🐳 CSC640 Project

Welcome! This tutorial walks you through running my CSC640 project, an animal shelter API, using Docker. The project can be downloaded from https://github.com/jonesa122/CSC640-Project

📁 Project Structure

Your folder should include:

```
CSC640-Project/
├── compose.yaml
├── Dockerfile.php
├── Dockerfile.nginx
├── .dockerignore
├── php/
├── nginx/
├── sql/
```

🚀 Start the Project with Docker

Make sure Docker Desktop is running, then open a terminal in your project folder and run:

```
docker compose up --build
```

🧪 Run API Test Script

To test all endpoints in the CSC640 project, use the provided shell script:

`test-api.sh`

It covers:

- User registration and login
- Animal listing, creation, update, deletion
- Adoption submission and updates

🚀 How to Run It

1. Open your terminal in the project folder
2. Make the script executable:

```
chmod +x test-api.sh
```

3. Run the script:

```
./test-api.sh
```

🌐 Open the App

http://localhost/frontend/index.html

To access a frontend page to interact with the API

👤 Users Endpoint Overview

The frontend accesses:

`http://localhost/index.php?endpoint=users`

This endpoint supports:

- `POST` → register a user, login

📝 How Registration Works

The frontend form sends a POST request to:

`http://localhost/index.php?endpoint=users&action=register`

with a JSON body like:

```json
{
  "username": "newuser",
  "email": "newuser@example.com",
  "password": "securepass"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "id": 4
}
```

If unsuccesful:

```json
{
  "error": "Missing required fields"
}
```

```json
{
  "error": "Account creation failed"
}
```

🔑 How Login Works

The frontend sends a POST request to:

`http://localhost/index.php?endpoint=users&action=login`

with a JSON body like:

```json
{
  "email": "newuser@example.com",
  "password": "securepass"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "token": "JWT_TOKEN_HERE"
}
```

If unsuccessful:

```json
{
  "error": "Invalid credentials"
}
```

```json
{
  "error": "Missing email or password"
}
```

🐾 Animals Endpoint Overview

The frontend accesses:

`http://localhost/index.php?endpoint=animals`

This endpoint supports:

- `GET` → list, view, or search animals
- `POST` → create a new animal
- `PATCH` → update an animal
- `DELETE` → remove an animal

🔍 GET All animals

```
GET /index.php?endpoint=animals
```

→ Returns an array of all animals

🔍 GET All animals

```
[
 {
  "id":"1",
  "name":"Bella",
  "species":"Dog",
  "breed":"Labrador Retriever",
  "age":"3",
  "gender":"Female",
  "arrival_date":"2025-10-01",
  "status":"Adopted"
 },
 {
  "id":"2",
  "name":"Max",
  "species":"Cat",
  "breed":"Siamese","age":"2","gender":"Male",
  "arrival_date":"2025-09-15",
  "status":"Fostered"
  }
 ]
```

🔍 Get animal by ID

`GET /index.php?endpoint=animals&id=3`

→ Returns one animal or 404 if not found

```
{
  "id":"3",
  "name":"Max",
  "species":"Cat",
  "breed":"Siamese","age":"2","gender":"Male",
  "arrival_date":"2025-09-15",
  "status":"Fostered"
}
```

🔍 Search animals by attributes

- Include whichever parameters and values are to be searched for, no need to include parameters that are not being searched
  `GET /index.php?endpoint=animals&action=search&name=Bella&breed=Retriever&species=Dog&age=3&gender=Female&status=Available`
  → Returns filtered results

🔍 Search animals by attributes

```
[
 {
   "id":"4",
   "name":"Bella",
   "species":"Dpg",
   "breed":"Retriever",
   "age":"3",
   "gender":"Female",
   "arrival_date":"2025-09-15",
   "status":"Available"
 }
]
```

+ Add New Animal

To add an animal, the frontend sends a POST request to:

`http://localhost/index.php?endpoint=animals&action=create`

Requires JWT in `Authorization` header

```
{
  "name": "Luna",
  "species": "Dog",
  "breed": "Beagle",
  "age": 2,
  "gender": "Female",
  "status": "Available"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "token": "JWT_TOKEN_HERE"
}
```

If unsuccessful:

```json
{
  "error": "Insert failed"
}
```

```json
{
  "error": "Unauthorized"
}
```

✏️ Update Animal

To update an animal, the frontend sends a PATCH request to:

`http://localhost/index.php?endpoint=animals&action=update&id=3`

Requires JWT in `Authorization` header

The JSON body includes only the fields to update:

```json
{
  "name": "Luna",
  "status": "Adopted"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "updated_id": 3
}
```

✅ What It Returns

If unsuccessful:

```
{
  "error": "No valid fields provided for update"
}
```

```
{
  "error": "Unauthorized"
}
```

```
{
  "error": "Update failed",
  "details": "SQL error message"
}
```

🗑️ Delete Animal

To delete an animal, the frontend sends a DELETE request to:

```
http://localhost/index.php?endpoint=animals&action=delete&id=3
```

Requires JWT in `Authorization` header

✅ What It Returns

If successful:

```json
{
  "success": true,
  "deleted_id": 3
}
```

✅ What It Returns

If unsuccessful:

```json
{
  "error": "Unauthorized"
}
```

```json
{
  "error": "Unauthorized"
}
```

🐾 Adoptions Endpoint Overview

The frontend accesses:

`http://localhost/index.php?endpoint=animals`

This endpoint supports:

- `GET` → adoption request by id
- `POST` → create a new adoption request
- `PATCH` → update an adoption request

📄 Get Adoption Request by ID

To retrieve a specific adoption request, the frontend sends a GET request to:

`http://localhost/index.php?endpoint=adoptions&id=5`

This returns the full adoption record for the given ID, if it exists.

✅ What It Returns

If successful:

```
{
  "id": 5,
  "animal_id": 2,
  "adoption_date": "2025-11-03",
  "adopter_name": "Liam Smith",
  "adopter_phone": "555-5678",
  "adopter_email": "liam.smith@example.com",
  "adopter_address": "456 Oak Ave, Fort Mitchell, KY",
  "status": "approved"
}
```

✅ What It Returns

If unsuccessful:

```
{
  "error": "Adoption request not found"
}
```

🐶 Submit Adoption Request

To submit an adoption request, the frontend sends a POST request to:

`http://localhost/index.php?endpoint=adoptions`

with a JSON body like:

```json
{
  "animal_id": 2,
  "adopter_name": "Liam Smith",
  "adopter_phone": "555-5678",
  "adopter_email": "liam.smith@example.com",
  "adopter_address": "456 Oak Ave, Fort Mitchell, KY"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "id": 5
}
```

✅ What It Returns

If unsuccessful:

```
{
  "error": "Invalid animal_id: no such animal exists"
}
```

```
{
  "error": "Missing required fields"
}
```

```
{
  "error": "Adoption request failed",
  "details": "SQL error message"
}
```

✏️ Update Adoption Request

To update an adoption request, the frontend sends a PATCH request to:

`http://localhost/index.php?endpoint=adoptions&action=update&id=5`

Requires JWT in `Authorization` header

The JSON body includes any fields to update:

```json
{
  "status": "approved",
  "adopter_phone": "555-9999"
}
```

✅ What It Returns

If successful:

```json
{
  "success": true,
  "updated_id": 5
}
```

✅ What It Returns

If unsuccessful:

```json
{
  "error": "No valid fields provided for update"
}
```

```json
{
  "error": "Invalid or expired token"
}
```

```json
{
  "error": "Failed to update adoption request",
  "details": "SQL error message"
}
```

🛑 Stopping the Project

To shut delete the CSC640 Docker containers, open your terminal in the project folder and run:

```
docker compose down -v
```