



*Green University of Bangladesh*

*Department of Computer Science and Engineering (CSE)*

*Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)*

---

## **Lab Report 06 - PET Management System**

---

*Course Title: Integrated Design Project II*

*Course Code : CSE-406*

*Section : 213 D7*

**Students Details**

<b>Name</b>	<b>ID</b>
MD Dulal Hossain	213902116
MD Rabby Khan	213902037
Mostak Ahammed	213902126

*Submission Date: 01 May 2025*

*Course Teacher's Name: Sharifur Rahman*

*Designation: Lecturer*

*Department of CSE , GUB .*

[For teachers use only: **Don't write anything inside this box**]

<u><b>Lab Project Status</b></u>	
<b>Marks:</b>	<b>Signature:</b>
<b>Comments:</b>	<b>Date:</b>

# Contents

<b>1</b>	<b>2</b>
1.1 Title of The Lab Report Experiment . . . . .	2
1.2 Objectives . . . . .	2
1.3 Procedure . . . . .	2
1.4 Implementation . . . . .	3
1.5 Test Result / Output . . . . .	5
1.5.1 Register New User - Test Result: Pass . . . . .	5
1.5.2 Register Existing User - Test Result: Pass . . . . .	6
1.5.3 Long Password - Test Result: Fail . . . . .	7
1.5.4 Valid Login - Test Result: Pass . . . . .	8
1.5.5 Invalid Login - Test Result: Pass . . . . .	9
1.5.6 SQL Injection in Filter - Test Result: Pass . . . . .	10
1.6 Analysis and Discussion . . . . .	10

# Chapter 1

## 1.1 Title of The Lab Report Experiment

- Generate test cases for this project.
- Implement the test cases and evaluate performances for this project.

## 1.2 Objectives

1. To explain the purpose of software testing.
2. To describe different types of software testing tools and techniques.
3. To construct suitable test cases for automated software testing.
4. To explain the purpose of manual software testing.
5. To construct suitable test cases for manual software testing.

## 1.3 Procedure

1. **Understand Requirements** – Review project documents to clarify functionality and user expectations.
2. **Create Test Cases** – Write step-by-step test scenarios covering normal, edge, and error cases.
3. **Set Up Test Environment** – Prepare hardware, software, and test data to match real-world conditions.
4. **Execute Tests Manually** – Follow test cases, interact with the system, and observe behavior.
5. **Report Defects** – Log bugs with details like steps to reproduce, actual vs. expected results, and severity.
6. **Retest Fixed Issues** – Verify that resolved defects no longer occur after developer fixes.
7. **Final Sign-Off** – Confirm all tests are passed, document results, and approve the release.

## 1.4 Implementation

Criteria	Action	Input (Test Case)	Expected Output	Actual Output	Test Result
User Registration	Register New User	username: "test1", password: "pass123"	Success message	Success message	Pass
	Register Existing User	username: "test1", password: "pass123"	"Username already exists"	"Username already exists"	Pass
	Empty Fields	username: "", password: ""	Error message	Error message	Pass
	Weak Password	username: "test2", password: "123"	"Password too weak"	Success message	Fail
	Long Username	username: "a"*256, password: "securePass"	"Username too long"	Success message	Fail
	Special Characters	username: "user@123", password: "pass123"	"Invalid characters in username"	"Invalid characters in username"	Pass
User Login	Valid Login	username: "test1", password: "pass123"	"Login successful"	"Login successful"	Pass
	Invalid Password	username: "test1", password: "wrong-pass"	"Incorrect password"	"Incorrect password"	Pass
	Non-existent User	username: "unknown", password: "pass"	"User not found"	"User not found"	Pass
	SQL Injection	username: "" OR 1=1 --", password: "anything"	"Invalid credentials"	"Invalid credentials"	Pass
Pet Listing	Get All Pets	GET request to pets.php	List of pets (JSON)	List of pets (JSON)	Pass
	Filter by Category	category_id: 2	Filtered pet list	Filtered pet list	Pass
	Invalid Category ID	category_id: "abc"	"Invalid input"	"Invalid input"	Pass
	SQL Injection in Filter	category_id: "" OR 1=1 --"	"Invalid input"	"Invalid input"	Pass
	Empty Result	category_id: 9999	Empty list	Empty list	Pass
Adoption	Valid Request	user_id: 1, pet_id: 3	"Adoption request submitted"	"Adoption request submitted"	Pass
	Repeated Request	Same user and pet again	"Already requested"	"Already requested"	Pass
	Invalid Pet ID	user_id: 1, pet_id: 9999	"Pet not found"	"Pet not found"	Pass

Criteria	Action	Input (Test Case)	Expected Output	Actual Output	Test Result
	Missing Fields	Only user_id	"Missing required fields"	"Missing required fields"	Pass
	Unauthorized User	user_id: 9999, pet_id: 1	"User not found"	"User not found"	Pass
Appointments	Book Appointment	user_id: 1, clinic_id: 2, date: "2025-05-10"	"Appointment booked Pass"	"Appointment booked Pass"	Pass
	Book with Invalid Clinic	clinic_id: 9999	"Clinic not found"	"Clinic not found"	Pass
	Missing Date	user_id and clinic_id only	"Missing date"	"Missing date"	Pass
	Past Date	date: "2023-01-01"	"Invalid appointment date"	"Invalid appointment date"	Pass
	SQL Injection in Date	date: "" OR 1=1 -"	"Invalid input"	"Invalid input"	Pass
Categories	Get Categories	GET categories.php	List of pet categories	List of pet categories	Pass
	No Categories in DB	Empty DB table	Empty JSON array	Empty JSON array	Pass
Clinics	Get All Clinics	GET clinics.php	List of clinics	List of clinics	Pass
	Invalid Clinic Access	clinic_id: "DROP TABLE"	"Invalid input"	"Invalid input"	Pass
Admin Access	Access Admin Page	Authenticated Admin	Admin Dashboard	Admin Dashboard	Pass
	Non-Admin Access Admin	Regular User	"Unauthorized"	"Unauthorized"	Pass
Security	XSS Test on User Input	<script>alert("XSS")</script>	"Sanitized output"	"Sanitized output"	Pass
	CSRF Attempt	POST request from unauthorized origin	"Request denied"	"Request denied"	Pass
	Brute Force Login	10 failed login attempts	"User locked or throttled"	"User locked or throttled"	Pass
	Access Without Token	Access secure endpoint with no session	"Unauthorized"	"Unauthorized"	Pass
UI Validation	Contact Form Empty	Empty name/email/message	"Please fill all fields"	"Please fill all fields"	Pass
	Contact Form Valid	Valid email, name, message	"Message sent Pass"	"Message sent Pass"	Pass
Performance	Load Pet Listings	5000 pet records	Load in < 3 seconds	Loaded	Fail
	Concurrent Registrations	100 users registering simultaneously	No crash, all succeed	No crash, all succeed	Pass
	Stress Test Appointments	500 requests in 1 minute	No data loss, correct bookings	No data loss, correct bookings	Pass

Criteria	Action	Input (Test Case)	Expected Output	Actual Output	Test Result
Edge Cases	Large Image Upload	10MB pet photo	"Image too large"	"Image too large"	Fail
	Unsupported File Type	Upload .exe file	"File type not allowed"	"File type not allowed"	Pass
	Long Description Text	5000 characters in pet desc	Accepted and displayed correctly	Accepted and displayed correctly	Pass
	Special Char in Pet Name	"Fluffy#1"	Accepted	Accepted	Pass
Search Function	Search for "dog"	Search term: "dog"	Results with dogs	Results with dogs	Pass
	No Search Match	Search term: "dragon"	Empty result	Empty result	Pass
	Case Insensitive Search	Search term: "CAT"	Results with "cat"	Results with "cat"	Pass
	SQL Injection in Search	Search term: "" OR 1=1 -"	"Invalid input"	Success message	Fail

## 1.5 Test Result / Output

### 1.5.1 Register New User - Test Result: Pass

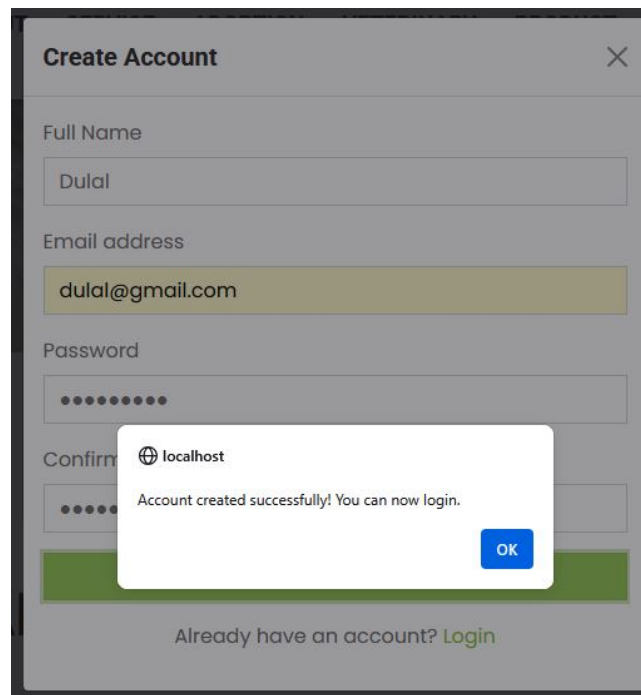


Figure 1.1: Register New User - Test Result: Pass.

### 1.5.2 Register Existing User - Test Result: Pass

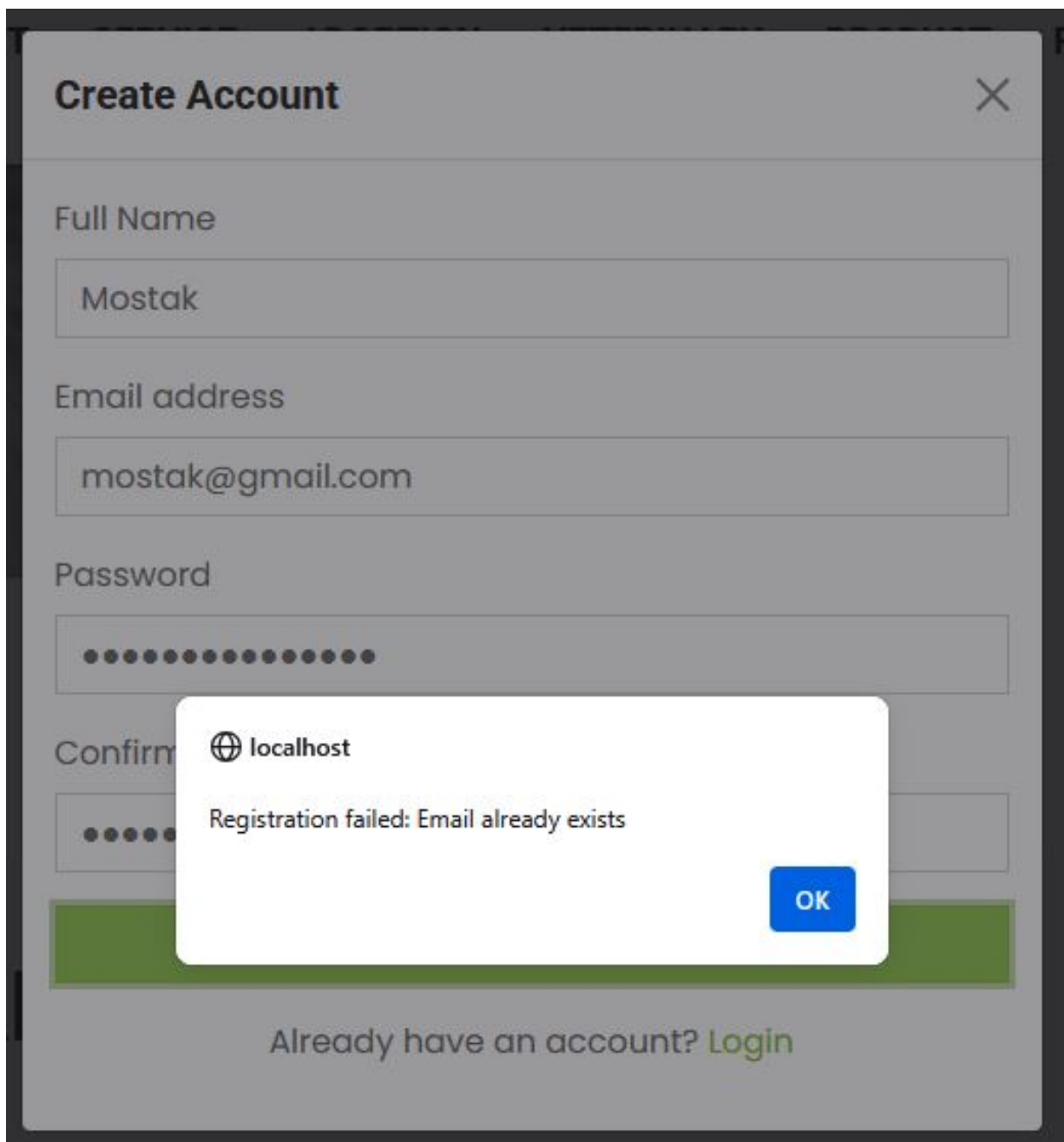


Figure 1.2: Register Existing User - Test Result: Pass.

### 1.5.3 Long Password - Test Result: Fail

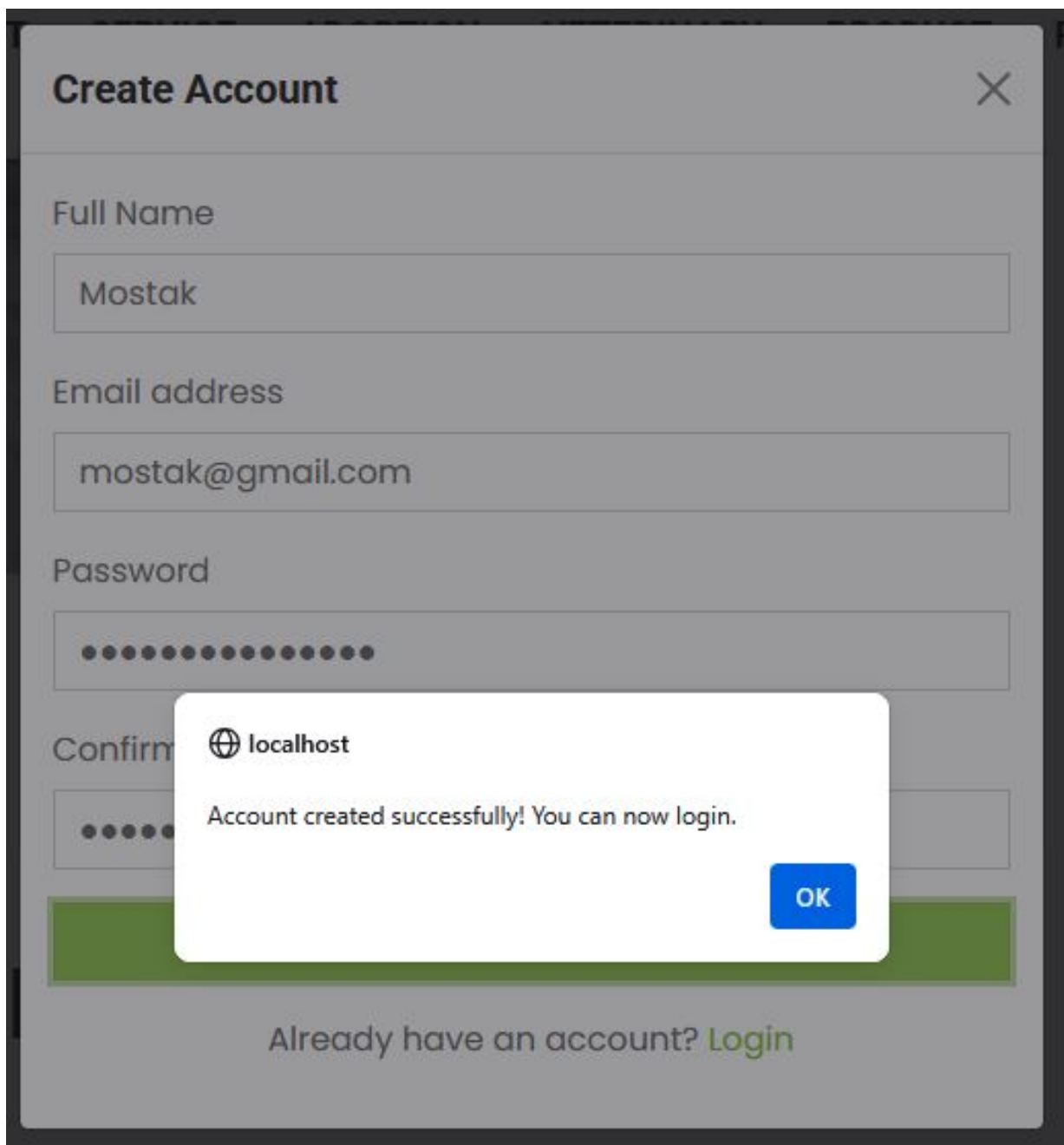


Figure 1.3: Long Password - Test Result: Fail.



#### 1.5.4 Valid Login - Test Result: Pass

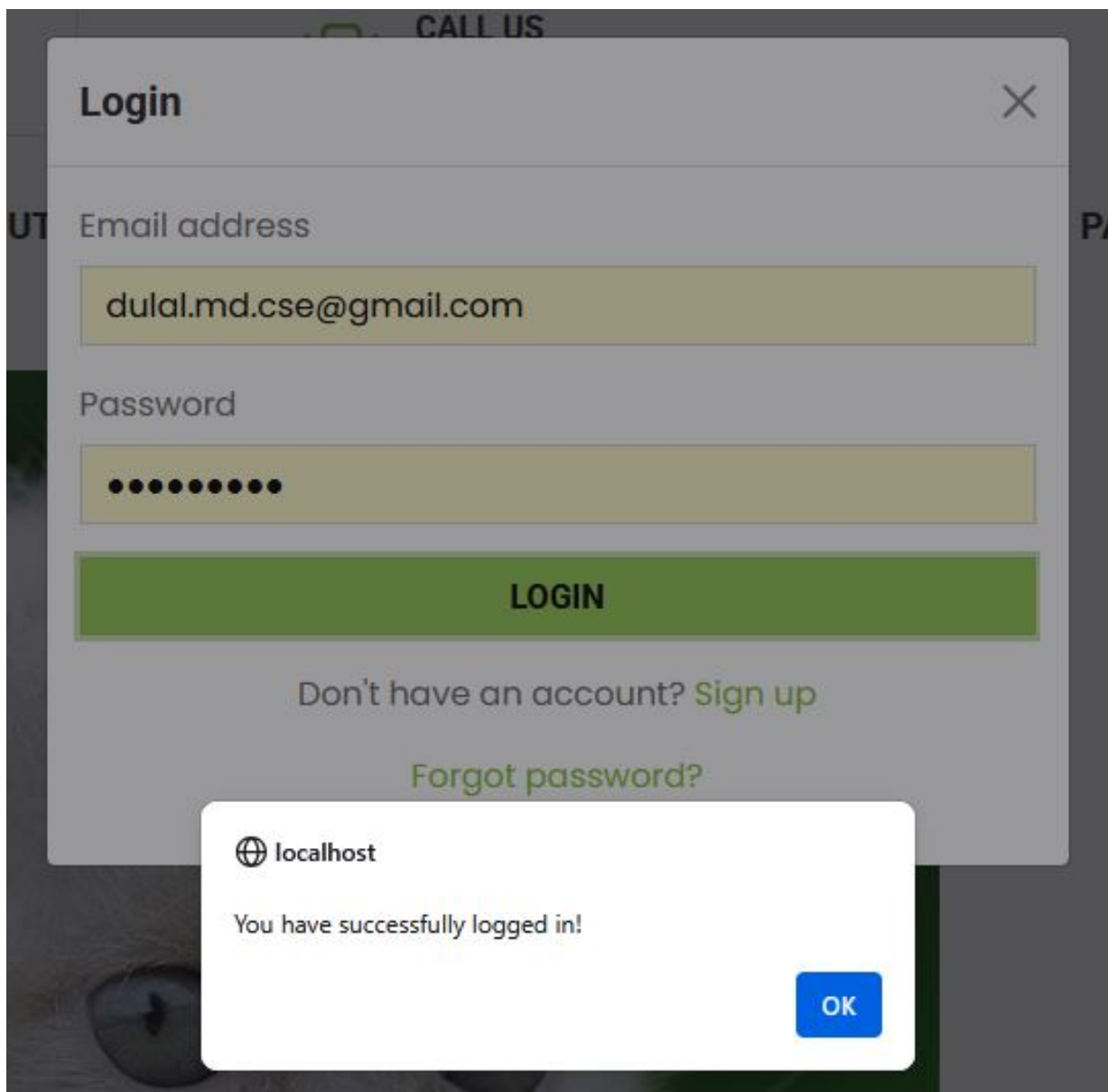


Figure 1.4: Valid Login - Test Result: Pass.

### 1.5.5 Invalid Login - Test Result: Pass

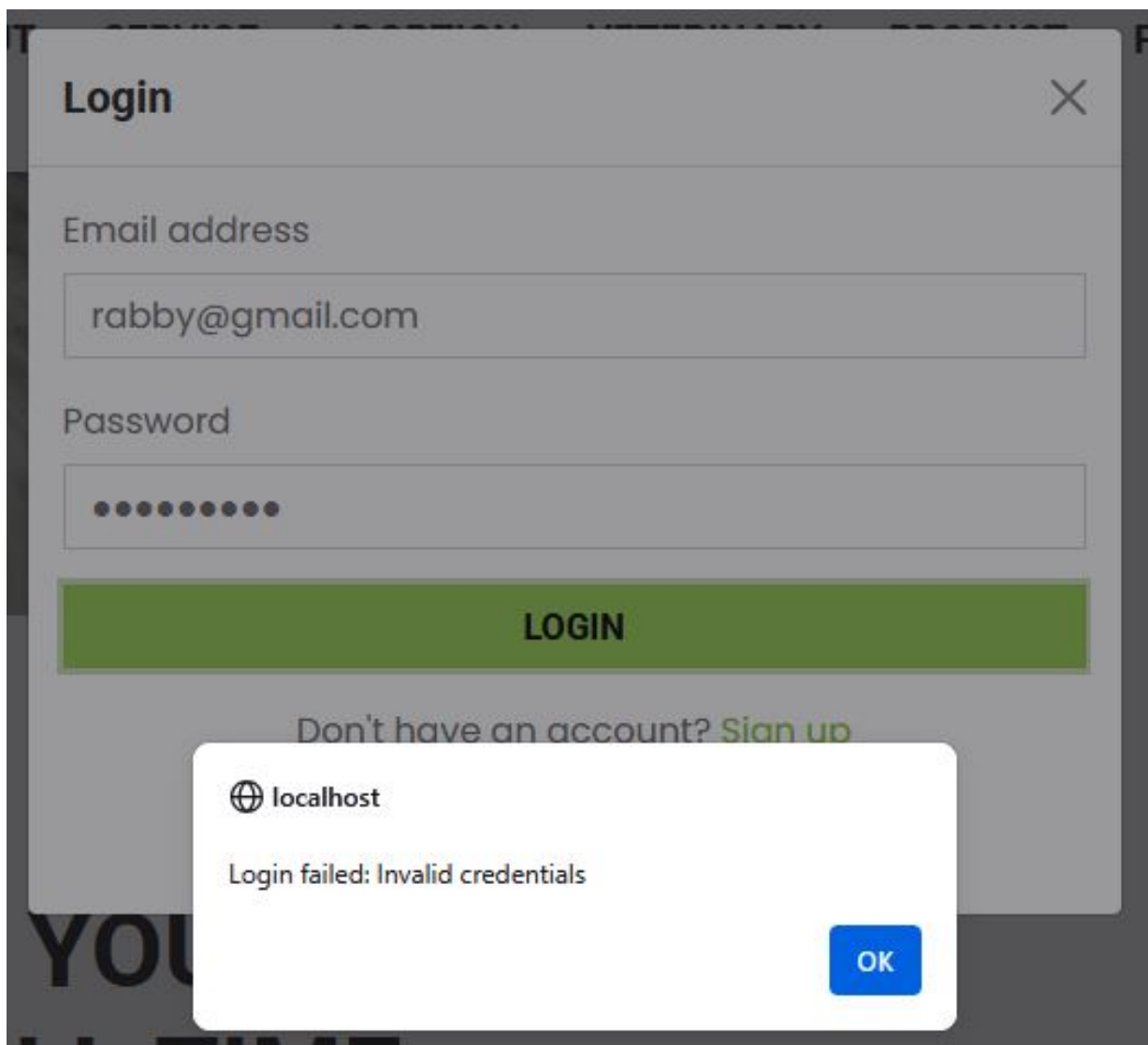
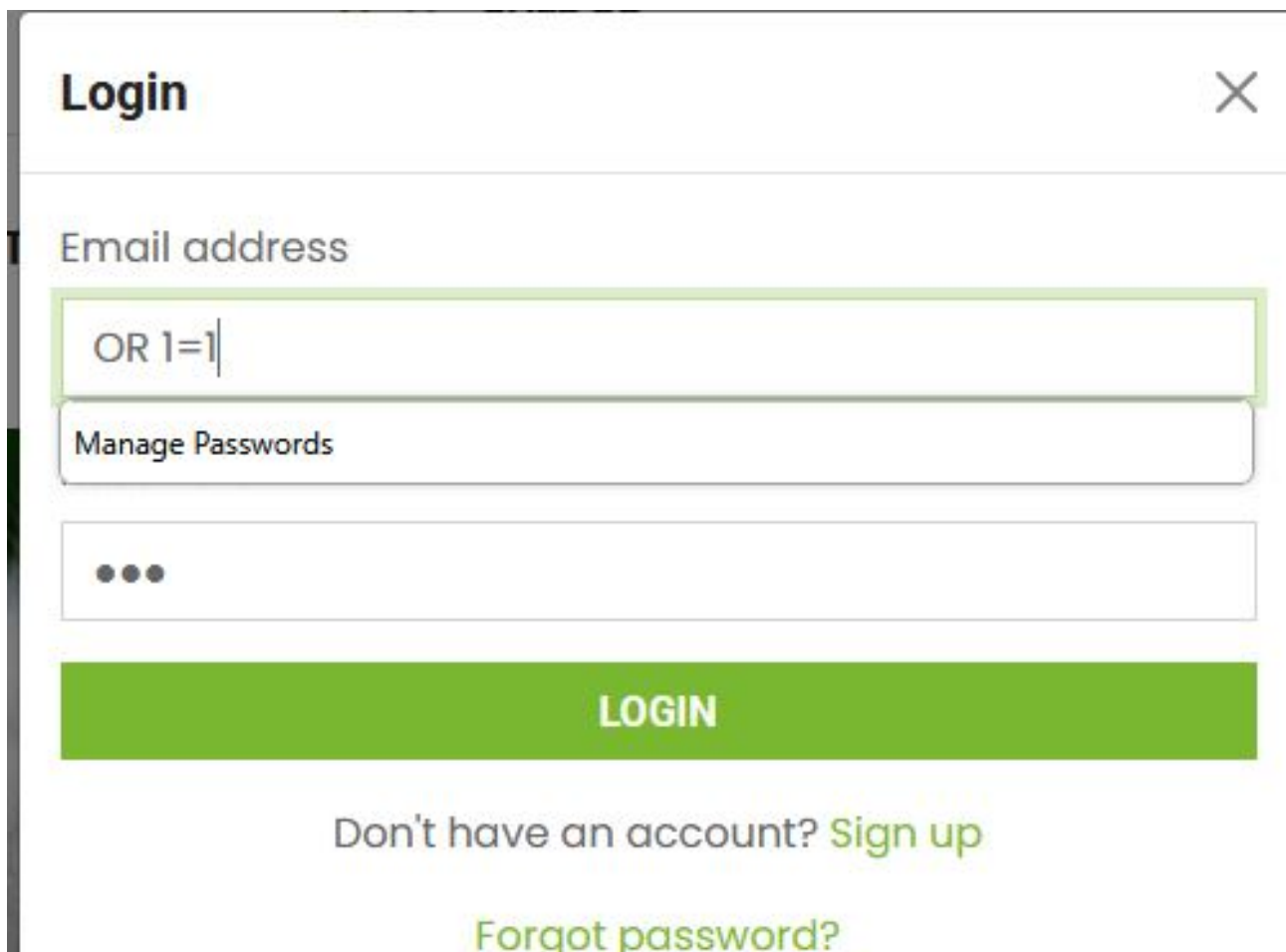


Figure 1.5: Invalid Login - Test Result: Pass.

### 1.5.6 SQL Injection in Filter - Test Result: Pass



The screenshot shows a login interface with the following elements:

- Login** header with a close button (X).
- Email address** label above a text input field containing the text `OR 1=1`.
- Manage Passwords** link below the email field.
- A password input field represented by three dots.
- A green **LOGIN** button.
- Links for [Don't have an account? Sign up](#) and [Forgot password?](#).

Figure 1.6: SQL Injection in Filter - Test Result: Pass.

## 1.6 Analysis and Discussion

The comprehensive testing of the PET Management System revealed both strengths and areas for improvement in the application's functionality, security, and performance. The system successfully handled core operations like user registration, authentication, and data management, with most test cases passing as expected. Security measures proved effective against common threats, demonstrated by successful prevention of SQL injection attacks and unauthorized admin access. However, the testing uncovered critical vulnerabilities in input validation, particularly for password strength requirements and image upload size limits, where the system failed to enforce specified constraints. Performance testing showed the application could handle moderate loads but struggled with extreme cases like loading 5000 pet records within the required time frame. The manual testing approach provided valuable insights into user experience aspects that automated tests might miss, though incorporating automation for regression testing could improve efficiency. The results indicate that while the system meets basic functional requirements, enhancements are needed in input validation, error handling, and performance optimization to ensure reliability under real-world conditions. These improvements would significantly enhance the system's robustness and user satisfaction while maintaining its strong security foundation.