**Ministry of Higher Education and Scientific Research **

**University of Ghardaia**

**College of Science and Technology**

**Department of Mathematics and Informatics**

**Specialization in information systems**

Conception and realization a social media website

**Created by :**

AbiMhamed Manel

Goumgoum Soumia

Mrahrah Selma

**:supervisor**

Drgha Houssam Eddine

academic year 2023/2024

Introduction

In the age of digital connectivity, social media platforms have become integral parts of our daily lives, facilitating communication, sharing of ideas, and networking on a global scale. Developing a social media website requires careful consideration of various factors such as user engagement, data flexibility, scalability, and security. In this project, we aim to design and implement a database for a social media website called SMS media with a focus on accommodating dynamic user profiles, posts, interactions, and diverse content types.

*To create a social networking site, we have two ways to display data a Structured Data the Semi Structured Data*

**PARTIE I :**

I.1 Structured data

I.1.1 Definition   
Structured data is data whose elements are addressable for effective analysis. It has been organized into a formatted repository that is typically a database. It concerns all data which can be stored in database SQL in a table with rows and columns. They have relational keys and can easily be mapped into pre-designed fields. Today, those data are most processed in the development and simplest way to manage information. *Example:* Relational data.

I.1.2 Description of workspace & the editor

In our project, we've chosen to leverage Visual Studio Code (VS Code) as our primary code editor, utilizing HTML, CSS, PHP, and MySQL managed through phpMyAdmin for structured data handling. This combination offers a versatile and robust foundation for building dynamic web applications, including social media platforms called MSS media . and we inspired some design of instagram

I.2 *Conception of the project*

**User**

User\_id

User Name

Email

PassWord

Birthdate

Location

Admin

Bio

Creat\_at

img

avoir

1..1

0..N

**Posts**

Post\_id

User\_Id

Content

Content\_Img

Creat\_at

**Comment**

Coment\_Id

Post\_Id

Content

Creat\_at

avoir

1..1

0..N

1..1

contient

0..N

**Like**

Like\_Id

Post\_Id

MCD of the social media

Admin

* Show all users
* Delate user
* Update user
* Add user
* Update his information

User

* My profil
* See all his posts
* See all his comments
* Add comments
* Add posts
* Make Like to a post
* See the other users
* See the number of likes on a post.

Differents functionality for admin and user

I.3 Implimentation  
Therefore, we have 3 interfaces, one for logging in, by entering the email and password, and the data checks whether he is an administrator or user until he is directed to the main page. The second interface is for the administrator, who is responsible for adding users, changing their information, and removing them. The third interface for the user and the home page will be activated. Later, an addition to his personal page, where he was able to add new posts, comments,likes, and show all his comments in every post of his.

LOGIN Interface

This code creates a simple login page for a social media platform called "MSS Media." The page includes input fields for email and password, along with a submit button for user authentication. If there's an error during login (e.g., invalid credentials), an error message is displayed. Additionally, the page includes a footer with links to various platform-related pages and a language selector dropdown. JavaScript files for additional functionality and icon support are also included at the end of the document

ADMIN Interface

This code provides a user-friendly interface for an admin to manage user data on the social media platform, allowing them to view user profiles, edit information, and perform administrative tasks efficiently.   
profile Display:  
Displays the admin's profile information such as username, location, and bio.  
Provides an option to edit the profile.  
User Table:  
Renders a table displaying user data fetched from the database.  
Allows the admin to delete users or edit their information.  
Overlay Forms:  
Displays overlay forms for adding or updating user information.  
These forms include input fields for user details like full name, email, password, birthday, location, bio, and an option to upload an image.  
JavaScript Functions:  
showOverlay(): Displays the overlay form for adding a new user.  
hideOverlay(): Hides the overlay form.  
show(user\_id): Displays the overlay form for updating user information with pre-filled data based on the selected user.  
hide(): Hides the overlay form.

Additionally, a functionality is implemented wherein when an admin deletes a user, it also triggers the deletion of all associated posts, comments, and likes attributed to that user.

USER Interface

Main page :

after user's login to our social media platform he will direct to main page . It includes HTML markup and PHP code to display the other users , posts, and a side menu.  
  
The PHP code retrieves user information from the database and dynamically generates HTML elements to display user accounts in a loop.  
Each accounts includes the user's avatar, username.  
The script also displays posts and the images, usernames, descriptions, and timestamps.  
The side menu section includes the user's profile picture, username, and bio, retrieved from the session data and also there is a header contient buttons such theme, home ,logout

Interface PROFILE

When a user clicks on the profile picture in the side menu section, he is directed to their profile page where they can:

**1. User Profile Display:**

Display their profile picture, name, username, location, birthdate, and bio.

Navigate to their comments page and add a new post.

**2. Main Profile Section:**

View their own posts.

Each post includes their profile picture, username, post content, and options to like or comment.

**My Comments Button:**

Navigate to the user's comments page, linked to "mycomment.php".

**Add Post Button:**

Trigger the display of an overlay for adding a new post, using the show() JavaScript function.

Main Content Section:

Display each post as an <article> element containing:

Post Header: User's profile picture, username, and the post creation date.

Post Content: Post content and an optional image.

Like Button: Allows users to like the post.

Comment Button: Allows users to add a comment to the post, triggering the display of an overlay.

Overlay Sections:

Add Post Overlay: Form for submitting a new post, including fields for post content and image upload.

Add Comment Overlay: Form for submitting a new comment on a specific post.

**Interaction Features:**

**Likes:** Users can like a post, and the number of likes is displayed.

**Comments:** Users can add comments to a post, and existing comments are shown.

Users can interact with posts by liking them and adding comments. These interactions enhance user engagement and foster community interaction within the social media platform.

**And this is how data is manipulated on XML**

<users>

<User\_Id> 1</User\_Id>

<UserName>john\_doe</UserName>

<Email>john@example.com</Email>

<Birthdate>1990-01-01</Birthdate>

<Location>City, Country</Location>

<Admin>false</Admin>

<bio>This is John's bio</bio>

<img> user.png </img>

<posts>

<Post\_Id>1</Post\_Id>

<User\_Id>1</User\_Id>

<Content>this is first John’s posts</Content>

<Content\_Img> summer\_vibe.png </Content\_Img>

<creat\_at>27 feb 2024</creat\_at>

<comments>

<comment\_Id>2</comment\_Id>

<Post\_Id>1</Post\_Id>

<Content> this is second John’s posts </Content>

<creat\_at>28 feb 2024</creat\_at>

<Like>

<post\_Id>1 </post\_Id>

<like\_Id>1</like\_Id>

</Like>

</comments>

</posts>

</users>

The end

Our MSS media social media platform stands as a testament to the power of structured data management and innovative technology integration in creating a dynamic and engaging online community. Our platform offers user-friendly interfaces for login, administration, and user engagement, ensuring a seamless and enjoyable experience for all users. With interactive features like likes and comments, our platform fosters meaningful interactions and connections among users, making MSS media a valuable addition to the social networking landscape.

**PARTIE II :**

II.1 Semi structured data

II.1.1 Definition

Semi-structured data is information that does not reside in a relational database but that has some organizational properties that make it easier to analyze. With some processes, you can store them in the relation database (it could be very hard for some kind of semi-structured data), but Semi-structured exist to ease space. Example: XML data.

II.1.2 Description of workspace & the editor

*We have created a DTD file for semi structured data called Fichier.xml and this is the content*

I.2 *Conception of the project*

<!ELEMENT users (User\*)>

<!ELEMENT User (User\_Id, UserName, Email, Birthdate, Location, Admin, bio, img, posts\*)>

<!ELEMENT User\_Id (#PCDATA)>

<!ELEMENT UserName (#PCDATA)>

<!ELEMENT Email (#PCDATA)>

<!ELEMENT Birthdate (#PCDATA)>

<!ELEMENT Location (#PCDATA)>

<!ELEMENT Admin (#PCDATA)>

<!ELEMENT bio (#PCDATA)>

<!ELEMENT img (#PCDATA)>

<!ELEMENT posts (Post\*)>

<!ELEMENT Post (Post\_Id, User\_Id, Content, Content\_Img, creat\_at, comments\*)>

<!ELEMENT Post\_Id (#PCDATA)>

<!ELEMENT Content (#PCDATA)>

<!ELEMENT Content\_Img (#PCDATA)>

<!ELEMENT creat\_at (#PCDATA)>

<!ELEMENT comments (comment\*)>

<!ELEMENT comment (comment\_Id, Post\_Id, Content, creat\_at, Like\*)>

<!ELEMENT comment\_Id (#PCDATA)>

<!ELEMENT Like (post\_Id, like\_Id)>

<!ELEMENT post\_Id (#PCDATA)>

<!ELEMENT like\_Id (#PCDATA)>

I.3 Implimentation

Our project for the second phase involves implementing a display system utilizing data stored in a file named 'database.json' as well as MongoDB to import the file

LOGIN Interface

The provided code is a PHP file that generates an HTML document for a login page. Here's a description of its structure and functionality:

- \*\*Session Start\*\*: The PHP `session\_start()` function initializes a new session or resumes the existing session.

- \*\*HTML Document Structure\*\*:

- \*\*`<head>` Section\*\*: Contains metadata and links to external resources like CSS and JavaScript files.

- Metadata includes character encoding, viewport settings, author information, and a page description.

- Links to external CSS files (`style.css`) and JavaScript files (`app.js`, `brands.min.js`, `fontawesome.min.js`).

- \*\*`<body>` Section\*\*: Contains the visible content of the web page.

- \*\*`<section>` Element with class "container"\*\*: Represents a container for the login form.

- \*\*`<article>` Element with class "form-area"\*\*: Represents the login form area.

- \*\*`<div>` Element with class "logo"\*\*: Displays the logo "MSS Media."

- \*\*Login Form\*\*:

- Inputs for email and password with placeholders.

- Submit button with the value "LOGIN."

- Links for other options like "Forget password?"

- \*\*Footer Section\*\*:

- \*\*`<footer>` Element\*\*: Represents the footer of the page.

- \*\*`<div>` Element with class "footer-content"\*\*: Contains a list of footer links.

- \*\*`<div>` Element with class "copyright"\*\*: Contains a select element for language selection and copyright information.

- \*\*Scripting\*\*:

- Includes JavaScript files (`app.js`, `brands.min.js`, `fontawesome.min.js`) for client-side scripting.

Admin Interface

The pqge is an administrative page that interacts with a JSON file to manage user data. Here's a description of the JSON structure based on the PHP code:

1. The JSON file (`database.json`) contains user data organized in an associative array with a key named `users`.

2. Each user is represented as an associative array within the `users` array.

3. The user array contains key-value pairs for various user attributes such as `User\_Id`, `UserName`, `Email`, `PassWord`, `Birthdate`, `Location`, `Bio`, `created\_at`, `img`, and `Admin`.

4. The `User\_Id` serves as a unique identifier for each user.

User Interface

The user interface it is an interface to display user profile information and manage posts and comments on a web page. The PHP code interacts with a JSON file to retrieve user data.

Here's a description of the JSON structure based on the PHP code:

1. The JSON file (`database.json`) contains user data organized in an associative array with a key named `users`.

2. Each user is represented as an associative array within the `users` array.

3. The user array contains key-value pairs for various user attributes such as `User\_Id`, `UserName`, `Email`, `PassWord`, `Birthdate`, `Location`, `Bio`, `created\_at`, and `img`.

4. The `User\_Id` serves as a unique identifier for each user.

5. Other attributes such as `UserName`, `Email`, `PassWord`, `Birthdate`, `Location`, `Bio`, `created\_at`, and `img` store the corresponding user information.

6. The JSON structure is used to populate the user profile page with the user's information and manage posts and comments associated with each user.

This JSON structure allows for easy retrieval and manipulation of user data within the PHP code to dynamically generate the user profile page and handle user interactions such as adding posts and comments.

5. Other attributes such as `UserName`, `Email`, `PassWord`, `Birthdate`, `Location`, `Bio`, `created\_at`, `img`, and `Admin` store the corresponding user information.

6. The `Admin` attribute is a boolean flag indicating whether the user is an administrator (`1`) or not (`0`).

Conclusion

the combination of structured and semi-structured data management techniques in our MSS platform enables us to deliver a versatile, scalable, and user-friendly platform that caters to the diverse needs of our online community.