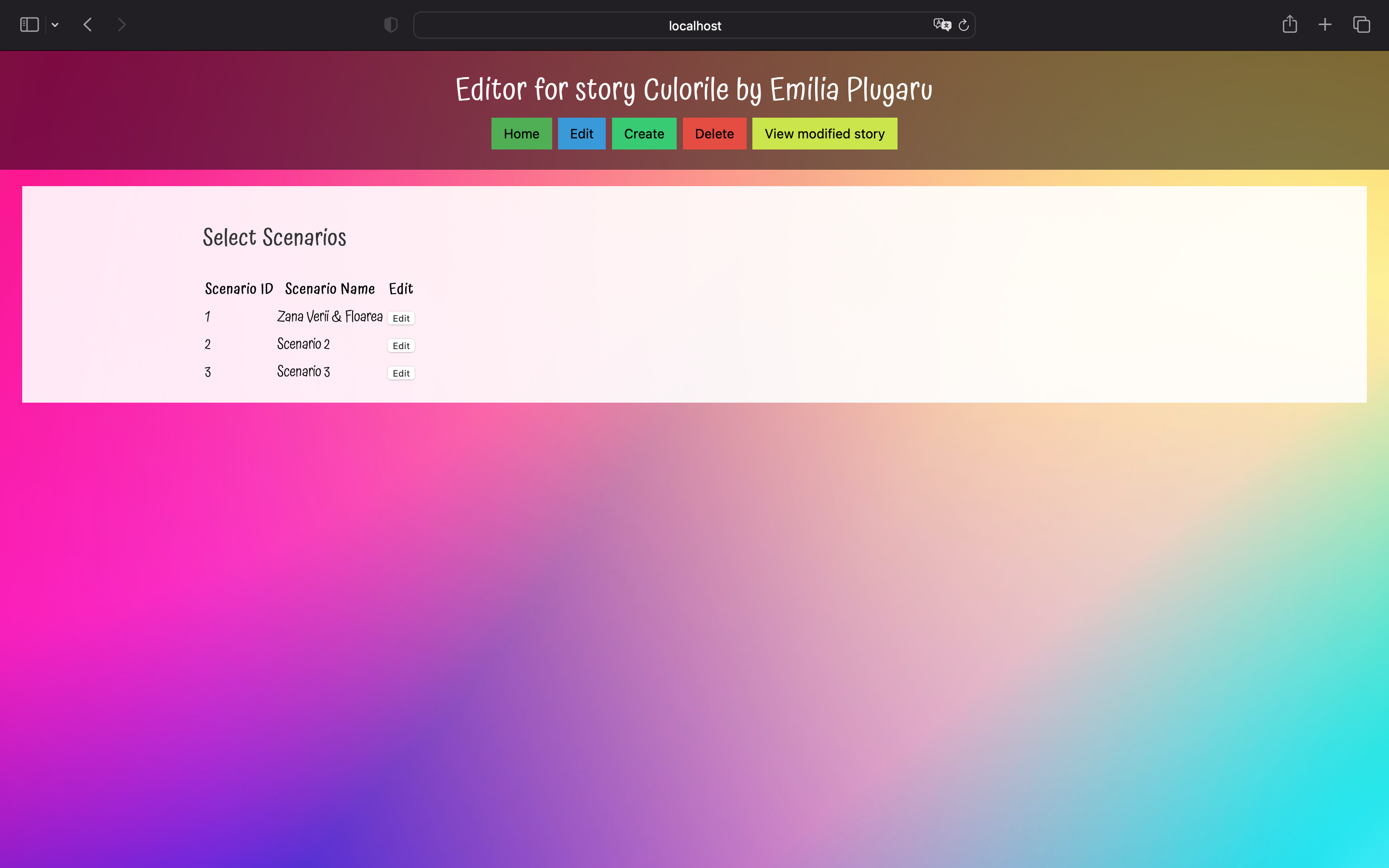
Lista cu scenariile create



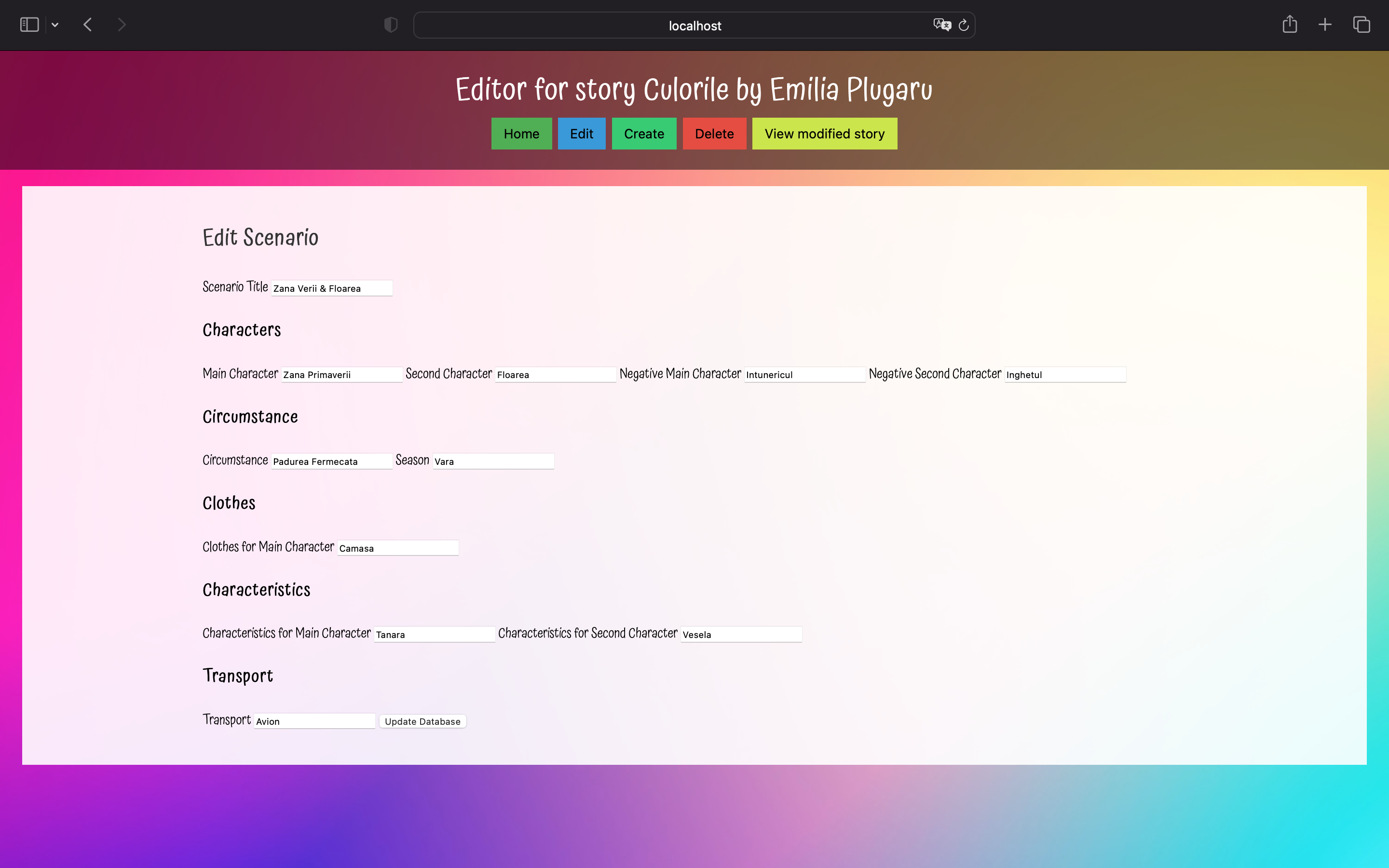
Povestea modificata dupa scenariul creat de utilizator , unde am evidentiat textul modificat



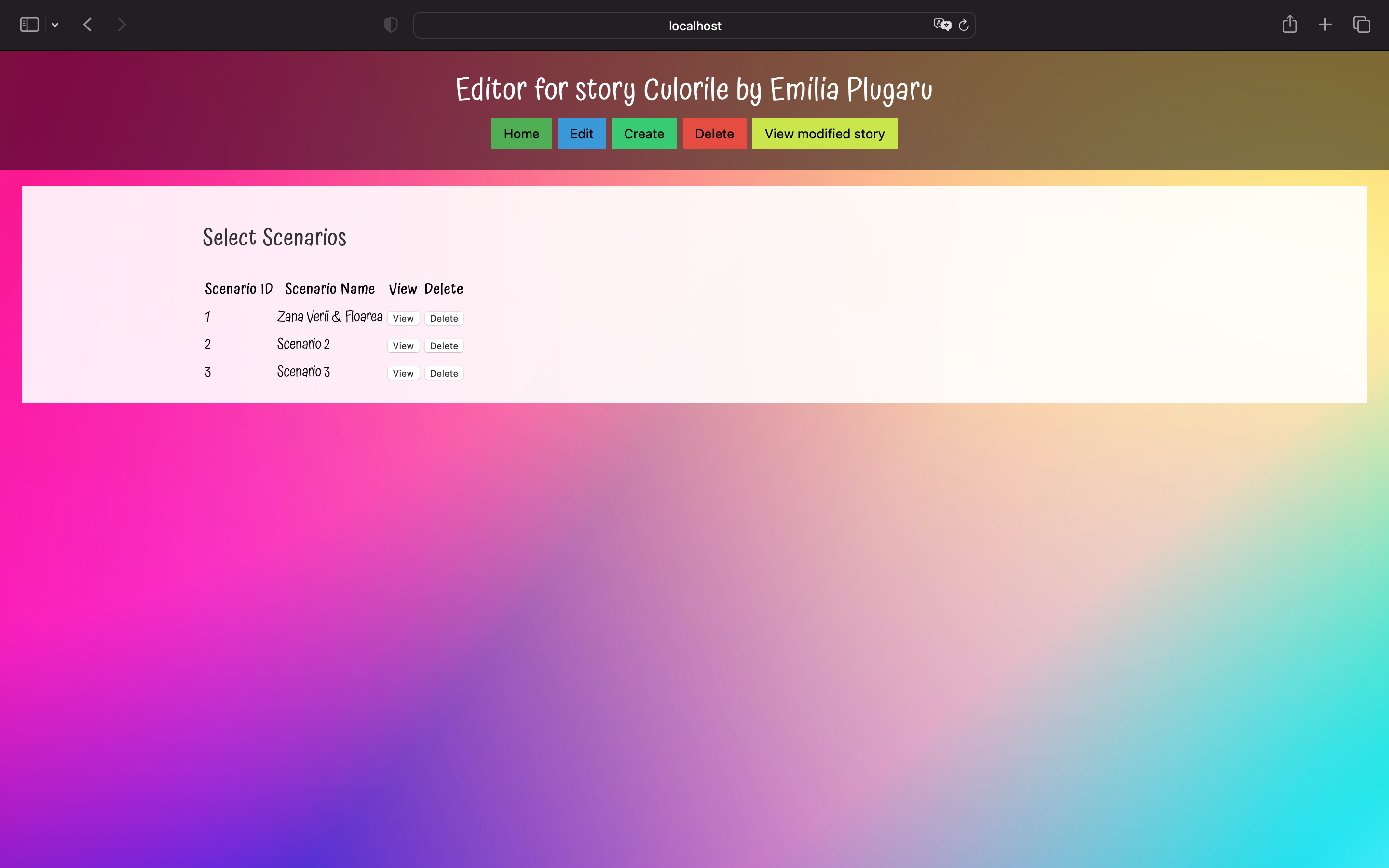
Lista cu scenariile pentru modificare



Forma de editare a scenariului



Lista cu scenarii pentru a fi stearse din baza de date



# 

# CREATE

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8"/>

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

<link

*href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap"

*rel*="stylesheet"

/>

<link *rel*="stylesheet" *href*="styles.css"/>

<title>Create a new scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<?php

include 'connect.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$scenario\_name = $\_POST['field11'];

$insert\_scenario\_query = "INSERT INTO scenario (scenario\_name) VALUES ('$scenario\_name')";

$conn->query($insert\_scenario\_query);

$new\_scenario\_id = $conn->insert\_id;

$main\_character = $\_POST['field1'];

$second\_character = $\_POST['field2'];

$negative\_main\_character = $\_POST['field3'];

$negative\_second\_character = $\_POST['field4'];

$insert\_characters\_query = "INSERT INTO characters (scenario\_id, main\_character, second\_character, negative\_main\_character, negative\_second\_character) VALUES ($new\_scenario\_id, '$main\_character', '$second\_character', '$negative\_main\_character', '$negative\_second\_character')";

$conn->query($insert\_characters\_query);

$circumstance = $\_POST['field5'];

$season = $\_POST['field6'];

$insert\_circumstance\_query = "INSERT INTO circumstance (scenario\_id, circumstance, season) VALUES ($new\_scenario\_id, '$circumstance', '$season')";

$conn->query($insert\_circumstance\_query);

$clothes = $\_POST['field7'];

$insert\_clothes\_query = "INSERT INTO clothes (scenario\_id, clothes\_for\_main\_character) VALUES ($new\_scenario\_id, '$clothes')";

$conn->query($insert\_clothes\_query);

$characteristics\_main = $\_POST['field8'];

$characteristics\_second = $\_POST['field9'];

$insert\_characteristics\_query = "INSERT INTO characteristics (scenario\_id, characteristics\_for\_main\_character, characteristics\_for\_second\_character) VALUES ($new\_scenario\_id, '$characteristics\_main', '$characteristics\_second')";

$conn->query($insert\_characteristics\_query);

$transport = $\_POST['field10'];

$insert\_transport\_query = "INSERT INTO transport (scenario\_id, transport\_type) VALUES ($new\_scenario\_id, '$transport')";

$conn->query($insert\_transport\_query);

header("Location: ".$\_SERVER['PHP\_SELF']);

exit();

}

?>

<section *id*="formSection">

<h2>Create New Scenario</h2>

<form *id*="title1Form" *method*="post">

<label *for*="field11">Scenario Title</label>

<input *type*="text" *id*="field11" *name*="field11" *value*="" *required*/>

<h3>Characters</h3>

<label *for*="field1">Main Character</label>

<input *type*="text" *id*="field1" *name*="field1" *required*/>

<label *for*="field2">Second Character</label>

<input *type*="text" *id*="field2" *name*="field2" *required*/>

<label *for*="field3">Negative Main Character</label>

<input *type*="text" *id*="field3" *name*="field3" *required*/>

<label *for*="field4">Negative Second Character</label>

<input *type*="text" *id*="field4" *name*="field4" *required*/>

<h3>Circumstance</h3>

<label *for*="field5">Circumstance</label>

<input *type*="text" *id*="field5" *name*="field5" *required*/>

<label *for*="field6">Season</label>

<input *type*="text" *id*="field6" *name*="field6" *required*/>

<h3>Clothes</h3>

<label *for*="field7">Clothes for Main Character</label>

<input *type*="text" *id*="field7" *name*="field7" *required*/>

<h3>Characteristics</h3>

<label *for*="field8">Characteristics for Main Character</label>

<input *type*="text" *id*="field8" *name*="field8" *required*/>

<label *for*="field9">Characteristics for Second Character</label>

<input *type*="text" *id*="field9" *name*="field9" *required*/>

<h3>Transport</h3>

<label *for*="field10">Transport</label>

<input *type*="text" *id*="field10" *name*="field10" *required*/>

<button *type*="submit">Create Scenario</button>

</form>

</section>

</main>

</body>

</html>

Acest script PHP este responsabil pentru gestionarea creării unui scenariu într-o aplicație web. Vom examina pașii principali din script:

1. Conectarea la baza de date:

**include 'connect.php';**

Acesta include un fișier (`connect.php`) care probabil conține codul pentru conectarea la baza de date.

2. Verificarea metodei HTTP:

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

Scriptul verifică dacă cererea HTTP este de tip POST.

3. Procesarea datelor primite prin POST:

$scenario\_name = $\_POST['field11'];

$main\_character = $\_POST['field1'];

// ... (alți parametri)

Scriptul preia datele trimise prin metoda POST din formularul HTML și le salvează în variabile.

4. Inserarea datelor în baza de date:

$insert\_scenario\_query = "INSERT INTO scenario (scenario\_name) VALUES ('$scenario\_name')";

$conn->query($insert\_scenario\_query);

// ... (inserarea în alte tabele)

Utilizând datele primite, scriptul construiește și execută interogări SQL pentru a introduce informațiile în baza de date.

5. Redirecționarea utilizatorului:

header("Location: ".$\_SERVER['PHP\_SELF']);

exit();

După ce datele au fost introduse cu succes în baza de date, utilizatorul este redirecționat înapoi pe aceeași pagină pentru a împiedica trimiterea repetată a datelor prin reîmprospătarea paginii.

6. Formular HTML:

Scriptul include un formular HTML cu diferite câmpuri pentru introducerea datelor. Când utilizatorul completează și trimite formularul, datele sunt procesate de scriptul PHP.

În ansamblu, acest script PHP funcționează împreună cu un formular HTML pentru a permite utilizatorilor să creeze scenarii într-o aplicație web. Datele introduse sunt stocate într-o bază de date, iar utilizatorii sunt redirecționați înapoi pe pagina curentă după finalizarea procesului.

# READ

modifed\_story.PHP

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap" *rel*="stylesheet" />

<link *rel*="stylesheet" *href*="styles.css" />

<title>List of existing scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<section *id*="formSection">

<h2>Select Scenarios</h2>

<table *id*="scenariosTable">

<thead>

<tr>

<th>Scenario ID</th>

<th>Scenario Name</th>

<th>View</th>

</tr>

</thead>

<tbody>

<?php include 'connect.php'; ?>

<?php

$sql = "SELECT scenario\_id, scenario\_name FROM scenario";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "<tr>

<td>{$row['scenario\_id']}</td>

<td>{$row['scenario\_name']}</td>

<td><a href='modified\_story\_choice.php?id={$row['scenario\_id']}'><button>View</button></a></td>

</tr>";

}

} else {

echo "<tr><td colspan='3'>No scenarios found</td></tr>";

}

?>

</tbody>

</table>

</section>

</main>

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

<script *src*="story.js"></script>

</body>

</html>

modified\_story\_choice.PHP

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap" *rel*="stylesheet" />

<link *rel*="stylesheet" *href*="styles.css" />

<title>Story with modified scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="story-content">

<?php

include 'connect.php';

*// Assuming scenario\_id is passed through the URL*

$scenario\_id = isset($\_GET['id']) ? $\_GET['id'] : null;

if ($scenario\_id) {

*// Fetch data from the characters table*

$characters\_query = "SELECT FROM characters WHERE scenario\_id = $scenario\_id";

$characters\_result = $conn->query($characters\_query);

$characteristics\_query = "SELECT FROM characteristics WHERE scenario\_id = $scenario\_id";

$characteristics\_result = $conn->query($characteristics\_query);

$circumstance\_query = "SELECT FROM circumstance WHERE scenario\_id = $scenario\_id";

$circumstance\_result = $conn->query($circumstance\_query);

$clothes\_query = "SELECT FROM clothes WHERE scenario\_id = $scenario\_id";

$clothes\_result = $conn->query($clothes\_query);

$transport\_query = "SELECT FROM transport WHERE scenario\_id = $scenario\_id";

$transport\_result = $conn->query($transport\_query);

if ($characters\_result->num\_rows > 0) {

$characters\_data = $characters\_result->fetch\_assoc();

$characteristics\_data = $characteristics\_result -> fetch\_assoc();

$circumstance\_data = $circumstance\_result -> fetch\_assoc();

$clothes\_data = $clothes\_result -> fetch\_assoc();

$transport\_data = $transport\_result -> fetch\_assoc();

$variables = array(

'second\_character',

'main\_character',

'negative\_main\_character',

'negative\_second\_character',

'characteristics\_pers\_principal',

'characteristics\_pers\_secundar',

'circumstance\_circumstance',

'circumstance\_season',

'clothes\_clothes',

'transport\_transport'

);

*// Replace placeholders with data*

$second\_character = $characters\_data['second\_character'];

$main\_character = $characters\_data['main\_character'];

$negative\_main\_character = $characters\_data['negative\_main\_character'];

$negative\_second\_character = $characters\_data['negative\_second\_character'];

$characteristics\_pers\_principal = $characteristics\_data['characteristics\_for\_main\_character'];

$characteristics\_pers\_secundar = $characteristics\_data['characteristics\_for\_second\_character'];

$circumstance\_circumstance = $circumstance\_data['circumstance'];

$circumstance\_season = $circumstance\_data['season'];

$clothes\_clothes = $clothes\_data['clothes\_for\_main\_character'];

$transport\_transport = $transport\_data['transport\_type'];

foreach ($variables as $variable) {

$$variable = "<span style='background-color: yellow;'>{$$variable}</span>";

}

echo "<div id='part1' class='story-part'>

<p>– Dormi, $second\_character, vorbim altă dată. <br>

– $main\_character! Insistă $second\_character. Nu vreau să dorm! Vreau să vorbim!<br>

– Bine, $second\_character! îşi deschide un ochi $characteristics\_pers\_principal $main\_character. Despre ce să vorbim?<br>

– Despre cât de frumos e acolo sus…<br>

– Vai, $second\_character, ţi-am povestit de atâtea ori.<br>

– Mai vreau, $main\_character.<br>

– Atunci ascultă… și $main\_character îşi începe povestea. Acolo $circumstance\_circumstance, şopteşte $main\_character, adică deasupra pământului – e o altă lume. Nu zic, şi la noi e bine şi e frumos, căci aici ne este casa, dar acolo, cum sa-ţi spun, e altfel… În primul rând e multă lumină. Şi sunt multe, multe culori. Totul e colorat. Da… În sfârşit am înţeles… Culorile… Iată de ce sus e atât de frumos.<br>

– $main\_character, când ieşim afară?<br>

– Vai $second\_character. De câte ori să-ţi zic? Ai niţică răbdare. Acum e $circumstance\_season. Ne-am ascuns în căsuţa noastră. Multe gâze şi vietăţi mici când e cald mişună pe afară, când e frig se imbraca in $clothes\_clothes. Chiar şi culorile pare-se dorm pe vreme de $circumstance\_season. Aşa că… Închide ochii, îţi voi cânta un cântecel, iar tu vei face nani.<br>

– Mai vreau, $main\_character.<br>

– Atunci ascultă… şi $characteristics\_pers\_principal - $main\_character îşi începe povestea. Acolo sus, şopteşte $main\_character, adică deasupra $circumstance\_circumstance – e o altă lume. Nu zic, şi la noi e bine şi e frumos, căci aici ne este casa, dar acolo, cum sa-ţi spun, e altfel… În primul rând e multă lumină. Şi sunt multe, multe culori. Totul e colorat. Da… În sfârşit am înţeles… Culorile… Iată de ce sus e atât de frumos.<br>

</p>

</div>";

echo "<div id='part2' class='story-part'>

<p>– Nu vreau nani! Nu-mi place aici!<br>

Oricât de $characteristics\_pers\_principal ar fi $main\_character, până la urmă îşi iese din fire:<br>

– Încetează! strigă $main\_character. Vom merge sus cu $transport\_transport doar când ni se va da de ştire că e posibil! Gata! Nici o vorbă mai mult! Dormi!<br>

– $main\_character, începe iarăşi $second\_character. Spune-mi… Cine ne va da de ştire?<br>

– Înţelegi scumpule, deasupra $circumstance\_circumstance e $negative\_main\_character. $negative\_main\_character sunt diferiţi… Adică de diferite culori, de diferite mărimi. Când e $circumstance\_season şi cerul se acoperă cu $negative\_main\_character negri, $negative\_main\_character cel mai important îşi ia toba şi începe să bată: Bah! Bah! Bah! Bah! Toba lui se aude până la noi. Deci vorbim după ce auzim bubuitul. Acum dormi!<br>

– {$main\_character}! Auzi!? sare {$second\_character} din pătucul său. {$negative\_main\_character} cel mai important bate din tobă!

– Da? nu-i vine să creadă {$main\_character} O fi venit {$circumstance\_season}?<br>

– A venit, {$main\_character}! A venit! Hai sa mergem afară! Repede! Vreau să văd culorile!<br>

– Nu, {$second\_character}. Încă nu…<br>

– De ce?! M-ai minţit!? Atunci… plec singur, fără tine!<br>

</p>

</div>";

echo "<div id='part3' class='story-part'>

<p>– Stai, {$second\_character}! Aşteaptă! Doar un pic. Mai întâi de toate ţărâna trebuie să devină reavănă, puhavă, ca să putem săpa tunelul prin care ne vom strecura. Nu durează mult. Acuş picăturile calde de ploaie se vor scurge în sol… Ah! E atât de plăcut când faci prima scăldătoare de {$circumstance\_season} Atât de plăcut… Vezi? Solul începe să devină umed. Deci… putem să pornim săpatul.<br>

Cele două stau într-o grădină plină de flori, plină de culori, {$characteristics\_pers\_secundar} {$second\_character} nu încetează să se minuneze de frumuseţea din jur.<br>

– {$main\_character}, zice {$second\_character}, nu mă mai întorc acasă. Îmi place atât de mult aici. Mai ales îmi plac florile… sunt foarte frumoase!<br>

– Eh, oftează {$characteristics\_pers\_principal} {$main\_character}, prea multe primejdii ne înconjoară… $negative\_second\_character, iată răul cel mare. $negative\_second\_character!<br>

– {$main\_character}! Priveşte! Ce fiinţă ciudată aleargă spre noi?! E nespus de nostimă! Iar în urma ei – o mulţime de ghemuleţe aurii! Mă duc să-i salut! strigă {$second\_character}, apoi porneşte cu îndrăzneală în întâmpinarea $negative\_second\_character.<br>

– Vai, {$second\_character}, ce faci!? e disperată {$main\_character} cea {$characteristics\_pers\_principal}, reuşind totuşi să-şi împingă odrasla înapoi în tunel. Am scăpat, răsuflă {$main\_character} uşurată. Să nu te fi oprit la timp… Dar… să nu mai vorbim. Principalul e că ai văzut… culorile. Cred că altădată nu mai riscăm. Era cât pe ce să ne înghită o $negative\_second\_character<br>

E mult de lucru {$circumstance\_circumstance}. {$second\_character} şi {$main\_character} sunt convinşi că şi datorită lor din solul pufos răsar în fiecare {$circumstance\_season} culorile.</

</p>

</div>";

} else {

echo "<p>No data found for the provided scenario ID in the characters table.</p>";

}

} else {

echo "<p>No scenario ID provided.</p>";

}

?>

<div *class*="navigation-buttons">

<button *id*="prevButton" *onclick*="toggleStoryPart('prev')" *style*="display:none">Previous</button>

<button *id*="nextButton" *onclick*="toggleStoryPart('next')">Next</button>

</div>

</main>

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

</body>

</html>

# Acest script PHP și pagina HTML asociată sunt concepute pentru a afișa un scenariu modificat în funcție de un scenariu selectat anterior. Aici sunt pașii principali:

# modifed\_story.php:

# 1. Conectarea la baza de date:

# <?php include 'connect.php'; ?>

# Acesta include un fișier (`connect.php`) care conține codul pentru conectarea la baza de date.

# 2. Interogarea bazei de date pentru a obține datele scenariului:

# $scenario\_id = isset($\_GET['id']) ? $\_GET['id'] : null;

# // Interogări pentru a obține date din diferite tabele

# $characters\_query = "SELECT FROM characters WHERE scenario\_id = $scenario\_id";

# // ... (alte interogări)

# 3. Prezentarea datelor în pagină:

# if ($characters\_result->num\_rows > 0) {

# // Extrage datele și le afișează în HTML

# // Atribuie valori variabilelor pentru a fi utilizate în șabloanele din poveste

# foreach ($variables as $variable) {

# $$variable = "<span style='background-color: yellow;'>{$$variable}</span>";

# }

# // Afișează părți ale poveștii în funcție de datele obținute

# echo "<div id='part1' class='story-part'>

# <p>... părți ale poveștii ...</p>

# </div>";

# // ... (alte părți ale poveștii)

# echo "<div id='part3' class='story-part'>

# <p>... părți ale poveștii ...</p>

# </div>";

# } else {

# echo "<p>No data found for the provided scenario ID in the characters table.</p>";

# }

# 4. Butoane de navigare între părțile poveștii:

# <div class="navigation-buttons">

# <button id="prevButton" onclick="toggleStoryPart('prev')" style="display:none">Previous</button>

# <button id="nextButton" onclick="toggleStoryPart('next')">Next</button>

# </div>

# - Pagina `modified\_story\_choice.php?id={$row['scenario\_id']}` este legată de butonul "View" în lista de scenarii (`/modifed\_story.php`).

# - Variabilele definite în partea PHP a codului sunt utilizate pentru a înlocui anumite cuvinte sau părți ale poveștii cu datele din baza de date. Acest lucru se realizează cu scopul de a evidenția anumite elemente ale poveștii.

# - Scriptul Javascript `script.js` este inclus pentru a furniza funcționalitatea de navigare între părțile poveștii.

# 

# UPDATE

edit.PHP

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap" *rel*="stylesheet" />

<link *rel*="stylesheet" *href*="styles.css" />

<title>Edit existing scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<section *id*="formSection">

<h2>Select Scenarios</h2>

<table *id*="scenariosTable">

<thead>

<tr>

<th>Scenario ID</th>

<th>Scenario Name</th>

<th>Edit</th>

</tr>

</thead>

<tbody>

<?php include 'connect.php'; ?>

<?php

$sql = "SELECT scenario\_id, scenario\_name FROM scenario";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "<tr>

<td>{$row['scenario\_id']}</td>

<td>{$row['scenario\_name']}</td>

<td><a href='edit\_choice.php?id={$row['scenario\_id']}'><button>Edit</button></a></td>

</tr>";

}

} else {

echo "<tr><td colspan='3'>No scenarios found</td></tr>";

}

?>

</tbody>

</table>

</section>

</main>

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

<script *src*="story.js"></script>

</body>

</html>

edit\_choice.PHP

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8" />

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

<link

*href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap"

*rel*="stylesheet"

/>

<link *rel*="stylesheet" *href*="styles.css" />

<title>CRUD APP</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<?php

include 'connect.php';

$scenario\_id = isset($\_GET['id']) ? $\_GET['id'] : null;

if ($scenario\_id) {

$scenario\_query = "SELECT FROM scenario WHERE scenario\_id = $scenario\_id";

$scenario\_result = $conn->query($scenario\_query);

$characters\_query = "SELECT FROM characters WHERE scenario\_id = $scenario\_id";

$characters\_result = $conn->query($characters\_query);

$characteristics\_query = "SELECT FROM characteristics WHERE scenario\_id = $scenario\_id";

$characteristics\_result = $conn->query($characteristics\_query);

$circumstance\_query = "SELECT FROM circumstance WHERE scenario\_id = $scenario\_id";

$circumstance\_result = $conn->query($circumstance\_query);

$clothes\_query = "SELECT FROM clothes WHERE scenario\_id = $scenario\_id";

$clothes\_result = $conn->query($clothes\_query);

$transport\_query = "SELECT FROM transport WHERE scenario\_id = $scenario\_id";

$transport\_result = $conn->query($transport\_query);

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$scenario\_name = $\_POST['field11'];

$update\_scenario\_query = "UPDATE scenario SET scenario\_name = '$scenario\_name' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_scenario\_query);

$main\_character = $\_POST['field1'];

$second\_character = $\_POST['field2'];

$negative\_main\_character = $\_POST['field3'];

$negative\_second\_character = $\_POST['field4'];

$update\_characters\_query = "UPDATE characters SET main\_character = '$main\_character', second\_character = '$second\_character', negative\_main\_character = '$negative\_main\_character', negative\_second\_character = '$negative\_second\_character' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_characters\_query);

$circumstance = $\_POST['field5'];

$season = $\_POST['field6'];

$update\_circumstance\_query = "UPDATE circumstance SET circumstance = '$circumstance', season = '$season' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_circumstance\_query);

*// Example: Update clothes table*

$clothes = $\_POST['field7'];

$update\_clothes\_query = "UPDATE clothes SET clothes\_for\_main\_character = '$clothes' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_clothes\_query);

*// Example: Update characteristics table*

$characteristics\_main = $\_POST['field8'];

$characteristics\_second = $\_POST['field9'];

$update\_characteristics\_query = "UPDATE characteristics SET characteristics\_for\_main\_character = '$characteristics\_main', characteristics\_for\_second\_character = '$characteristics\_second' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_characteristics\_query);

*// Example: Update transport table*

$transport = $\_POST['field10'];

$update\_transport\_query = "UPDATE transport SET transport\_type = '$transport' WHERE scenario\_id = $scenario\_id";

$conn->query($update\_transport\_query);

*// Redirect to the same page to avoid form resubmission*

header("Location: ".$\_SERVER['PHP\_SELF']."?id=$scenario\_id");

exit();

}

if ($characters\_result->num\_rows > 0) {

$scenario\_data = $scenario\_result->fetch\_assoc();

$characters\_data = $characters\_result->fetch\_assoc();

$characteristics\_data = $characteristics\_result -> fetch\_assoc();

$circumstance\_data = $circumstance\_result -> fetch\_assoc();

$clothes\_data = $clothes\_result -> fetch\_assoc();

$transport\_data = $transport\_result -> fetch\_assoc();

*// Replace placeholders with data*

$scenario\_name = $scenario\_data['scenario\_name'];

$second\_character = $characters\_data['second\_character'];

$main\_character = $characters\_data['main\_character'];

$negative\_main\_character = $characters\_data['negative\_main\_character'];

$negative\_second\_character = $characters\_data['negative\_second\_character'];

$characteristics\_pers\_principal = $characteristics\_data['characteristics\_for\_main\_character'];

$characteristics\_pers\_secundar = $characteristics\_data['characteristics\_for\_second\_character'];

$circumstance\_circumstance = $circumstance\_data['circumstance'];

$circumstance\_season = $circumstance\_data['season'];

$clothes\_clothes = $clothes\_data['clothes\_for\_main\_character'];

$transport\_transport = $transport\_data['transport\_type'];

}} ?>

<section *id*="formSection">

<h2>Edit Scenario</h2>

<form *id*="title1Form" *method*="post">

<label *for*="field11">Scenario Title</label>

<input *type*="text" *id*="field11" *name*="field11" *value*="<?php echo $scenario\_name; ?>" *required* />

*<!-- Example: Table 1 -->*

<h3>Characters</h3>

<label *for*="field1">Main Character</label>

<input *type*="text" *id*="field1" *name*="field1" *value*="<?php echo $main\_character; ?>" *required* />

<label *for*="field2">Second Character</label>

<input *type*="text" *id*="field2" *name*="field2" *value*="<?php echo $second\_character; ?>" *required* />

<label *for*="field3">Negative Main Character</label>

<input *type*="text" *id*="field3" *name*="field3" *value*="<?php echo $negative\_main\_character; ?>" *required* />

<label *for*="field4">Negative Second Character</label>

<input *type*="text" *id*="field4" *name*="field4" *value*="<?php echo $negative\_second\_character; ?>" *required* />

*<!-- Example: Table 2 -->*

<h3>Circumstance</h3>

<label *for*="field5">Circumstance</label>

<input *type*="text" *id*="field5" *name*="field5" *value*="<?php echo $circumstance\_circumstance; ?>" *required* />

<label *for*="field6">Season</label>

<input *type*="text" *id*="field6" *name*="field6" *value*="<?php echo $circumstance\_season; ?>" *required* />

*<!-- Example: Table 3 -->*

<form *id*="table3Form" *method*="post">

<h3>Clothes</h3>

<label *for*="field7">Clothes for Main Character</label>

<input *type*="text" *id*="field7" *name*="field7" *value*="<?php echo $clothes\_clothes; ?>" *required* />

*<!-- Example: Table 4 -->*

<h3>Characteristics</h3>

<label *for*="field8">Characteristics for Main Character</label>

<input *type*="text" *id*="field8" *name*="field8" *value*="<?php echo $characteristics\_pers\_principal; ?>" *required* />

<label *for*="field9">Characteristics for Second Character</label>

<input *type*="text" *id*="field9" *name*="field9" *value*="<?php echo $characteristics\_pers\_secundar; ?>" *required* />

*<!-- Example: Table 5 -->*

<h3>Transport</h3>

<label *for*="field10">Transport</label>

<input *type*="text" *id*="field10" *name*="field10" *value*="<?php echo $transport\_transport; ?>" *required* />

<button *type*="submit">Update Database</button>

</form>

</section>

</main>

</body>

</html>

1. Conectarea la baza de date:

<?php include 'connect.php'; ?>

2. Interogarea bazei de date pentru a obține datele scenariului:

$scenario\_id = isset($\_GET['id']) ? $\_GET['id'] : null;

// Interogări pentru a obține date din diferite tabele

$scenario\_query = "SELECT FROM scenario WHERE scenario\_id = $scenario\_id";

$characters\_query = "SELECT FROM characters WHERE scenario\_id = $scenario\_id";

// ... (alte interogări pentru celelalte tabele)

3. Actualizarea bazei de date în urma unei cereri POST:

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Obțineți valorile actualizate din formular și efectuați actualizările

// ... (actualizări pentru fiecare tabel)

// Redirectați pentru a evita retrimiterile formularului

header("Location: ".$\_SERVER['PHP\_SELF']."?id=$scenario\_id");

exit();

}

4. Prezentarea datelor în formular:

if ($characters\_result->num\_rows > 0) {

// Extrageți datele și utilizați-le pentru a completa valorile inițiale ale formularului

// ... (prezentarea datelor în câmpurile de input)

}

5. Formularul pentru editare:

<form id="title1Form" method="post">

<!-- Câmp pentru titlul scenariului -->

<label for="field11">Scenario Title</label>

<input type="text" id="field11" name="field11" value="<?php echo $scenario\_name; ?>" required />

<!-- ... (alte câmpuri pentru celelalte tabele) -->

<!-- Butoane de actualizare și închidere a formularului -->

<button type="submit">Update Database</button>

</form>

6. Structura HTML a paginii:

- Există două secțiuni principale, o secțiune pentru afișarea scenariilor și o secțiune pentru formularul de editare.

- Sunt prezentate tabele cu scenarii pentru a alege unul pentru editare.

- Există un formular pentru editarea detaliilor scenariului selectat.

# DELETE

delete.php

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap" *rel*="stylesheet" />

<link *rel*="stylesheet" *href*="styles.css" />

<title>List of existing scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<section *id*="formSection">

<h2>Select Scenarios</h2>

<table *id*="scenariosTable">

<thead>

<tr>

<th>Scenario ID</th>

<th>Scenario Name</th>

<th>View</th>

<th>Delete</th>

</tr>

</thead>

<tbody>

<?php include 'connect.php'; ?>

<?php

$sql = "SELECT scenario\_id, scenario\_name FROM scenario";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "<tr>

<td>{$row['scenario\_id']}</td>

<td>{$row['scenario\_name']}</td>

<td><a href='edit\_choice.php?id={$row['scenario\_id']}'><button>View</button></a></td>

<td><a href='delete\_choice.php?id={$row['scenario\_id']}'><button>Delete</button></a></td>

</tr>";

}

} else {

echo "<tr><td colspan='3'>No scenarios found</td></tr>";

}

?>

</tbody>

</table>

</section>

</main>

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

<script *src*="story.js"></script>

</body>

</html>

delete\_choice.php

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

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

<link *href*="https://fonts.googleapis.com/css2?family=Atma:wght@300&display=swap" *rel*="stylesheet" />

<link *rel*="stylesheet" *href*="styles.css" />

<title>Delete Scenario</title>

</head>

<body>

<header>

<h1>Editor for story Culorile by Emilia Plugaru</h1>

<div *class*="action-buttons">

<a *href*="/index.html"><button *id*="homeButton">Home</button></a>

<a *href*="/edit.php"><button *id*="editButton">Edit</button></a>

<a *href*="/create.php"> <button *id*="createButton">Create</button></a>

<a *href*="/delete.php"> <button *id*="deleteButton">Delete</button></a>

<a *href*="/modifed\_story.php"><button *id*="viewButton">View modified story</button></a>

</div>

</header>

<main *class*="form-content">

<section *id*="formSection">

<h2>Delete Scenario</h2>

<?php

include 'connect.php';

if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET['id'])) {

$scenario\_id = $\_GET['id'];

*// Delete related data from other tables (characters, circumstance, clothes, characteristics, transport)*

$delete\_characters\_query = "DELETE FROM characters WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_characters\_query);

$delete\_circumstance\_query = "DELETE FROM circumstance WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_circumstance\_query);

$delete\_clothes\_query = "DELETE FROM clothes WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_clothes\_query);

$delete\_characteristics\_query = "DELETE FROM characteristics WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_characteristics\_query);

$delete\_transport\_query = "DELETE FROM transport WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_transport\_query);

*// Delete scenario from scenario table*

$delete\_scenario\_query = "DELETE FROM scenario WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_scenario\_query);

echo "<p>Scenario and related data deleted successfully.</p>";

} else {

echo "<p>Error: Scenario ID not provided.</p>";

}

?>

</section>

</main>

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

<script *src*="story.js"></script>

</body>

</html>

Acest script PHP, `delete\_choice.php`, este conceput pentru a șterge un scenariu și datele sale asociate din baza de date.

1. Include Conexiunea la Bază de Date:

include 'connect.php';

Se include fișierul `connect.php`, care conține parametrii de conectare la baza de date și stabilește o conexiune cu aceasta.

2. Verificare Metoda HTTP și Existenta ID-ului de Scenariu:

if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET['id'])) {

Se verifică dacă cererea HTTP este de tip GET și dacă există un parametru `id` în URL.

3. Obținere ID Scenariu și Ștergere Date Asociate:

$scenario\_id = $\_GET['id'];

Se obține ID-ul scenariului din parametrul URL-ului. Apoi, scriptul execută interogări SQL pentru ștergerea datelor asociate scenariului din tabelele `characters`, `circumstance`, `clothes`, `characteristics`, `transport`.

$delete\_characters\_query = "DELETE FROM characters WHERE scenario\_id = $scenario\_id";

// ...

4. Ștergere Scenariu din Tabela Scenarii:

$delete\_scenario\_query = "DELETE FROM scenario WHERE scenario\_id = $scenario\_id";

$conn->query($delete\_scenario\_query);

Se execută o interogare pentru a șterge scenariul din tabela principală `scenario`.

5. Afișare Mesaj de Reușită sau Eroare:

echo "<p>Scenario and related data deleted successfully.</p>";

Dacă operațiunea de ștergere a avut succes, se afișează un mesaj de reușită. În caz contrar, se afișează un mesaj de eroare.

6. Tratarea Cazului în Care ID-ul Scenariului Lipsă:

} else {

echo "<p>Error: Scenario ID not provided.</p>";

}

Dacă ID-ul scenariului nu este furnizat în URL sau dacă cererea HTTP nu este de tip GET, se afișează un mesaj de eroare.

Scriptul are, de asemenea, o structură HTML și CSS pentru a oferi o interfață de utilizator. Acesta afișează un antet cu butoane de acțiune, o secțiune principală cu un formular pentru ștergerea scenariului și un script JavaScript pentru funcționalitățile specifice clientului.

# 3. Povestea modificata

– Dormi, Floarea, vorbim altă dată.

        – Zana Primaverii! Insistă Floarea. Nu vreau să dorm! Vreau să vorbim!

        – Bine, Floarea,! îşi deschide un ochi Zana Primaverii,. Despre ce să vorbim?

        – Despre cât de frumos e Padurea Fermecata…

        – Vai, Floarea,, ţi-am povestit de atâtea ori.

        – Mai vreau, Floarea,!.

        – Atunci ascultă… şi Zana Primaverii, îşi începe povestea. Acolo Padurea Fermecata, şopteşte Zana Primaverii,, adică deasupra Padurea Fermecata – e o altă lume. Nu zic, şi la noi e bine şi e frumos, căci aici ne este casa, dar acolo, cum sa-ţi spun, e altfel… În primul rând e multă lumină. Şi sunt multe, multe culori. Totul e colorat. Da… În sfârşit am înţeles… Culorile… Iată de ce sus e atât de frumos.

        – Zana Primaverii,, când ieşim Padurea Fermecata?

        – Vai Floarea,. De câte ori să-ţi zic? Ai niţică răbdare. Acum e Vara, Ne-am ascuns în căsuţa noastră ca sa nu îngheţăm. Multe gâze şi vietăţi mici când e cald mişună pe afară, când e frig se imbraca in Basma . Chiar şi culorile pare-se dorm pe vreme de Vara. Aşa că… Închide ochii, îţi voi cânta un cântecel, iar tu vei face nani.

        – Nu vreau nani! Nu-mi pasă de frig! Nu-mi place aici!

Oricât de Linistita ar fi Zana Primaverii,, până la urmă îşi iese din fire:

– Încetează! strigă Zana Primaverii,. Vom merge sus cu Avion doar când ni se va da de ştire că e posibil! Gata! Nici o vorbă mai mult! Dormi!

– Zana Primaverii,, începe iarăşi Floarea,. Spune-mi… Cine ne va da de ştire?

– Înţelegi Floarea,, deasupra pământului e cerul cu nori. Norii sunt diferiţi… Adică de diferite culori, de diferite mărimi. Când e primăvară şi cerul se acoperă cu nori negri, tuciurii – norul cel mai important îşi ia toba şi începe să bată: Bah! Bah! Bah! Bah! Toba lui se aude până la noi. Deci vorbim după ce auzim bubuitul. Acum dormi!

– Zana Primaverii,! Auzi!? sare Floarea, din pătucul său. Norul cel mai important bate din tobă!

– Da? nu-i vine să creadă Zana Primaverii,. O fi venit Vara?

– A venit, Zana Primaverii,! A venit! Hai sa mergem afară! Repede! Vreau să văd culorile!

– Nu, Floarea,. Încă nu…

– De ce?! M-ai minţit!? Atunci… plec singur, fără tine!

– Stai, Floarea,! Aşteaptă! Doar un pic. Mai întâi de toate ţărâna trebuie să devină reavănă, puhavă, ca să putem săpa tunelul prin care ne vom strecura. Nu durează mult. Acuş picăturile calde de ploaie se vor scurge în sol… Ah! E atât de plăcut când faci prima scăldătoare de primăvară… Atât de plăcut… Vezi? Solul începe să devină umed. Deci… putem să pornim săpatul.

Cele două râme stau într-o grădină plină de flori, plină de culori. Micuţul Floarea, nu încetează să se minuneze de frumuseţea din jur.

– Zana Primaverii,, zice el, nu mă mai întorc acasă. Îmi place atât de mult aici. Mai ales îmi plac florile… sunt foarte frumoase!

– Eh, oftează Zana Primaverii,. Frumos, frumos… Dar locul unei floare e în pământ. Dacă o floare stă mult timp sub razele soarelui, se usucă. Şi nici măcar nu acest lucru e cel mai important. Prea multe primejdii ne înconjoară… Intunericul, iată răul cel mare. Intunericul,!

– Zana Primaverii,.! Priveşte! Ce fiinţă ciudată aleargă spre noi?! E nespus de nostimă! Iar în urma ei – o mulţime de ghemuleţe aurii! Mă duc să-i salut! strigă Floarea,., apoi porneşte cu îndrăzneală în întâmpinarea cloştii.

– Vai, Floarea, ce faci!? e disperată Zana Primaverii, reuşind totuşi să-şi împingă odrasla înapoi în tunel. Am scăpat, răsuflă ea uşurată. Să nu te fi oprit la timp… Dar… să nu mai vorbim. Principalul e că ai văzut… culorile. Cred că altădată nu mai riscăm. Era cât pe ce să ne înghită o Zana Primaverii,.! …

E mult de lucru în împărăţie. Toată ziulica acestea fărâmă solul ca să-l facă pufos. Floarea şi Zana Primaverii sunt convinşi că şi datorită lor din solul pufos răsar în fiecare primăvară… culorile.

1. **Personaje:** „Zana Primaverii”, „Floarea”, „Lumina”, „Intunericul”, „Fluturasul”, „Lacusta”;
2. **Imprejurari:** „Padurea Fermecata”, „Muntii Inaltimilor”, „Raul Mandrei”;
3. **Anotimp:** „Vara”, „Iarna”, „Primavara”;
4. **Haine:** „Camasa”, „Pantaloni”, „Vesta”, „Caciula”, „Papuci”, „Costum”,;
5. **Caracteristici** „Tanara”, „Vesela”, „Neobisnuita”, „Stralucitor”, „Misterios”, „Optimism”;
6. **Transport:** „Avion”, „Corabie”, „Taxi”;

Concluzie

În cadrul acestei lucrări de laborator, am abordat cu succes dezvoltarea unei aplicații CRUD (Create, Read, Update, Delete) folosind limbajul de programare PHP și sistemul de gestionare a bazelor de date MySQL. Această experiență a oferit o înțelegere detaliată a procesului de interacțiune între server și bază de date, precum și implementarea operațiilor de bază pe datele stocate.

1. Conectarea la Baza de Date:

- Am învățat cum să stabilim o conexiune sigură și eficientă între aplicație și bază de date folosind PHP și MySQL. Folosirea fișierului `connect.php` a asigurat un management corespunzător a conexiunii.

2. Operații CRUD:

- Am implementat funcționalități complete CRUD pentru gestionarea datelor. Fiecare operație (Create, Read, Update, Delete) a fost tratată cu atenție pentru a asigura consistența și securitatea datelor.

3. Securitate:

- Am acordat o atenție deosebită securității prin utilizarea instrucțiunilor pregătite pentru a preveni injecțiile SQL. Această practică esențială contribuie la protejarea aplicației împotriva vulnerabilităților.

4. Interfață Utilizator și Experiență:

- Am dezvoltat pagini HTML clare și intuitive, oferind o interfață prietenoasă utilizatorului. Integrarea cu stiluri CSS și scripturi JavaScript a condus la o experiență de utilizare mai atractivă și dinamică.