

Documentation for Tamagotchi game

Installation instructions:

1. Download XAMPP from <https://www.apachefriends.org/download.html>
2. Install XAMPP with these minimal options: Apache, MySQL, PHP.
3. Open XAMPP Control Panel and start Apache and MySQL.
4. All PHP scripts should be stored inside /xampp/htdocs. That means that the Tamagotchi file should be stored inside this path.
5. To create the database, go to XAMPP Control Panel > MySQL > Admin button. You will be redirected to localhost/phpmyadmin. From there you can create/configure your DB.
The database should be named „Tamagotchi”, and it should contain the tables „user” and „pet”. When creating those tables, make sure that the „user” table has the attributes id(of type int, also you should check the autoincrement option), username(of type varchar, length 100) and password(of type varchar, length 100). It should look like this:

#	Nume	Tip	Colaționare	Atribut	Nul	Implicit	Comentarii	Suplimentar	Acțiune
<input type="checkbox"/> 1	id	int(11)			Nu	Niciuna		AUTO_INCREMENT	Modifică Elimină Mai mult
<input type="checkbox"/> 2	username	varchar(100)	utf8mb4_general_ci		Nu	Niciuna			Modifică Elimină Mai mult
<input type="checkbox"/> 3	password	varchar(100)	utf8mb4_general_ci		Nu	Niciuna			Modifică Elimină Mai mult

Also, the „pet” table has the attributes id(of type int, also you should check the autoincrement option), health(type int), happiness(type int), clean(type int), age(type int) and userid(type int). You should also establish the relation between the user and pet, by adding a foreign key constraint on the userid attribute from the pet table. Click on the pet table, go at „Structura”, then click on „Vizualizare relatie” and do it exactly like in this picture.

Rasfoiește Structură SQL Caută Inserează Exportă Importă Privilegii Operații Urmărire

Structură tabel Vizualizare relație

Constrângeri cheie străină

Acțiuni	Proprietăți constrângere	Coloană	Constrângere cheie străină (INNODB)		
			Bază de date	Tabel	Coloană
<input type="text" value="foreign key"/>					
Elimină	ON DELETE	CASCADE			
	ON UPDATE	CASCADE			

+ Aadaugă coloană

userid tamagotchi user id

Once you did those steps, the pet table should look like this:

#	Nume	Tip	Coloționare	Atribut	Nul	Implicit	Comentarii	Suplimentar	Acțiune
<input type="checkbox"/> 1	id	int(11)			Nu	Niciuna		AUTO_INCREMENT	Modifică Elimină Mai mult
<input type="checkbox"/> 2	health	int(11)			Nu	Niciuna			Modifică Elimină Mai mult
<input type="checkbox"/> 3	happiness	int(11)			Nu	Niciuna			Modifică Elimină Mai mult
<input type="checkbox"/> 4	clean	int(11)			Nu	Niciuna			Modifică Elimină Mai mult
<input type="checkbox"/> 5	age	int(11)			Nu	Niciuna			Modifică Elimină Mai mult
<input type="checkbox"/> 6	userid	int(11)			Nu	Niciuna			Modifică Elimină Mai mult

6. In order to start the app, you have to type in the search bar of your browser <http://localhost/Tamagotchi/index.php> .

Implementation details:

- The **configuration.php** file contains the configuration settings for connecting to the database(named „Tamagotchi”).
- The file **login.php** handles the user login process. To determine whether the supplied username and password are a match for any entries in the „user” table, it communicates with the database (using the settings from configuration.php). Data can be persistent across several pages for the same user once a session has been initiated. If the \$_POST['submit'] variable is set, the username and password supplied on the login form are retrieved upon submission.

The database's „user” table is then queried to see if the supplied username and password combination is present in the table. The user is directed to the backend.php page if a match is discovered. If not, it redirects the user back to index.php and displays an alert message informing the user that the login failed due to incorrect username or password.

- The file **index.php** includes configuration.php for connecting to the database. It also contains login.php, which is where the login logic is located.

Users can enter their username and password on the html form. The data from the login form is sent to login.php for processing when it is submitted. If the username field is empty,it displays an alert message informing the user that the login failed due to empty username and the user is redirected to index.php.

- The file **signin.html** offers an html form for people to use to register for a new account. It consists of buttons to submit the form or stop the process, as well as input fields for the username and password. The creation of a new user account is handled by the **signin.php** file. It gets information from the signin.html form and adds the username and password of the new user to the „user” table.
- The Tamagotchi game's home page, **backend.php**, allows users to interact with their virtual pet. It contains the configuration.php file to set up the database connection as well as other relevant functions.

To enable session management for the user, it creates a new session.

The user is sent back to index.php and the script is terminated if the the user is not logged in.

getUserId is a function that retrieves the user id based on the provided username.

createNewPet is a function that creates a new pet for the given user id with default attributes(health=100,happiness=100,clea=100,age=0 and user id is the current user id).

fetchPetData is a function that retrieves the pet's data based on the provided user id. If the pet data doesn't exist, it creates a new pet and then fetches its data.

isPetDead : Checks if the pet is dead based on its health, happiness, and clean attributes(if any of those attributes are 0 or less).

updatePetStatus: Updates the pet's health, happiness, clean, and age based on the provided changes.

decreaseAttributes: Reduces the pet's clean, happiness, and health attributes by 3.

The file verifies that the action parameter is set and that the form was submitted.

Based on the submitted action, the script updates the pet's status and then redirects the user back to backend.php to update the displayed information.

If the pet is still alive, JavaScript is used to automatically refresh the page every second. This makes sure that the user always sees the most recent information regarding the pet.

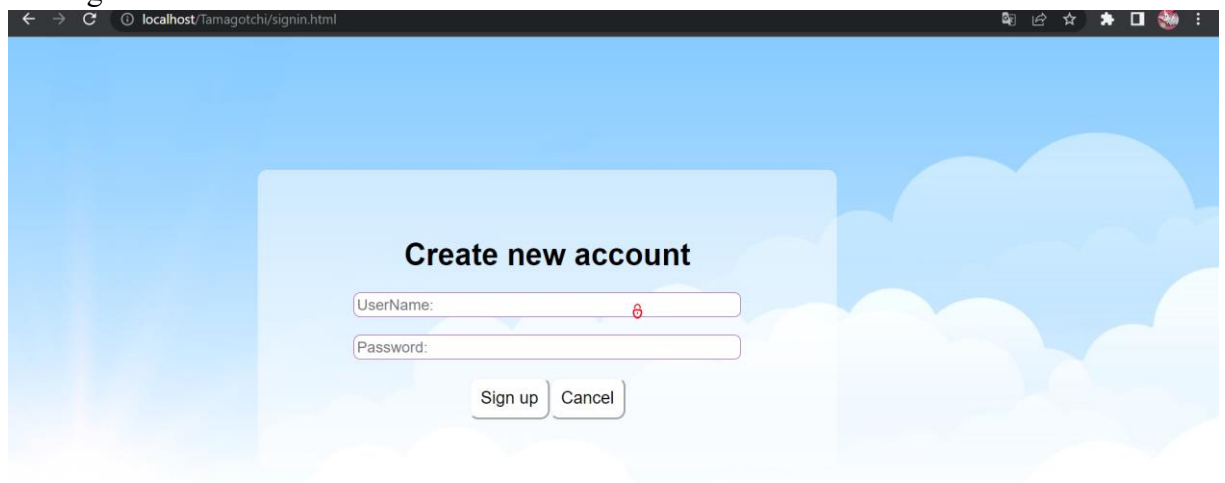
- The file **process_action.php** is responsible for processing the user's actions from the Tamagotchi game interface. It receives the action parameter from the form in backend.php, which specifies what action the user wants to perform on their pet (for example feed, play, clean, restart).

How the app looks like:

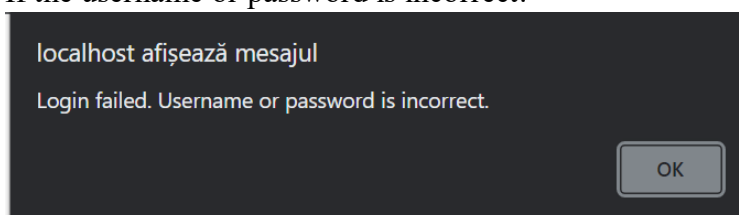
- The login form:



- The sign in form:

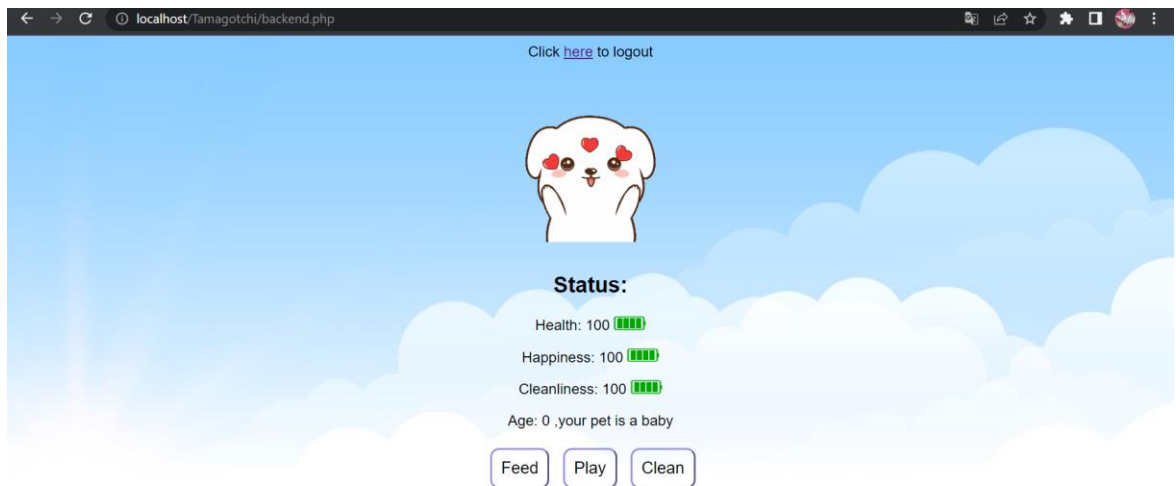


- If the username or password is incorrect:

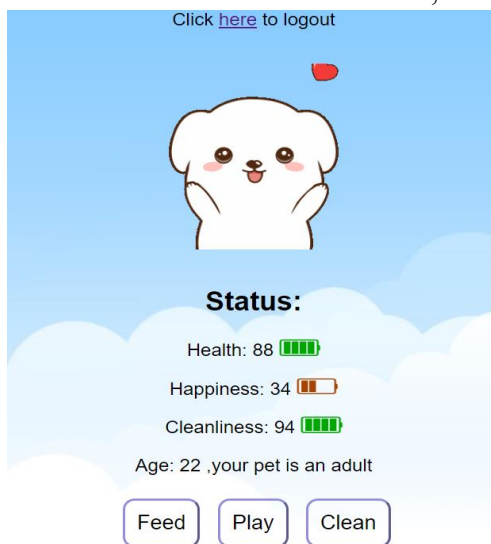


Note: it looks the same if the username is empty, it just displays “UserName field is empty!!!”

- At the beginning of the game, the pet has full health and age 0:



- The attributes decrease over time, while the age increases.



- When one of the attributes reaches 0, the pet dies.

