

# Mini Project Report

## Customer Feedback System

Name : J. N. I. B. Nimalarathna  
Reg. No. : 623609603  
Group : 08  
Date : 14.10.2025

# Table of Content

- Introduction ..... 3 Page
- Requirement Analysis..... 4 Page
- Data Modeling..... 5-7 Page
- Detail design..... 8-9 Page
- Test Plan and Results..... 10-13 Page
- Conclusion or Discussion..... 14 Page
- References..... 15 Page
- Appendix..... 16-20 Page

## ● Introduction

This project is designed as a small but comprehensive web-based system for collecting and managing product feedback. The main goal was to actually create something working end to end smoothly, without having any issue or limitations. I make sure to keep project clean by modularizing the projects, maintaining migration for databases, adding small but effective security layer (CSRF protection) and using GIT as a version control for better code maintainability.

The system is built with PHP, MySQL, HTML, and CSS. It is deployed locally via XAMPP. I definitely avoid using heavy external frameworks and libraries. The back end is written in pure PHP with PDO for secure database connectivity, and it's worth the effort. Added CSRF protection to prevent form abuse.

The system consists of two main components:

1. The public sector is where customers can browse products and see and add feedbacks.  
(Customers can only see the approved feedbacks)
2. Admin dashboard is where admin can create/read/update/delete Products,  
view/approve/deactivate /delete feedbacks while admin actions are automatically logged  
in audit table.

All comments remain inactive initially until approved by the administrator. This helps people clean spam and irrelevant messages. Once the feedback has been approved, it will appear on the product page and average rating for specific product will be updated with newly added rating. Average rating will be calculated automatically by the system. The administrator can remove or delete the feedback at any time.

A key component of this project is the audit log system. What does the administrator has done (create/update/delete/approve/disable), database records the action with a timestamp with userId. This provides a layer of transparency and a realistic indication of how modern applications maintain traceability.

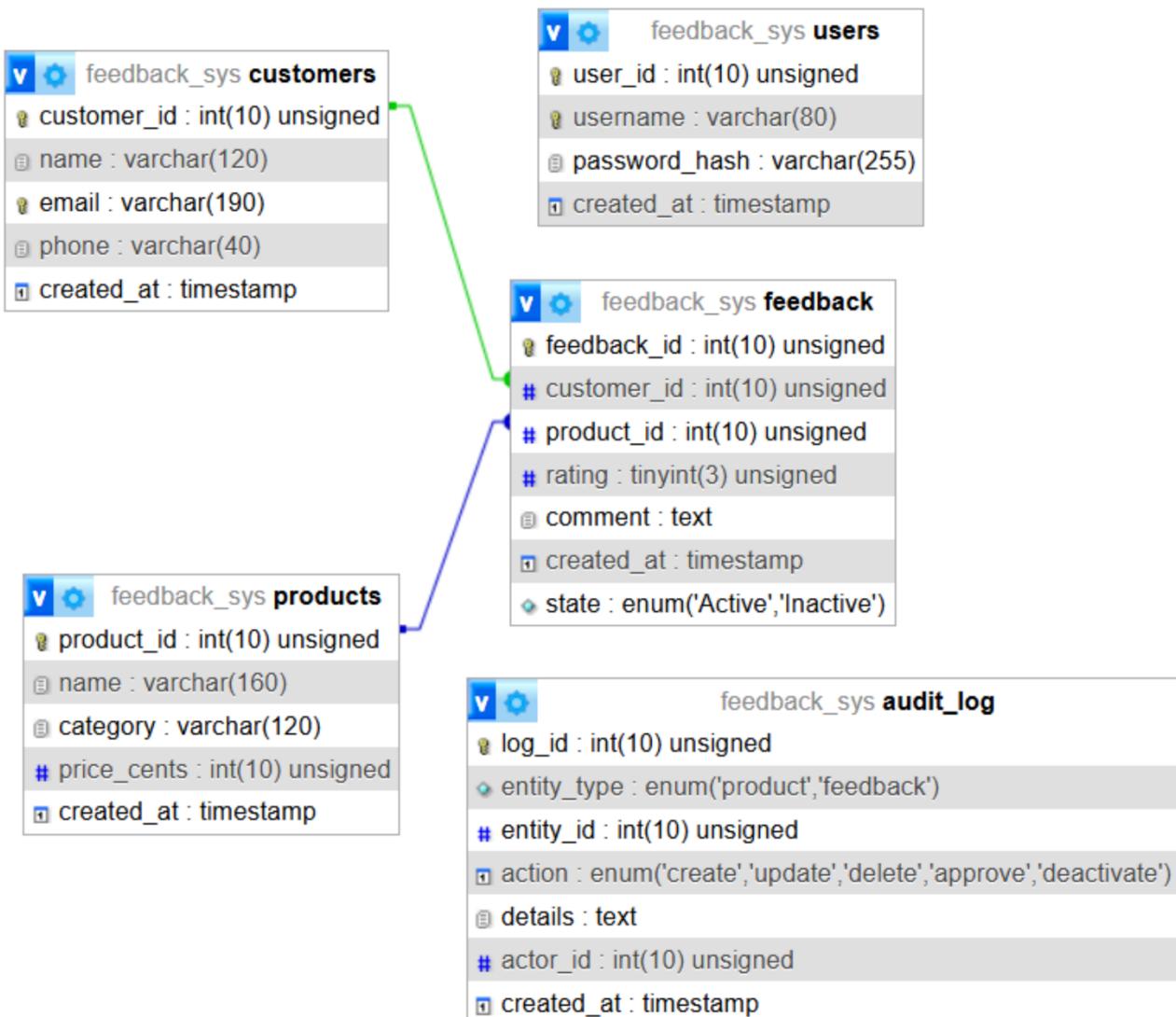
Overall the project helped me connect some concepts. It may look small. But it works like a real product recommendation system.

## • Requirement Analysis



## • Data Modeling

### 1) Relational Data Base



## 2) DATABASE NORMALIZATION ( 3NF )

Customer table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	customer_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(120)	utf8mb4_unicode_ci		No	None		
3	email	varchar(190)	utf8mb4_unicode_ci		No	None		
4	phone	varchar(40)	utf8mb4_unicode_ci		Yes	NULL		
5	created_at	timestamp			No	current_timestamp()		

Feedback table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	feedback_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	customer_id	int(10)		UNSIGNED	No	None		
3	product_id	int(10)		UNSIGNED	No	None		
4	rating	tinyint(3)		UNSIGNED	No	None		
5	comment	text	utf8mb4_unicode_ci		Yes	NULL		
6	created_at	timestamp			No	current_timestamp()		
7	state	enum('Active', 'Inactive')	utf8mb4_unicode_ci		No	Inactive		

Product table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	product_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(160)	utf8mb4_unicode_ci		No	None		
3	category	varchar(120)	utf8mb4_unicode_ci		No	None		
4	price_cents	int(10)		UNSIGNED	No	None		
5	created_at	timestamp			No	current_timestamp()		

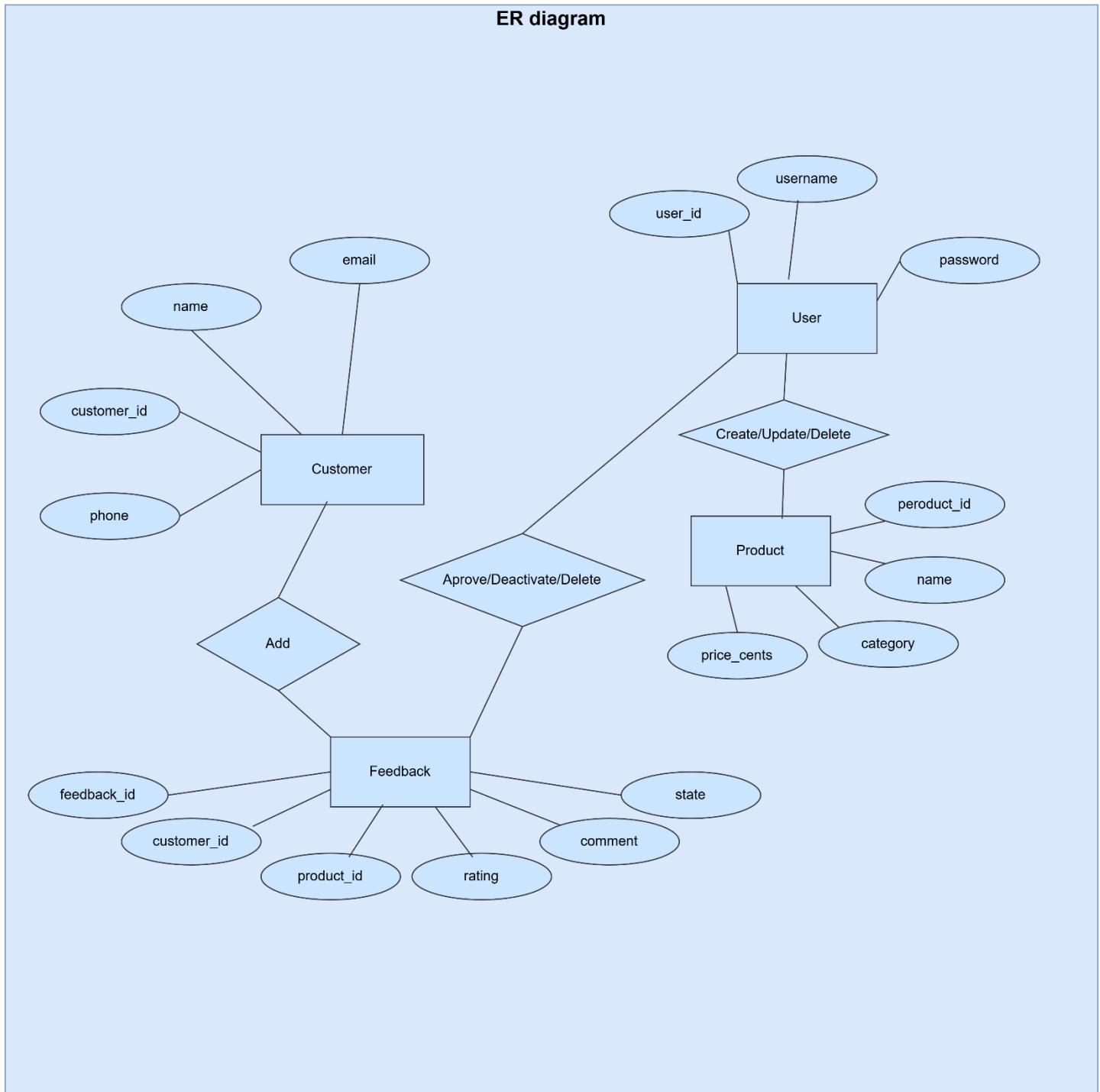
## User table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	user_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	username	varchar(80)	utf8mb4_unicode_ci		No	None		
3	password_hash	varchar(255)	utf8mb4_unicode_ci		No	None		
4	created_at	timestamp			No	current_timestamp()		

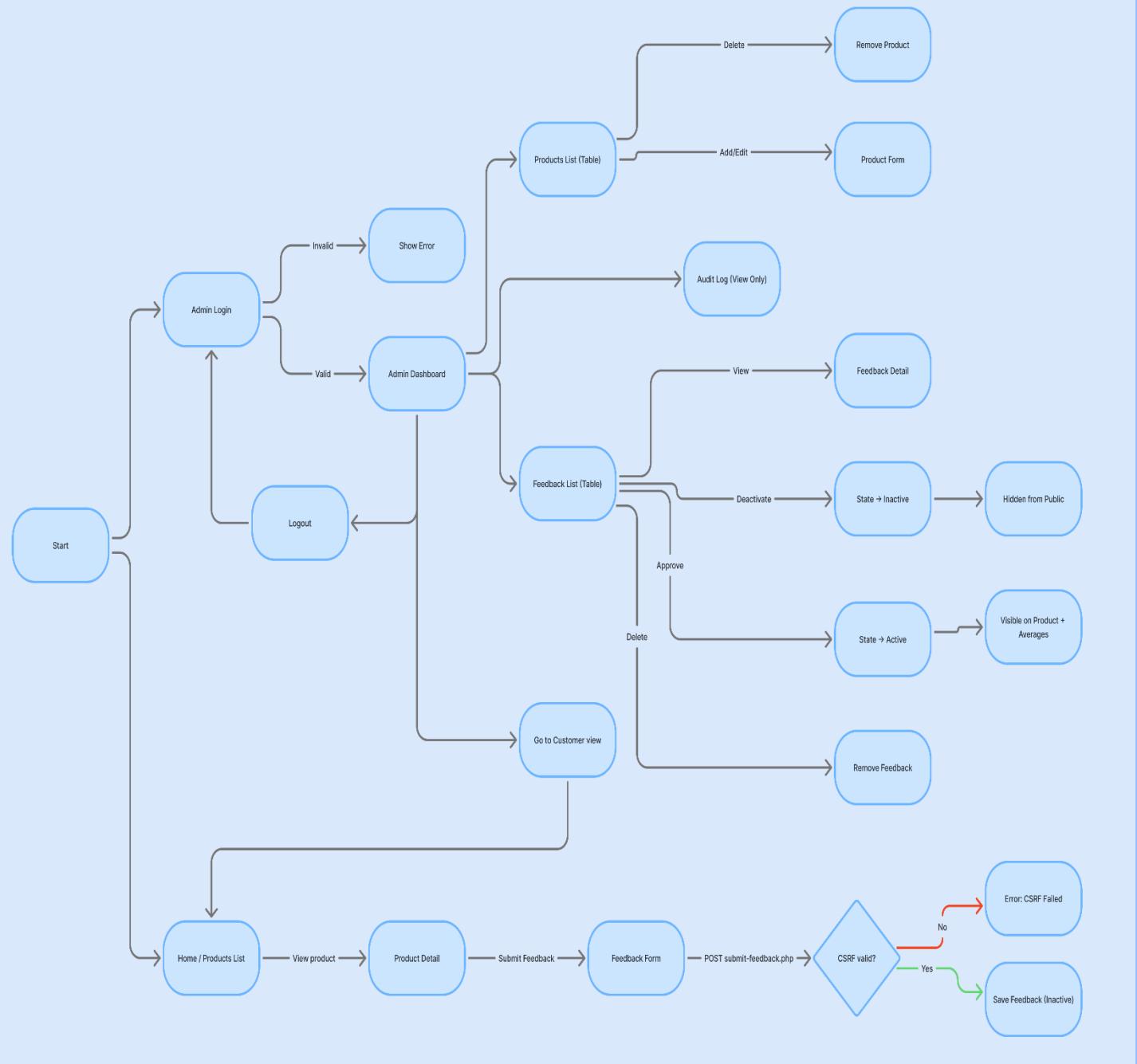
## Audit table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	log_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	entity_type	enum('product', 'feedback')	utf8mb4_general_ci		No	None		
3	entity_id	int(10)		UNSIGNED	No	None		
4	action	enum('create', 'update', 'delete', 'approve', 'de...')	utf8mb4_general_ci		No	None		
5	details	text	utf8mb4_general_ci		Yes	NULL		
6	actor_id	int(10)		UNSIGNED	Yes	NULL		
7	created_at	timestamp			No	current_timestamp()		

## • Detail Design



## Screen Flow Diagram



## • Test Plan and Result

### Test Strategy

This section outlines how the Customer Feedback System was tested and validated during development.

The main goal was simple, to make sure the system works smoothly under real-world use, behaves correctly even under stress, and stays secure and reliable at all times.

Each test focused on different angles of quality, from performance and security to user experience and documentation accuracy.

### System Test

The system test was done to check the entire workflow from end to end, from adding products to submitting and approving feedback. Every form, validation, and process was executed manually to confirm the logic and flow worked exactly as intended. Any errors or unexpected behaviors were fixed immediately until the system ran cleanly without warnings or broken links.

### Performance Test

Here the goal was to push the limits a bit. I tested how the system performed when many feedback entries and products were added, observing how fast pages loaded and how stable the database remained. Even under higher data volume, the system stayed responsive and handled operations smoothly, which shows that the queries and structure were efficient enough for a lightweight deployment.

### Security Test

Security testing focused on preventing unwanted access and protecting user data. I verified that admin pages could not be accessed without proper login, tested for SQL injection attempts, and checked that CSRF tokens worked correctly. All input fields were validated, and all critical actions (like feedback submission and product editing) were safely handled through prepared statements.

### Recovery Test

To test stability, I simulated small failures like disconnecting the database or interrupting form submissions midway to see how the system would react. The result was good, the application showed meaningful error messages and recovered without losing data or crashing the session.

## **Documentation Test**

The documentation part was also reviewed like a mini test. I compared what's written in the report and setup guide with what actually exists in the system. Every feature mentioned from adding feedback to viewing audit logs, was confirmed to work as described. This helped make the project documentation both accurate and self-contained.

## **Beta Test**

Before final submission, I ran a small beta phase where the system was used as if it were in real operation. A few colleagues tested the interface and gave feedback on usability and clarity. Based on that, I made small improvements, like aligning form labels and improving button placement in the admin area.

## **User Acceptance Test**

Finally, the user acceptance test was done to ensure that the system met all requirements and behaved as a ready-to-use solution. This was the stage where I stepped back and used the application like an end user, not as a developer, and confirmed that everything felt functional, intuitive, and complete.

## **QA Environment**

All tests were performed locally in a controlled environment using both Black Box Testing (validating behavior from the user's perspective) and White Box Testing (reviewing the internal PHP logic, queries, and trigger behaviors). Together, these methods ensured both the outer flow and inner code quality were fully verified.

## **Scope**

Test functional and administrative features of the Customer Feedback System.

- Admin login
- Reset admin password
- Admin add/edit/delete products
- Users submit feedback
- Admin approves/deactivates/Delete feedbacks.
- User only see approved feedbacks
- Automatically calculate average rating using approved feedbacks

## Environment

- Windows 11
- XAMPP (PHP 8.2.12, MariaDB 10.4.32)
- Microsoft Edge
- running locally at <http://localhost/customer-feedback-system/>

Area	Result	Comments
<b>Public Functions</b>	<input checked="" type="checkbox"/> Pass	Products and feedback display as expected. Only approved feedback visible.
<b>Feedback Submission</b>	<input checked="" type="checkbox"/> Pass	New feedback saved as <i>Inactive</i> . Admin approval updates visibility instantly.
<b>Admin Login</b>	<input checked="" type="checkbox"/> Pass	Login and session persistence working correctly.
<b>Product CRUD</b>	<input checked="" type="checkbox"/> Pass	Create, update, and delete operations work. Audit logs auto-recorded.
<b>Feedback Moderation</b>	<input checked="" type="checkbox"/> Pass	Approve, deactivate, delete all reflected correctly. State toggles, logged.
<b>Audit Logging</b>	<input checked="" type="checkbox"/> Pass	Triggers record all actions automatically under correct admin ID.
<b>Search Function</b>	<input checked="" type="checkbox"/> Pass	Keyword filtering by product, customer, or comment works correctly.
<b>Security (CSRF/XSS)</b>	<input checked="" type="checkbox"/> Pass	CSRF tokens validated; no script injection possible.
<b>UI/UX Layout</b>	<input checked="" type="checkbox"/> Pass	Admin tables responsive, cards centered correctly.

## Observations

- Feedback averages only include approved (Active) feedback.
- Audit triggers correctly record create/update/delete/approve/deactivate actions with actor\_id.
- Inactive feedback never shown publicly.
- CSRF and session isolation verified manually by attempting invalid POSTs.
- UI consistent across product and feedback modules

## ● Conclusion or Discussion

Working on this customer feedback system was more valuable than I expected. Idea looks like simple, But when I started connecting everything. Database relationships and PHP backend logics and security, it started to feel like a real application that could actually be used in production. This is what I wanted to do, making a project that functional, secure, complete and maintainable.

While building this, I faced some challenges. Properly handling the active and passive states of feedback requires ability to debug through business logics, especially when calculating product average ratings based on approved feedbacks. Here I'm implementing automatic logging using database triggers at the database level. It's a step towards production level software design.

Another important lesson is security and data normalization. I made sure the database followed the third normal form (3NF). This reduces redundancy and makes queries cleaner and faster. I also implemented a prepared statement with PDO to protect against SQL injection, and I used CSRF tokens to secure forms. These small things will not visible from the outside but They make a big difference in how reliable the system.

On the front-end I kept things minimal. Unstructured HTML and pure CSS. Because I want to focus on logic rather than design. Still, I made sure the layout was consistent. The admin dashboard has proper tables, the feedback forms are responsive, and the footer and header stay fixed for a full-screen experience.

Ultimately, the project achieved all of its goals. Product can be managed, customers can give feedbacks, Admin can review and control what is displayed to the public. It is reliable, secure, and easy to understand. More importantly, it helped me practice building entire systems from scratch.

If I expand this project in the future, I will implement image upload feature for products and send emails for new feedback. Even in its current state, it felt complete, professional, and a solid foundation for any similar system.

## ● References

Repository - [GIT](#)

<https://stackoverflow.com>

[W3Schools Online Web Tutorials](#)

<https://www.geeksforgeeks.org/>

## • Appendix

### Products page (Customer View)

The screenshot shows a web browser window with multiple tabs open. The active tab displays the 'Customer Feedback System' products page. The page has a blue header bar. Below it, the word 'Products' is centered in bold black font. Two product cards are displayed in a grid:

- Wireless Mouse**  
Category: Accessories  
★ 4.0 (1)  
LKR 5,999.99  
[View](#)
- USB-C Charger 65W**  
Category: Power  
★ — (0)  
LKR 899.99

At the bottom of the page, a grey footer bar contains the text "© 2025 Customer Feedback System — All rights reserved."

### Product page (Customer View) – Product details, Add review form, View recent reviews

The screenshot shows a web browser window with multiple tabs open. The active tab displays the 'Customer Feedback System' product page for a 'Wireless Mouse'. The page has a blue header bar. A back button is visible on the left. The product details are shown in a card:

- Wireless Mouse**  
Category: Accessories  
★ 4.0 (1)  
LKR 5,999.99

Below the product details is a section titled 'Add your feedback' containing four input fields:

- Name
- Email
- Phone (optional)
- Rating

At the bottom of the page, a grey footer bar contains the text "© 2025 Customer Feedback System — All rights reserved."

Customer Feedback System

### Add your feedback

Name

Email

Phone (optional)

Rating  
 Choose...

Comment (optional)

© 2025 Customer Feedback System — All rights reserved.

Customer Feedback System

Phone (optional)

Rating  
 Choose...

Comment (optional)

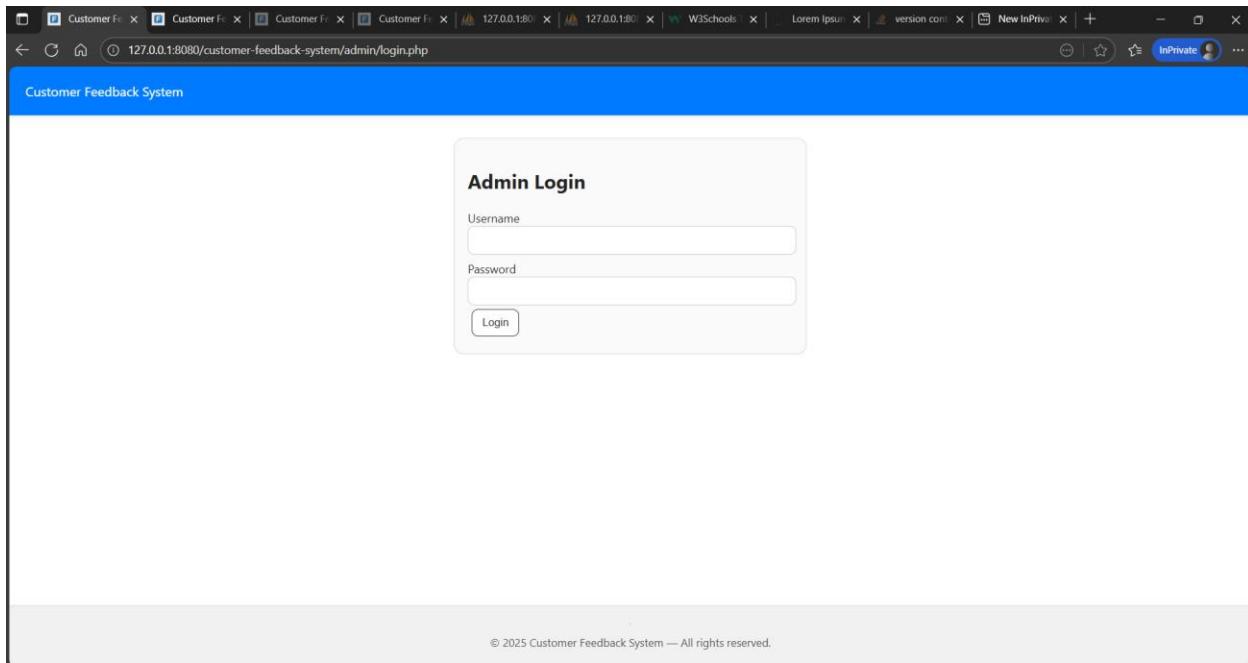
### Recent feedback

Nadun Nimalarathna · ★ 4 - 2025-10-14 21:19:54

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

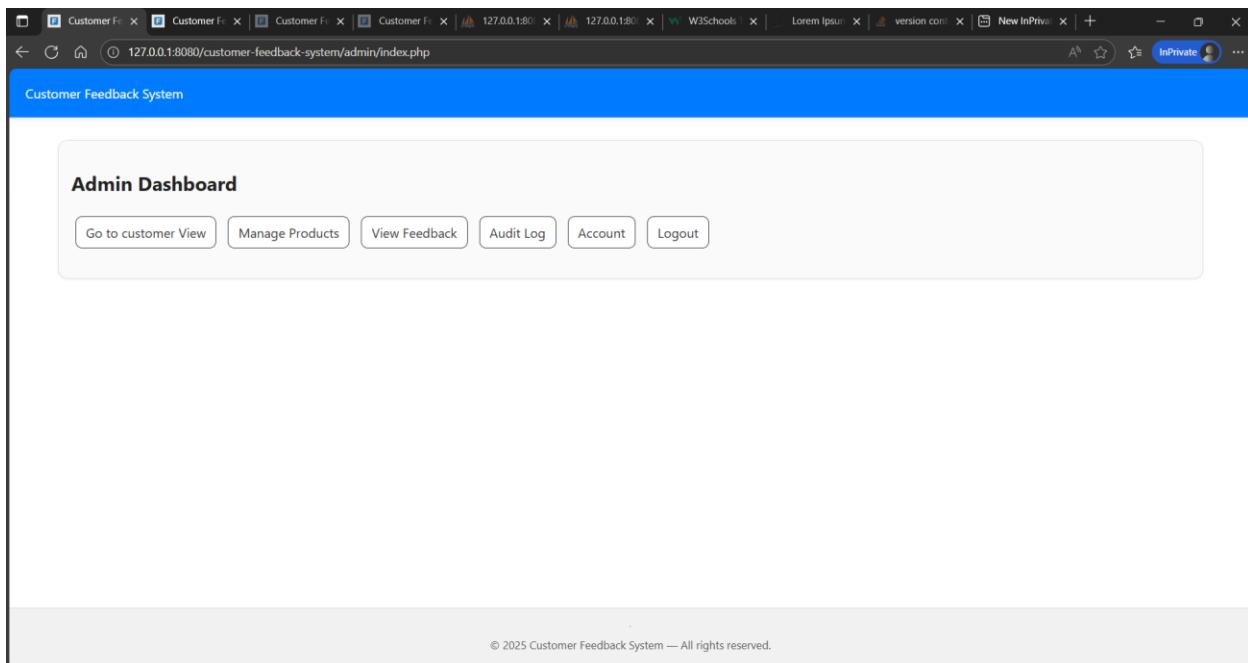
© 2025 Customer Feedback System — All rights reserved.

## Login (Admin Module)



A screenshot of a web browser showing the 'Customer Feedback System' Admin Login page. The URL in the address bar is 127.0.0.1:8080/customer-feedback-system/admin/login.php. The page has a blue header bar with the text 'Customer Feedback System'. Below it is a white content area with a rounded rectangle containing the 'Admin Login' form. The form includes fields for 'Username' and 'Password', and a 'Login' button. At the bottom of the content area, there is a grey footer bar with the copyright text '© 2025 Customer Feedback System — All rights reserved.'

## Admin dashboard



A screenshot of a web browser showing the 'Customer Feedback System' Admin Dashboard page. The URL in the address bar is 127.0.0.1:8080/customer-feedback-system/admin/index.php. The page has a blue header bar with the text 'Customer Feedback System'. Below it is a white content area with a rounded rectangle containing the 'Admin Dashboard' section. This section features several buttons: 'Go to customer View', 'Manage Products', 'View Feedback', 'Audit Log', 'Account', and 'Logout'. At the bottom of the content area, there is a grey footer bar with the copyright text '© 2025 Customer Feedback System — All rights reserved.'

## Product management page - CRUD

The screenshot shows a web browser window titled "Customer Feedback System" with the URL "127.0.0.1:8080/customer-feedback-system/admin/products.php". The page has a blue header bar with the title. Below it is a white content area. On the left, there's a sidebar with a "Products" section and a "+ Add Product" button. The main content area displays a table with four columns: "Name", "Category", "Price", and "Created". There are three rows of data:

Name	Category	Price	Created
Wireless Mouse	Accessories	LKR 5,999.99	2025-10-11 14:38:07
Mechanical Keyboard	Accessories	LKR 13,999.99	2025-10-11 14:38:07
USB-C Charger 65W	Power	LKR 899.99	2025-10-11 14:38:07

Each row has two buttons on the right: "Edit" and "Delete". At the bottom of the content area, there's a copyright notice: "© 2025 Customer Feedback System — All rights reserved."

## Feedbacks page (Admin view) - CRUD

The screenshot shows a web browser window titled "Customer Feedback System" with the URL "127.0.0.1:8080/customer-feedback-system/admin/feedback.php". The page has a blue header bar with the title. Below it is a white content area. On the left, there's a sidebar with a "Feedback" section and a "Search product / customer / comment" input field. The main content area displays a table with eight columns: "Product", "Customer", "Email", "Rating", "Comment", "State", "Date", and "Actions". There is one row of data:

Product	Customer	Email	Rating	Comment	State	Date	Actions
Wireless Mouse	Nadun Nimalarathna	nadunisurubuddhika@gmail.com	★ 4	<p>Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 150...</p>	Active	2025-10-14 21:19:54	<a href="#">View</a> <a href="#">Deactivate</a> <a href="#">Delete</a>

At the bottom of the content area, there's a copyright notice: "© 2025 Customer Feedback System — All rights reserved."

## Action audit log

The screenshot shows a web browser window for the 'Customer Feedback System'. The title bar includes multiple tabs for 'Customer Feedback' and 'Customer Feedback System'. The main content area has a blue header bar with the text 'Customer Feedback System'. Below this is a white panel titled 'Audit Log'. At the top of the panel are three dropdown menus: 'Type' (set to 'All'), 'Action' (set to 'All'), and a search input field containing 'name / rating / "from"'. To the right of these are 'Filter' and 'Clear' buttons. The main body of the panel contains a table with the following data:

When	Type	Entity	Action	Details	Actor
2025-10-14 21:39:29	feedback	#9	approve	{"from": "Inactive", "to": "Active"} {"product_id": 2, "rating": 3}	1
2025-10-14 21:36:07	feedback	#4	delete	{"from": "Active", "to": "Inactive"}	1
2025-10-14 21:36:03	feedback	#9	deactivate	{"from": "Active", "to": "Inactive"}	1
2025-10-14 21:21:19	feedback	#9	approve	{"from": "Inactive", "to": "Active"}	1
2025-10-14 21:21:17	feedback	#4	approve	{"from": "Inactive", "to": "Active"}	1
2025-10-14 21:19:54	feedback	#9	create	{"product_id": 1, "customer_id": 1, "rating": 4, "state": "Inactive"}	—
2025-10-14 21:05:46	feedback	#4	deactivate	{"from": "Active", "to": "Inactive"}	1
2025-10-14 21:04:30	product	#2	update	{"changed": true}	1

At the bottom of the panel, there is a footer bar with the text '© 2025 Customer Feedback System — All rights reserved.'

**END**