# SQL Injection Vulnerability in Ecommerce-Clothing-Website

The SQL injection vulnerability exists in the `/log.php` file, specifically in lines 69-79, where the following code is present:

```php

$log\_email = $\_POST['cemail'];

$log\_pass = $\_POST['password'];

$c\_id = $log\_email;

$sel\_customer = "select \* from customer where customer\_email = '$log\_email' AND customer\_pass = '$log\_pass'";

```

User input is \*\*directly concatenated into the SQL statement without any filtering\*\*, resulting in a SQL injection vulnerability in the login function.

## Vulnerability Principle

When a user enters `'or'1'='1` in the password field, the original SQL statement is spliced into:

```sql

select \* from customer where customer\_email = 'input\_email' AND customer\_pass = ''or'1'='1'

```

Since the condition `'1'='1'` is always true and connected via the `OR` operator, the entire query condition becomes permanently true. This causes the database to return all user records (or at least the first record), thereby bypassing password verification and allowing direct login to the system.

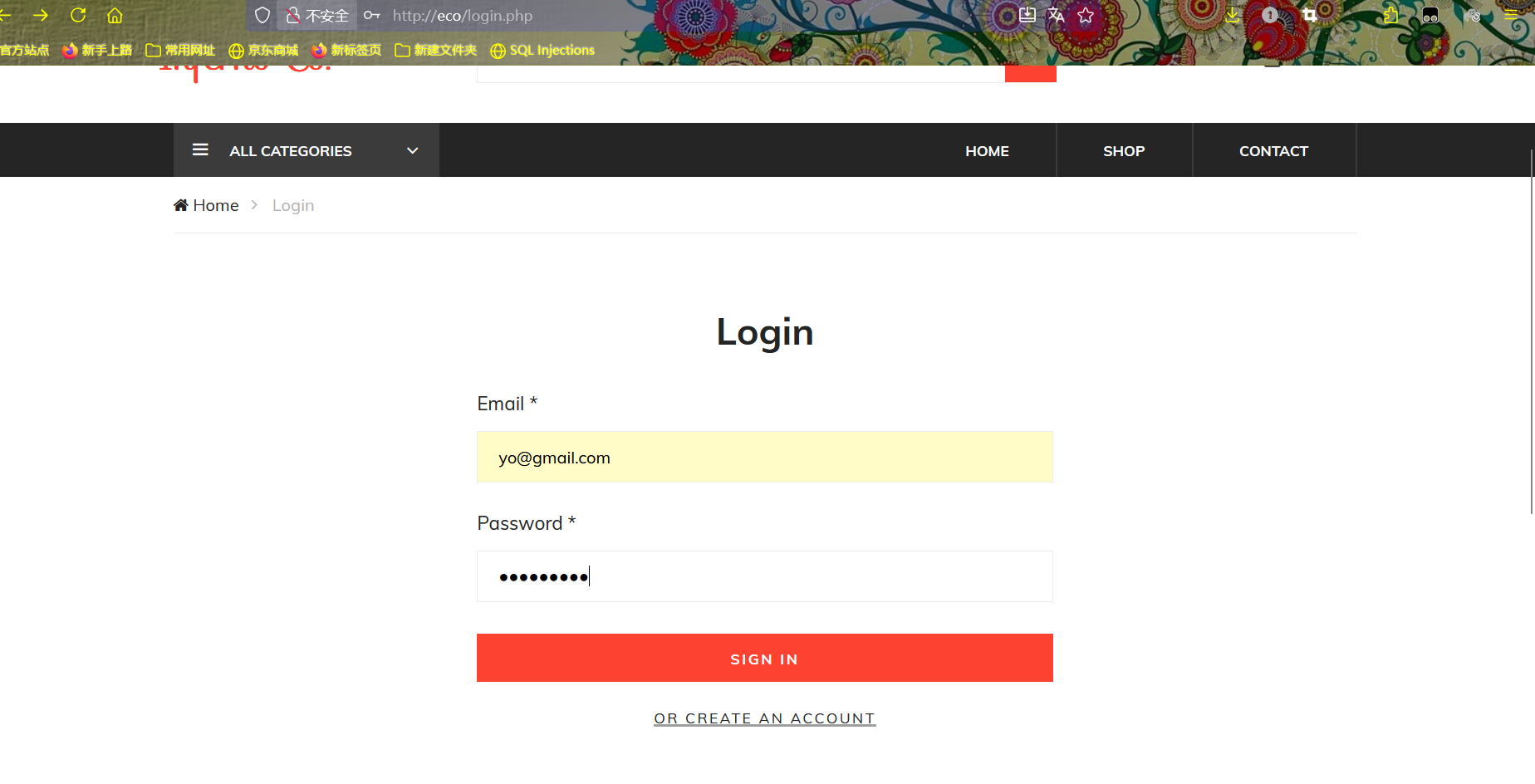
## Demonstration

1. \*\*Access the Login Page\*\*

   Navigate to `http://eco/login.php` and enter a valid email (e.g., `yo@gmail.com`) in the "Email" field. Leave the "Password" field to be filled with the malicious input later.

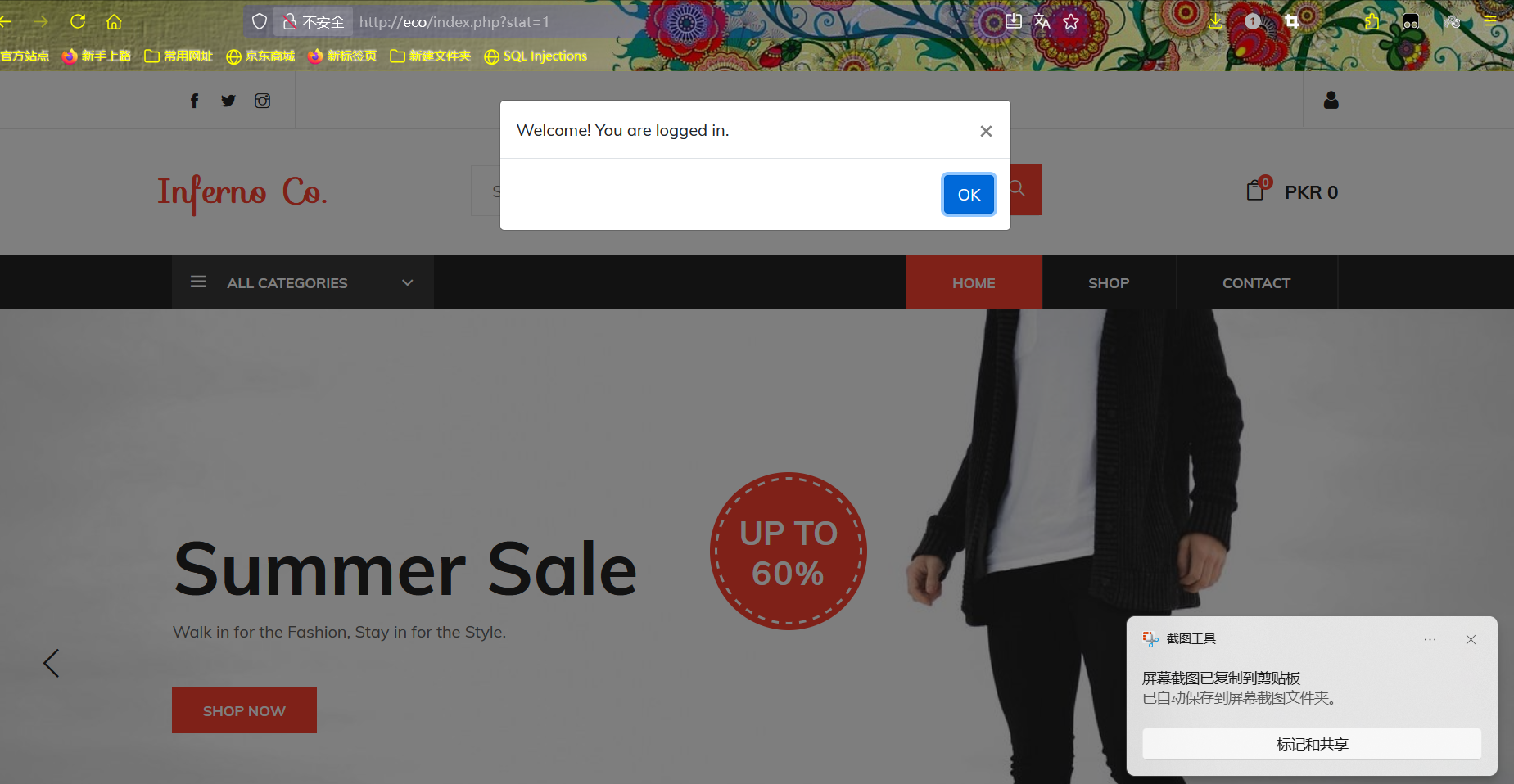
2. \*\*Inject Malicious Input\*\*

   On the `/log.php` page, enter `'or'1'='1` in the "Password" field.



3. \*\*Successful Bypassed Login\*\*

   After submitting the input, the system will bypass password verification and log in successfully, redirecting to the homepage with a "logged in" prompt.



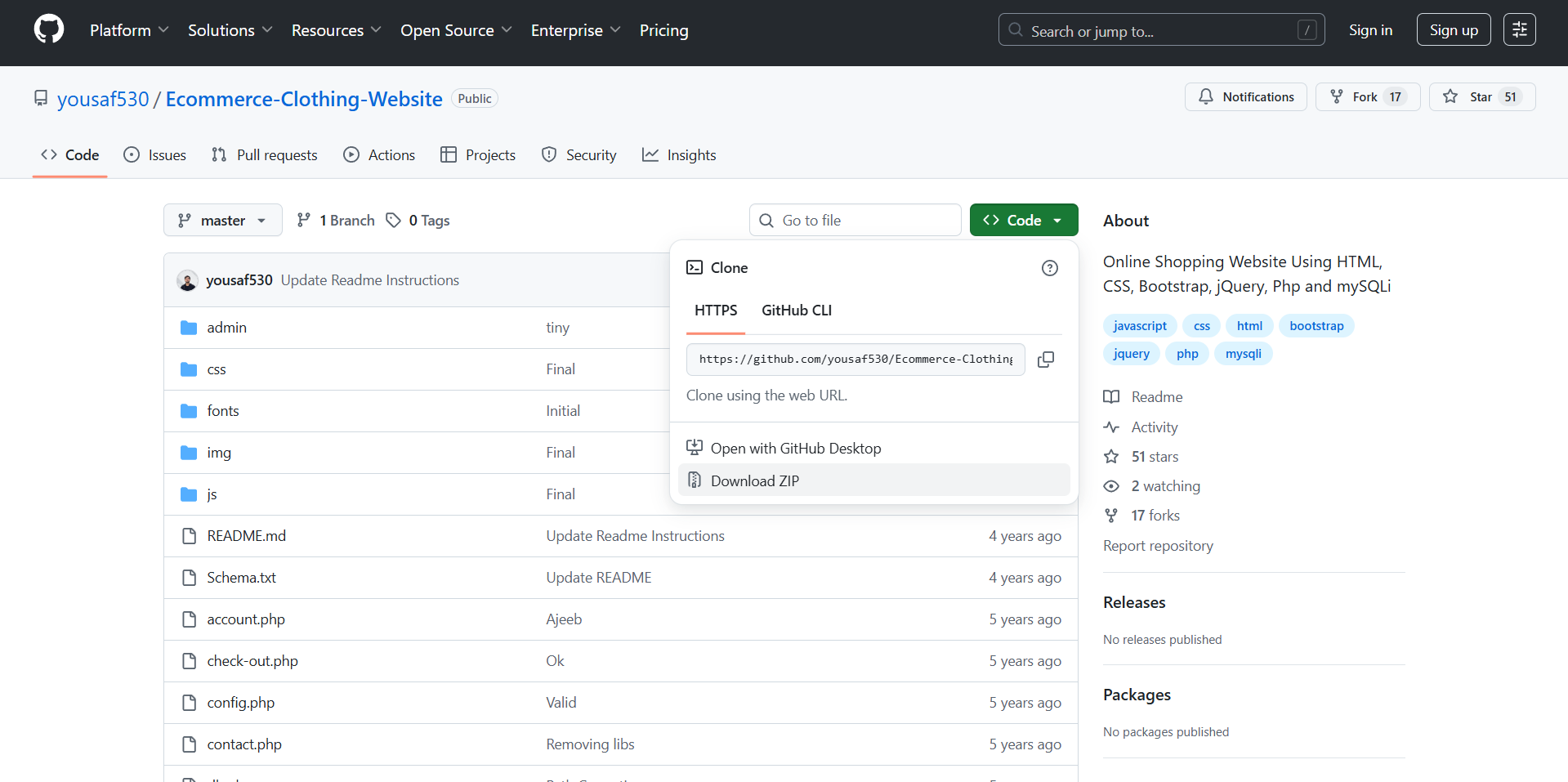
# Project Setup and Vulnerability Reproduction

## 1. Download the Project

Download the project from GitHub via the following link:

https://github.com/yousaf530/Ecommerce-Clothing-Website

### Main Directory Structure



## 2. Deploy with PHPStudy

### Step 1: Start PHPStudy Services

- Open PHPStudy.

- Start the \*\*Apache\*\* and \*\*MySQL\*\* services.

### Step 2: Place Project Files

- Copy the project folder `Ecommerce-Clothing-Website-master` to the web root directory of PHPStudy (usually `D:\phpstudy\_pro\WWW\`).

- Create a website in PHPStudy and set the root directory to:

  `D:\phpstudy\_pro\WWW\Ecommerce-Clothing-Website-master`



### Step 3: Create and Configure the Database

1. \*\*Create a New Database\*\*

   - Create a database named `threaderz\_store`. (Note: Since I already had an existing `threaderz\_store`, I created `threaderz\_store\_new` instead.)

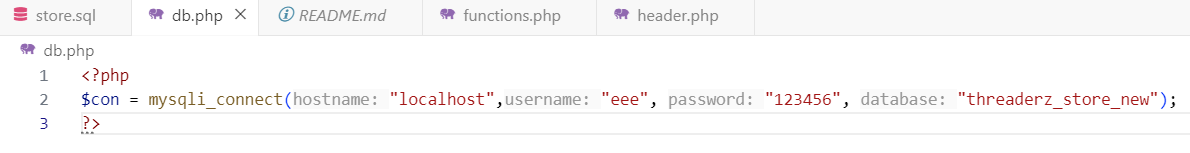
2. \*\*Import Database File\*\*

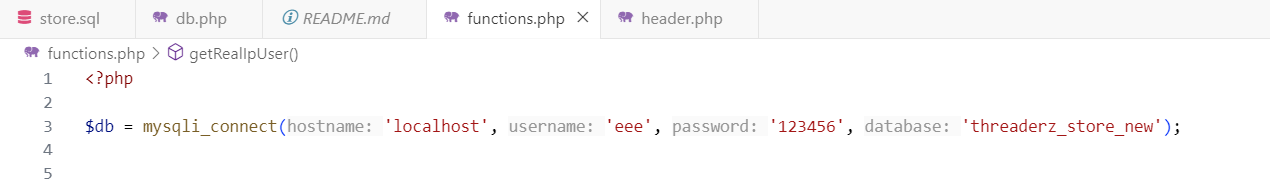
   - Import the `store.sql` file into the newly created database (e.g., `threaderz\_store\_new`).



### Step 4: Configure Database Connection

- Open `db.php` and `functions.php`, then modify the database connection parameters to match your own database credentials.





### Step 5: Access the Website

- In PHPStudy, open the configured website (e.g., access `http://eco` via a browser).

### Step 6: Test Basic Functions

1. \*\*Register/Login\*\*

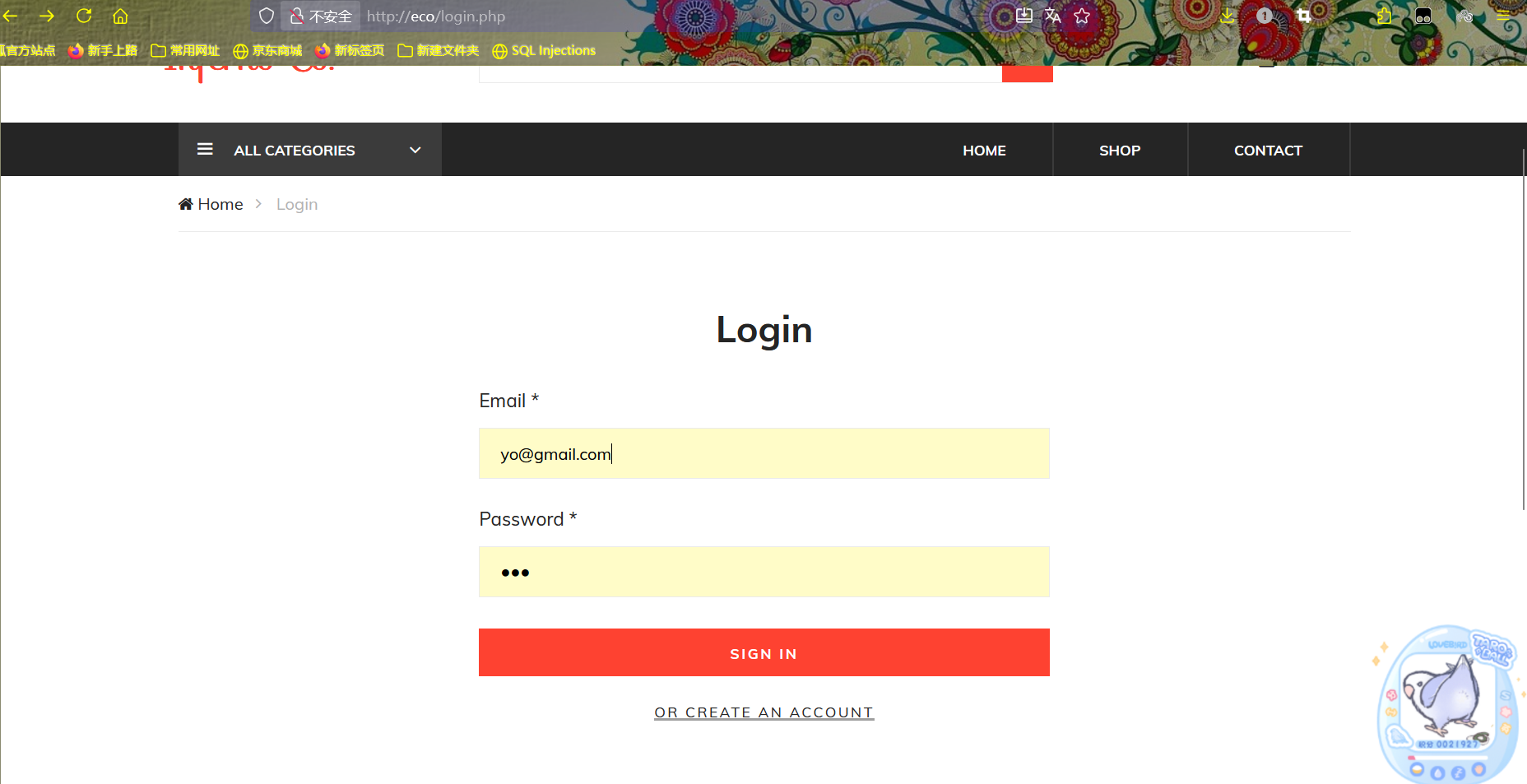
   - Navigate to `/log.php` for login.

2. \*\*Test with Default Account\*\*

   According to `store.sql`, there is a preconfigured test account:

   - Email: `yo@gmail.com`

   - Password: `123`

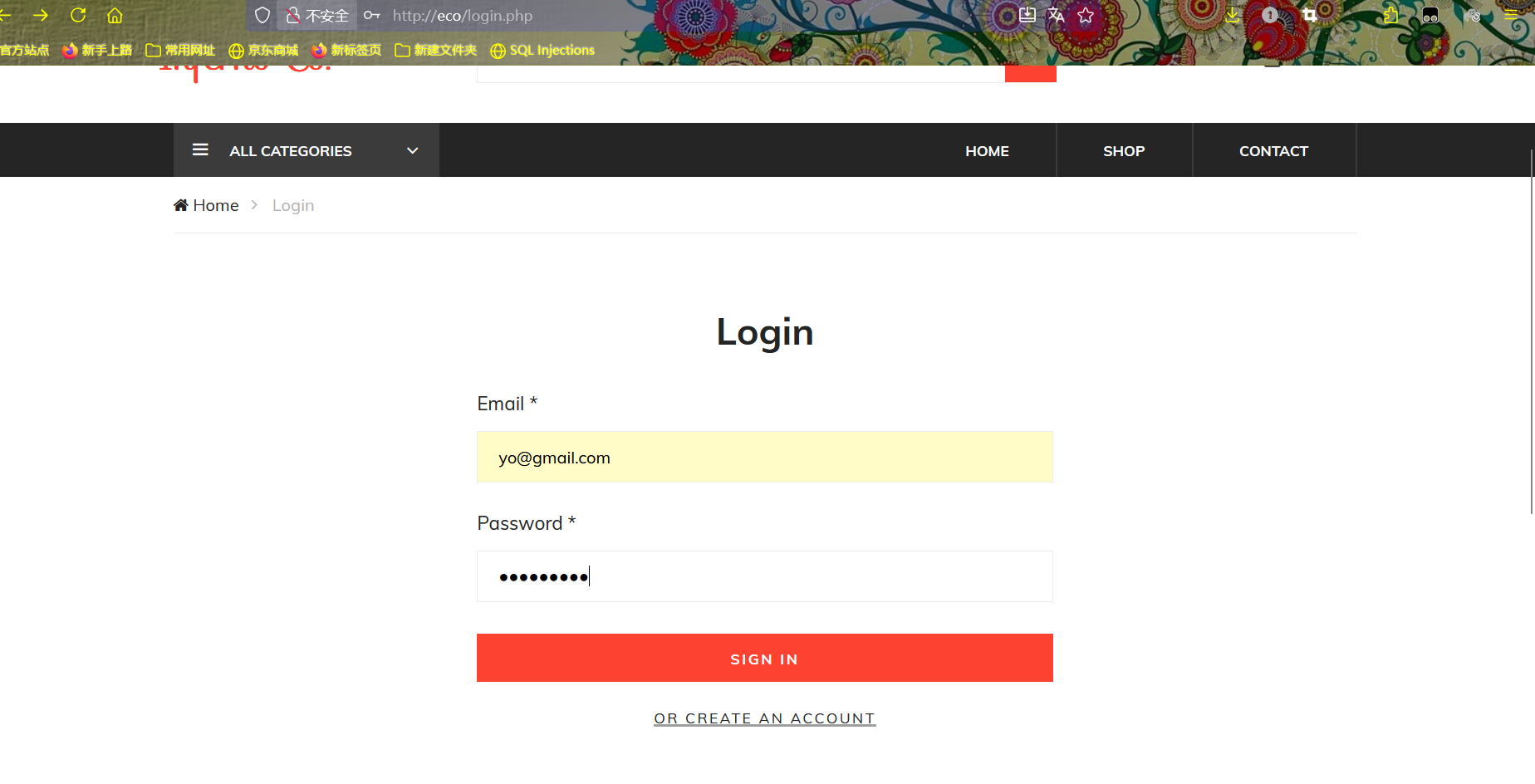


## 3. Reproduce the SQL Injection Vulnerability

1. Go to the `/log.php` login page.

2. Enter a valid email (e.g., `yo@gmail.com`) in the "Email" field.

3. Enter `'or'1'='1` in the "Password" field.



4. Click the "SIGN IN" button.

5. Observe that you can log in successfully without entering the correct password, confirming the SQL injection vulnerability.

