National University of Modern Languages (NUML)

Group Members:	M. Haider Ali (3347) M. Ashar Yousuf (3345) M. Abdullah (3342) Eman Mudassir (3358)
Class:	BSCS 4-A (Morning)
Course-Title:	Advance Programming
Project-Title:	Lost & Found Management System
Submitted to:	Mr. Ahtesham

Table of Contents

Project Report 3
1. Executive Summary 3
2. System Architecture & Interaction Design 3
2.1. Technology Stack 3
2.2. Use Case Analysis 3
Figure 1: Use Case Diagram 4
Explanation: 4
3. Database Design & Data Models 4
3.1. Entity Relationship Diagram (ERD) 5
Figure 2: Entity Relationship Diagram 5
Explanation: 5
4. Detailed Modules & Functionalities 5
4.1. User Identity Module 5
4.2. Item Reporting & Gallery 6
4.3. Claiming System & Workflow 6
Figure 3: Claim Lifecycle Timing Diagram 6
Explanation: 6
4.4. Admin Module 6
5. User Interface 7
5.1. Homepage 7
5.2. Found Items 7
5.3. User Claims 8
5.4. User Profile 8
5.5. Admin 9
5.6. Admin (Managing Found items) 9
5.7. Admin (Managing Claims) 10
5.8. Admin (Managing Users) 10
5.9. Admin (Contact messages) 11
6. Security Features 11
7. Conclusion 11

Project Report

1. Executive Summary

It is a robust, web-based Lost and Found Management System designed to digitize and streamline the process of recovering lost belongings within an organization, such as a university campus. Traditional methods like physical lost-and-found boxes or notice boards are often inefficient and lack transparency. Project addresses these issues by providing a centralized, secure, and accessible platform where users can report found items, search for lost belongings, and initiate claims.

Developed using the ASP.NET Core MVC framework, the system ensures a responsive user experience, secure data handling, and administrative oversight. It features a complete workflow from item reporting to final reclamation, supported by a dedicated Admin Panel for moderation.

Youtube Video Link: https://youtu.be/eCPVXcoFauE?si=d8QYY_IBI4B8unIg

2. System Architecture & Interaction Design

The project is built using a modern tiered architecture, ensuring separation of concerns between data, logic, and presentation.

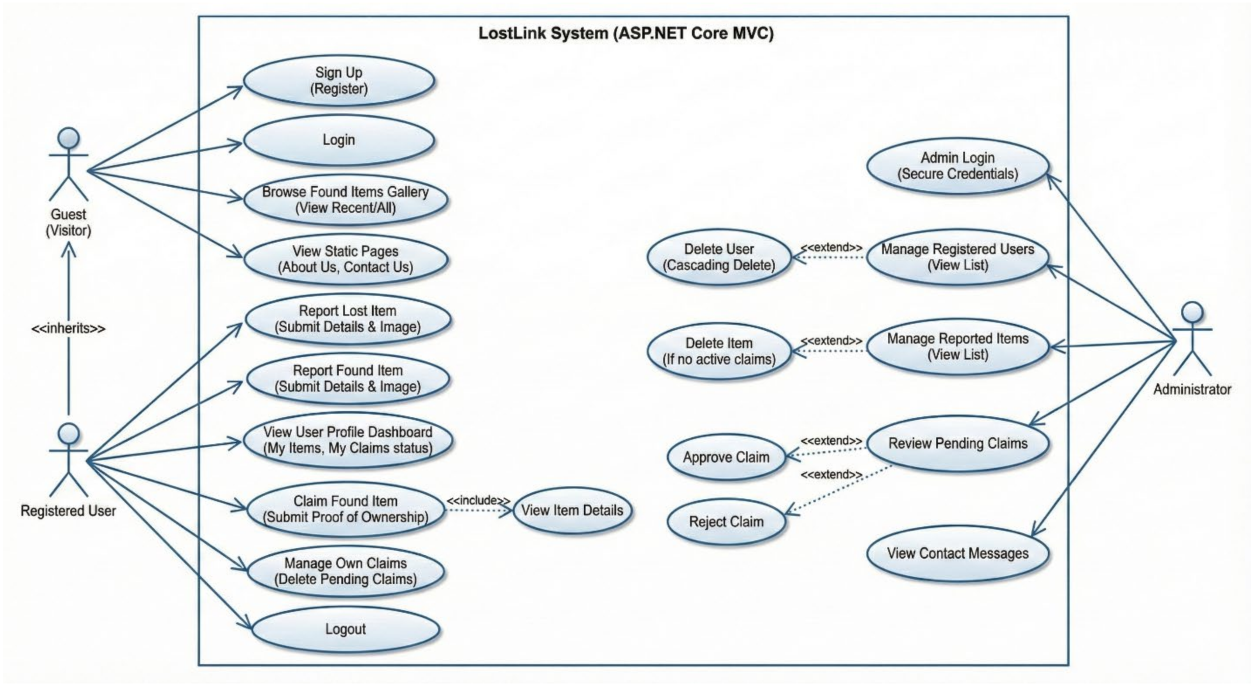
2.1. Technology Stack

- **Framework:** ASP.NET Core MVC (Model-View-Controller).
- **Language:** C#.
- **Frontend:** Razor Views (.cshtml), Bootstrap 5, FontAwesome.
- **Database:** Entity Framework Core (EF Core) utilizing a relational database (SQL Server/LocalDB).
- **Authentication:** Cookie-based Authentication with custom session management ("MyCookieAuth").
- **API:** RESTful API endpoints for external data access.

2.2. Use Case Analysis

The system divides functionality among three primary actors: Guests, Registered Users, and Administrators.

Figure 1: Use Case Diagram


LostLink System (ASP.NET Core MVC)
Guest (Visitor)
<<inherits>>
Registered User
Administrator
Sign Up (Register)
Login
Browse Found Items Gallery (View Recent/All)
View Static Pages (About Us, Contact Us)
Report Lost Item (Submit Details & Image)
Report Found Item (Submit Details & Image)
View User Profile Dashboard (My Items, My Claims status)
Claim Found Item (Submit Proof of Ownership)
<<include>>
View Item Details
Manage Own Claims (Delete Pending Claims)
Logout
Admin Login (Secure Credentials)
Delete User (Cascading Delete)
<<extend>>
Manage Registered Users (View List)
Delete Item (If no active claims)
<<extend>>
Manage Reported Items (View List)
Approve Claim
<<extend>>
Review Pending Claims
<<extend>>
Reject Claim
View Contact Messages

Explanation:

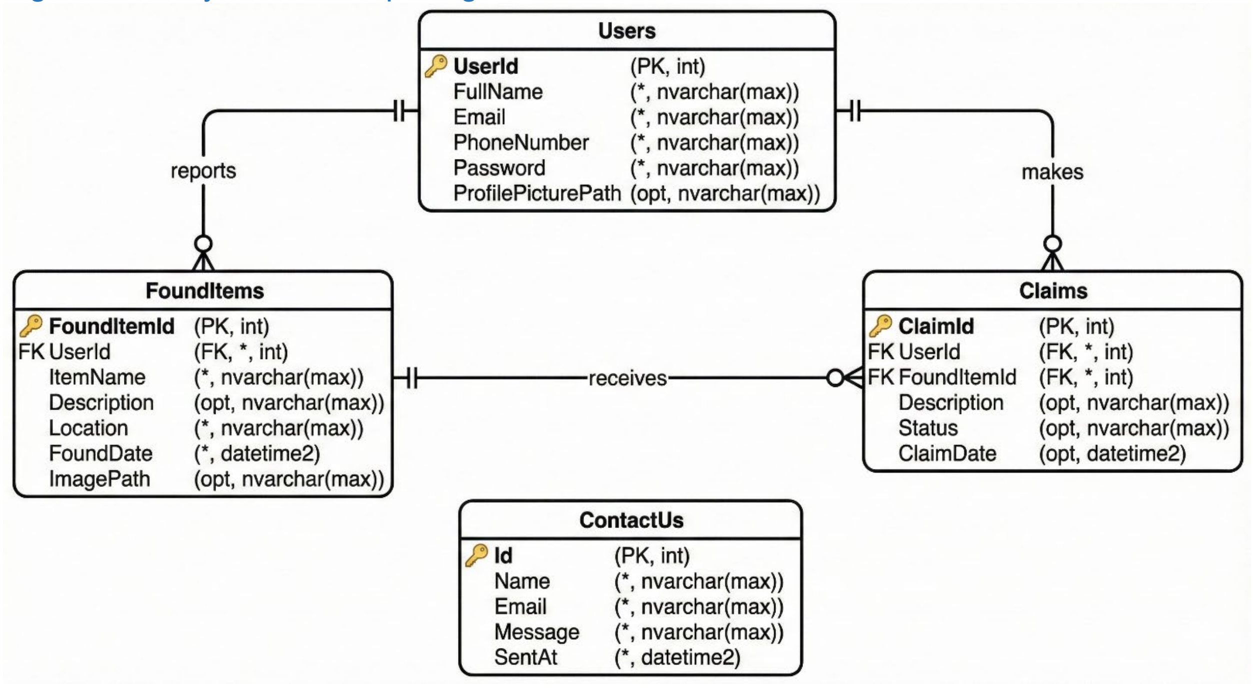
- **Guests:** Can browse the gallery and access static pages (About Us, Contact Us) but cannot interact with data.
- **Registered Users:** Inherit all guest privileges. They are uniquely authorized to Report Items (both lost and found), Claim Items found by others, and manage their personal Profile Dashboard.
- **Administrators:** Possess elevated privileges to moderate content. Key functions include the Cascading Delete of users (removing their associated data) and the adjudication of claims (Approve/Reject).

3. Database Design & Data Models

The system uses Entity Framework Core to manage the database via the ApplicationDbContext class. The schema is normalized to ensure data integrity and efficient querying.

3.1. Entity Relationship Diagram (ERD)

Figure 2: Entity Relationship Diagram


Users
UserId (PK, int)
FullName (*, nvarchar(max))
Email (*, nvarchar(max))
PhoneNumber (*, nvarchar(max))
Password (*, nvarchar(max))
ProfilePicturePath (opt, nvarchar(max))
reports
makes
FoundItems
FoundItemId (PK, int)
FK UserId (FK, *, int)
ItemName (*, nvarchar(max))
Description (opt, nvarchar(max))
Location (*, nvarchar(max))
FoundDate (*, datetime2)
ImagePath (opt, nvarchar(max))
receives
Claims
ClaimId (PK, int)
FK UserId (FK, *, int)
FK FoundItemId (FK, *, int)
Description (opt, nvarchar(max))
Status (opt, nvarchar(max))
ClaimDate (opt, datetime2)
ContactUs
Id (PK, int)
Name (*, nvarchar(max))
Email (*, nvarchar(max))
Message (*, nvarchar(max))
SentAt (*, datetime2)

Explanation:

- **Users Table:** The central entity. It has a One-to-Many relationship with FoundItems (a user can report multiple items) and Claims (a user can claim multiple items).
- **FoundItems Table:** Linked to the User who reported it. It acts as the parent entity for Claims.
- **Claims Table:** This is a crucial associative entity connecting a User (the claimant) to a FoundItem. It tracks the lifecycle of a specific transaction request via the Status field.
- **ContactUs:** A standalone table for storing user inquiries, decoupled from the authentication system.

4. Detailed Modules & Functionalities

4.1. User Identity Module

- **Registration:** The SignupController handles new user registration, supporting profile picture uploads to the server's wwwroot directory.
- **Authentication:** The LoginController creates a secure identity. A custom **Session Token** mechanism prevents stale session persistence after server restarts by checking ActiveServerSessions in memory.

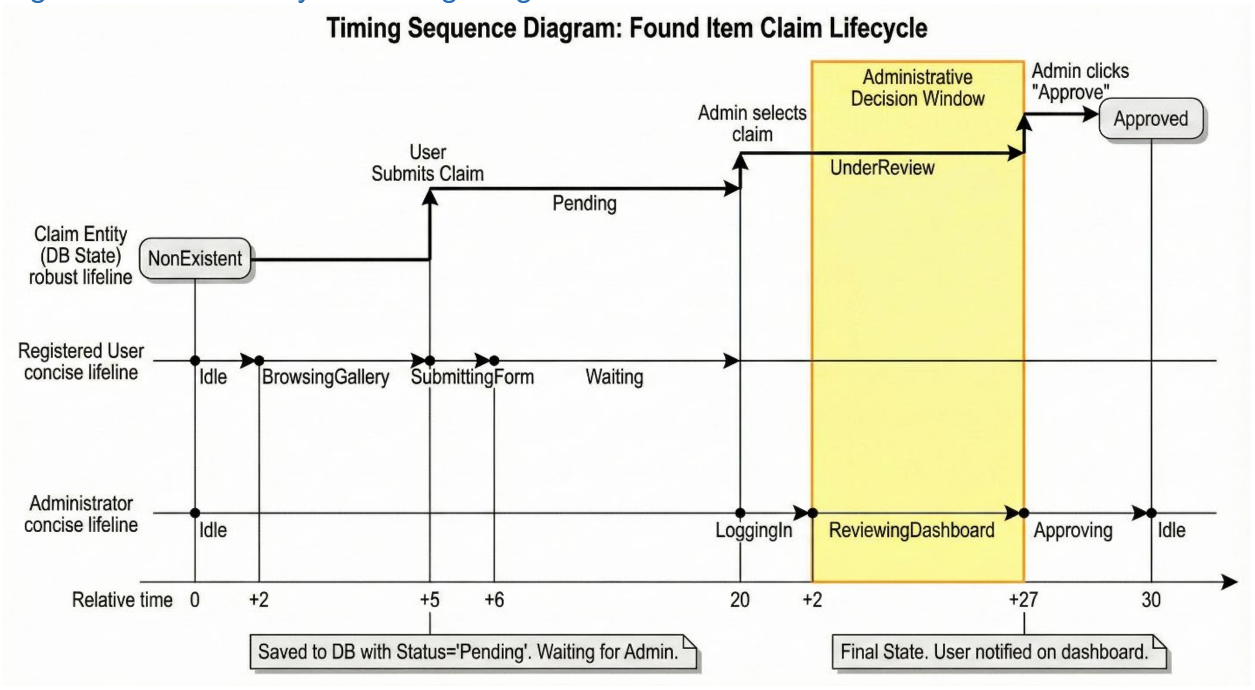
4.2. Item Reporting & Gallery

- **Reporting:** Authorized users access Found.cshtml to submit item details. Server-side validation ensures data integrity.
- **Gallery:** HomeController.Index displays the 6 most recent items, while Gallery.cshtml provides a full archive.

4.3. Claiming System & Workflow

The core business logic resides in the claiming process. A user identifies an item, submits a claim, and waits for administrative approval.

Figure 3: Claim Lifecycle Timing Diagram



Timing Sequence Diagram: Found Item Claim Lifecycle
Administrative Decision Window
Admin clicks "Approve"
Approved
Admin selects claim
UnderReview
User Submits Claim
Pending
Claim Entity (DB State) robust lifeline
NonExistent
Registered User concise lifeline
Idle
BrowsingGallery
SubmittingForm
Waiting
Administrator concise lifeline
Idle
LoggingIn
ReviewingDashboard
Approving
Idle
Relative time
0
+2
+5
+6
20
+2
+27
30
Saved to DB with Status='Pending'. Waiting for Admin.
Final State. User notified on dashboard.

Explanation:

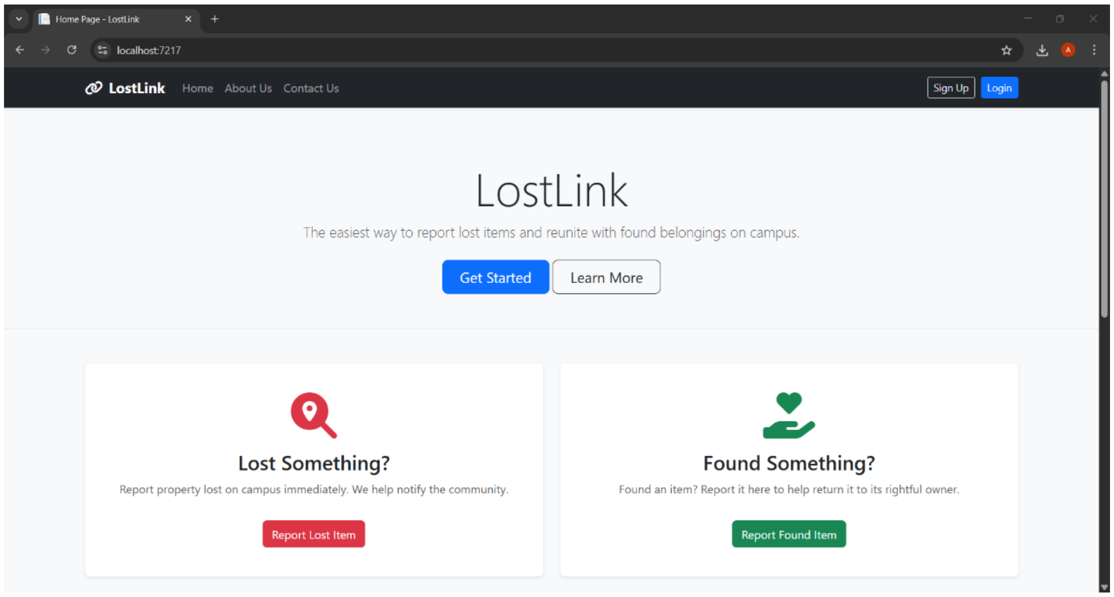
1. **Submission (T+5):** The User transitions the claim state from NonExistent to Pending by submitting the form.
2. **Waiting Period:** The claim remains Pending until an Administrator logs in.
3. **Review (T+22):** The Admin enters the "Decision Window," reviewing the proof of ownership against the item description.
4. **Resolution (T+27):** The Admin transitions the state to Approved (or Rejected), effectively closing the workflow.

4.4. Admin Module

- **User Management:** Admins can view and delete users. Deletion logic cascades to remove associated claims, preventing orphaned records.
- **Claim Adjudication:** The "All Claims" view allows Admins to inspect pending claims and toggle their status.

5. User Interface

5.1. Homepage



LostLink

The easiest way to report lost items and reunite with found belongings on campus.

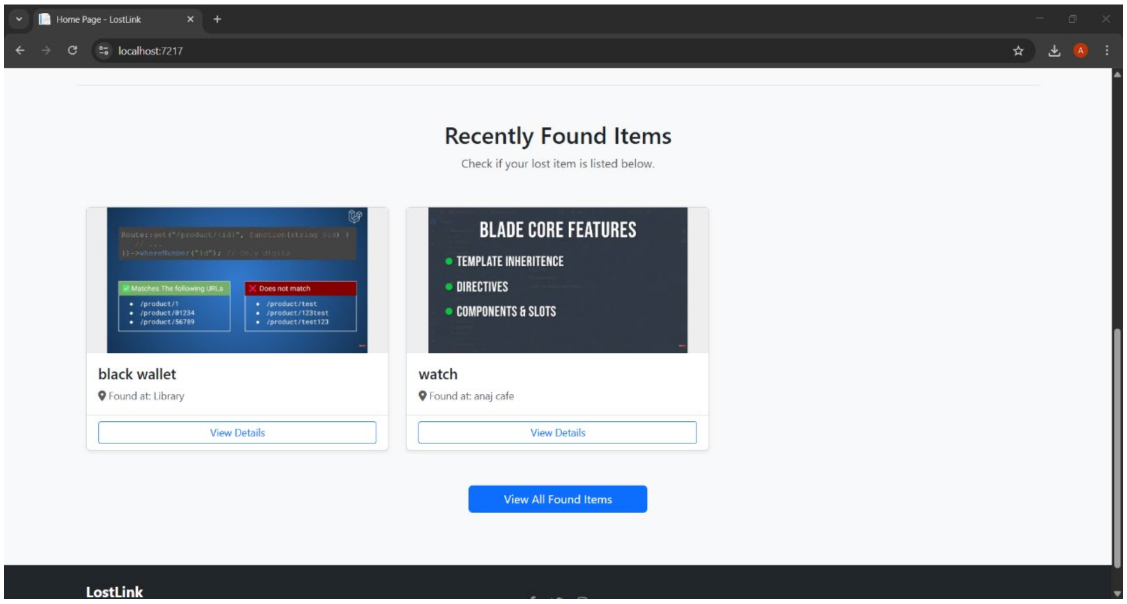
Lost Something?

Report property lost on campus immediately. We help notify the community.

Found Something?

Found an item? Report it here to help return it to its rightful owner.

5.2. Found Items



Recently Found Items

Check if your lost item is listed below.

```
Route::get("/product/{id}", function(string $id) {
    // ...
})->whereNumber("id"); // Only digits
```

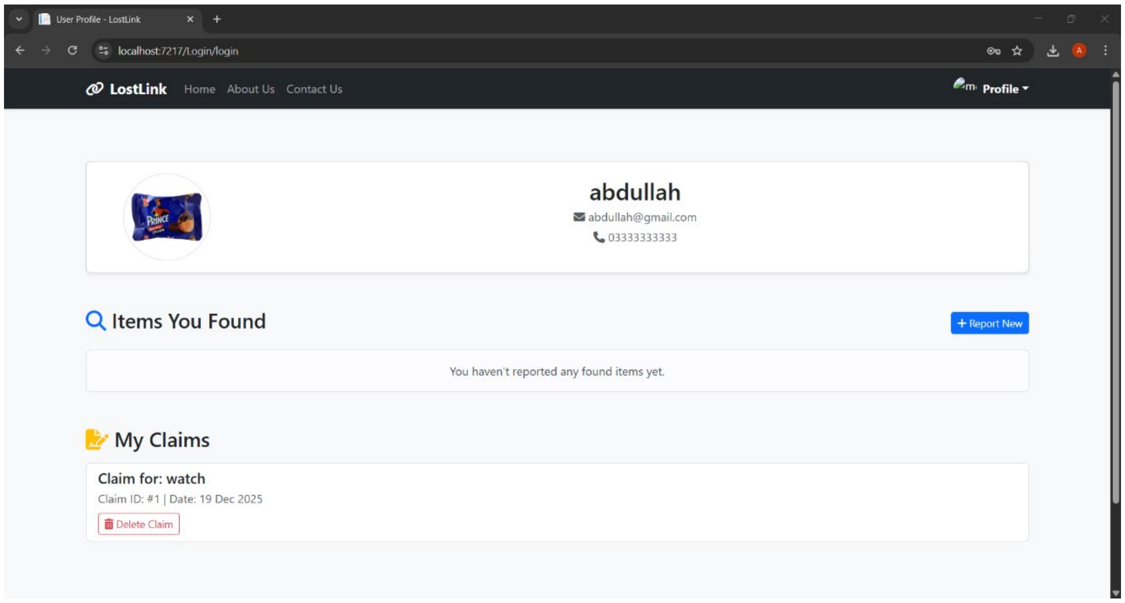
black wallet

Found at: Library

watch

Found at: anaj cafe

5.3. User Claims



User Profile - LostLink

localhost:7217/Login/login

LostLink Home About Us Contact Us Profile

abdullah

abdullah@gmail.com

03333333333

Items You Found

+ Report New

You haven't reported any found items yet.

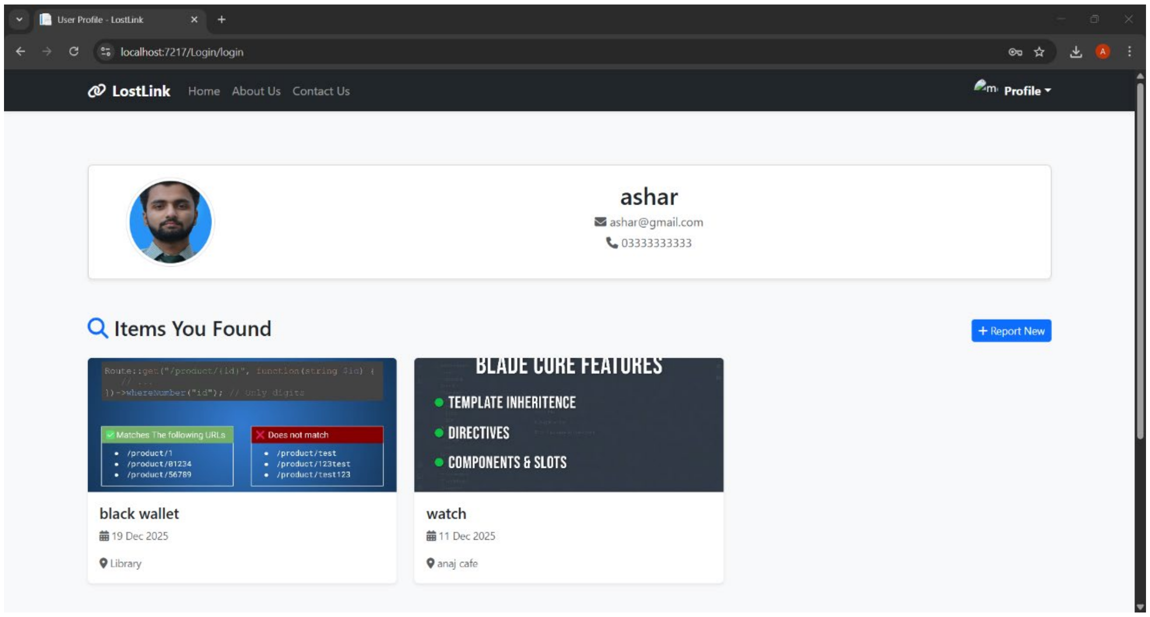
My Claims

Claim for: watch

Claim ID: #1 | Date: 19 Dec 2025

Delete Claim

5.4. User Profile



User Profile - LostLink

localhost:7217/Login/login

LostLink Home About Us Contact Us Profile

ashar

ashar@gmail.com

03333333333

Items You Found

+ Report New

```
Route::get("/product/{id}", function(string $id) {
    // ...
})->whereNumber("id"); // only digits
```

Matches The following URLs
- /product/1
- /product/01234
- /product/56789

Does not match
- /product/test
- /product/123test
- /product/test123

BLADE CORE FEATURES
- TEMPLATE INHERITENCE
- DIRECTIVES
- COMPONENTS & SLOTS

black wallet

19 Dec 2025

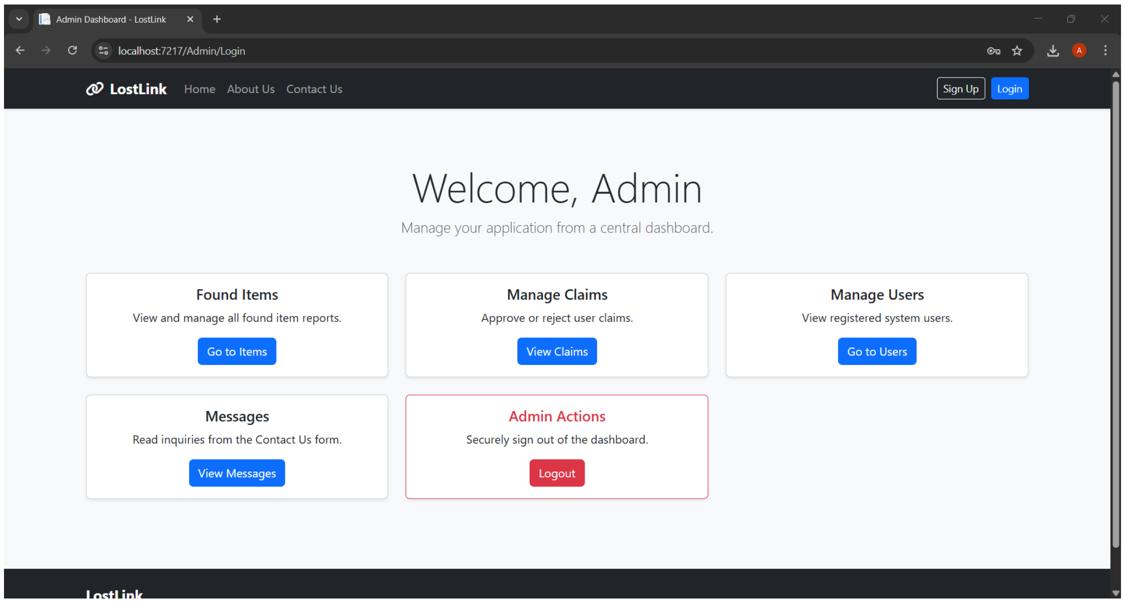
Library

watch

11 Dec 2025

anaj cafe

5.5. Admin



Welcome, Admin

Manage your application from a central dashboard.

Found Items

View and manage all found item reports.

Manage Claims

Approve or reject user claims.

Manage Users

View registered system users.

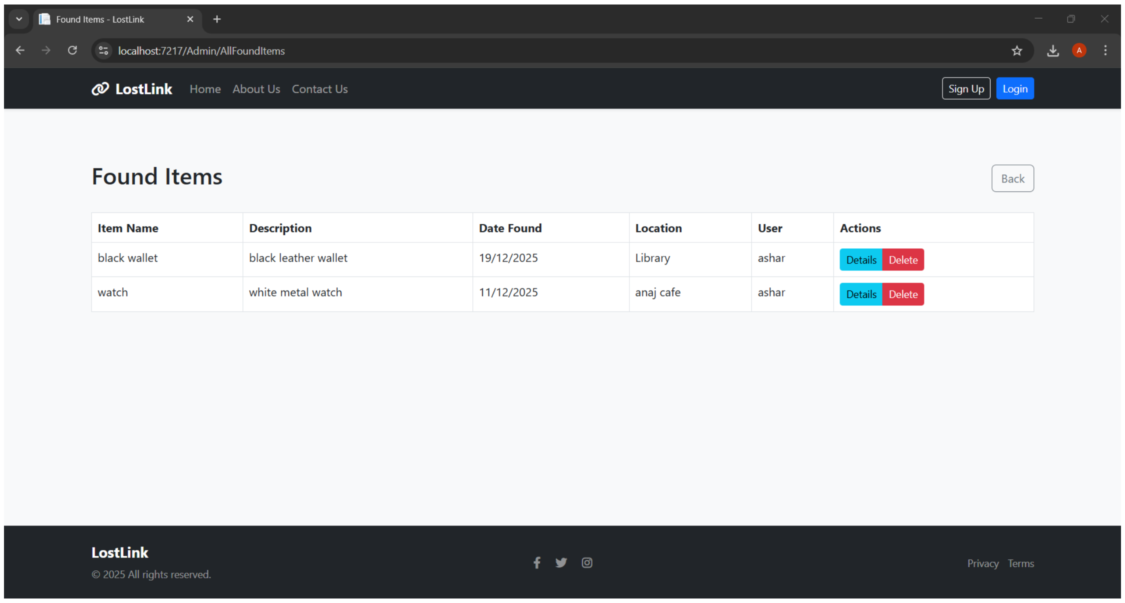
Messages

Read inquiries from the Contact Us form.

Admin Actions

Securely sign out of the dashboard.

5.6. Admin (Managing Found items)



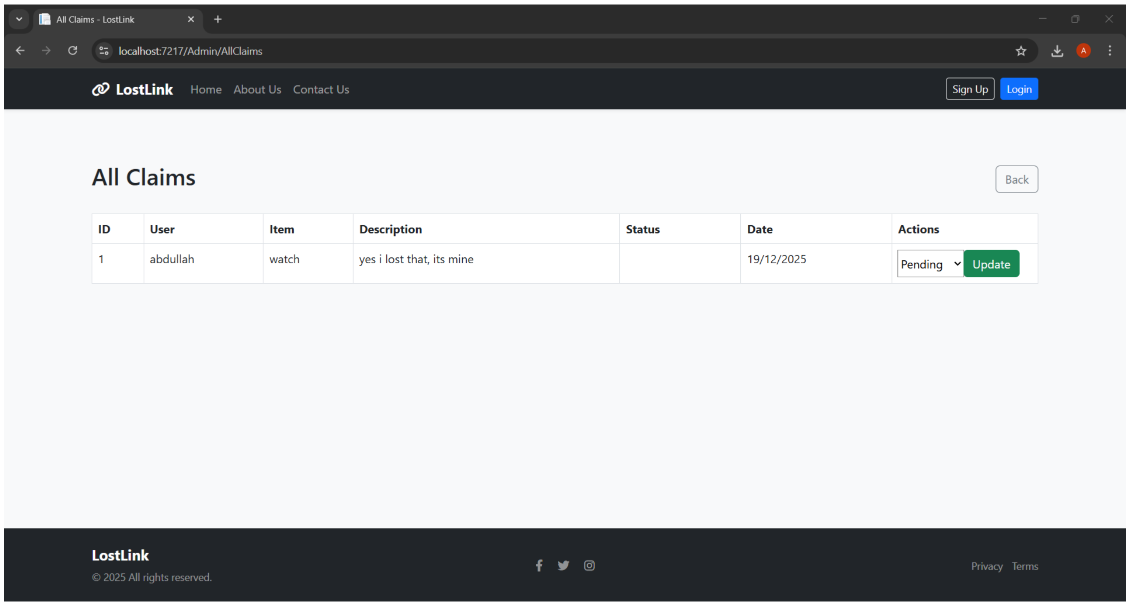
Found Items

Item Name	Description	Date Found	Location	User	Actions
black wallet	black leather wallet	19/12/2025	Library	ashar	Details Delete
watch	white metal watch	11/12/2025	anaj cafe	ashar	Details Delete

LostLink

© 2025 All rights reserved.

5.7. Admin (Managing Claims)

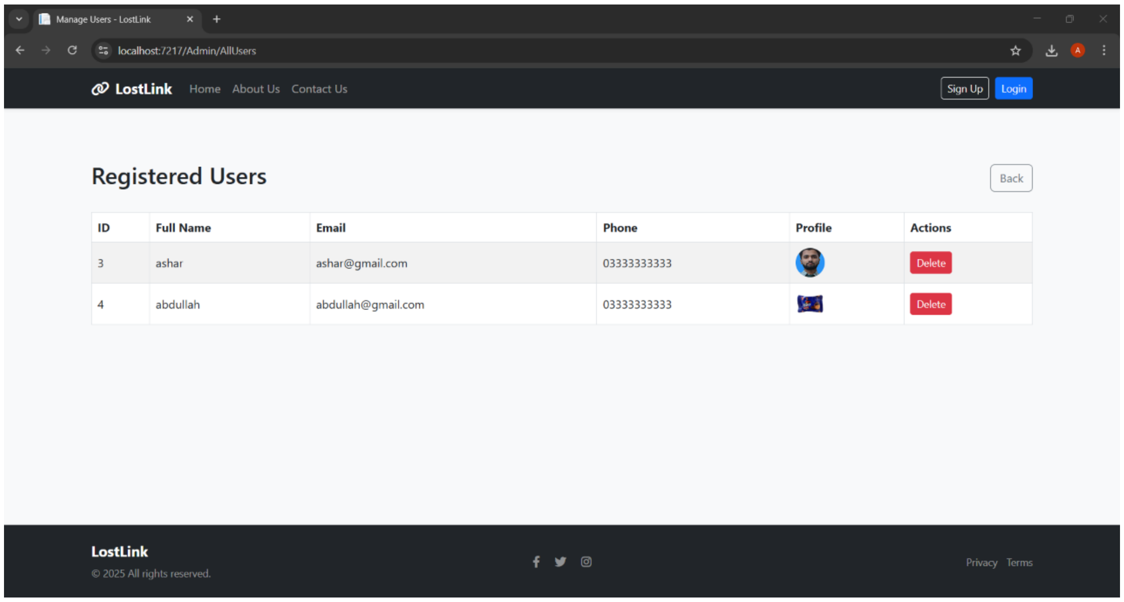


All Claims

ID	User	Item	Description	Status	Date	Actions
1	abdullah	watch	yes i lost that, its mine		19/12/2025	Pending Update

LostLink
© 2025 All rights reserved.
Privacy Terms

5.8. Admin (Managing Users)

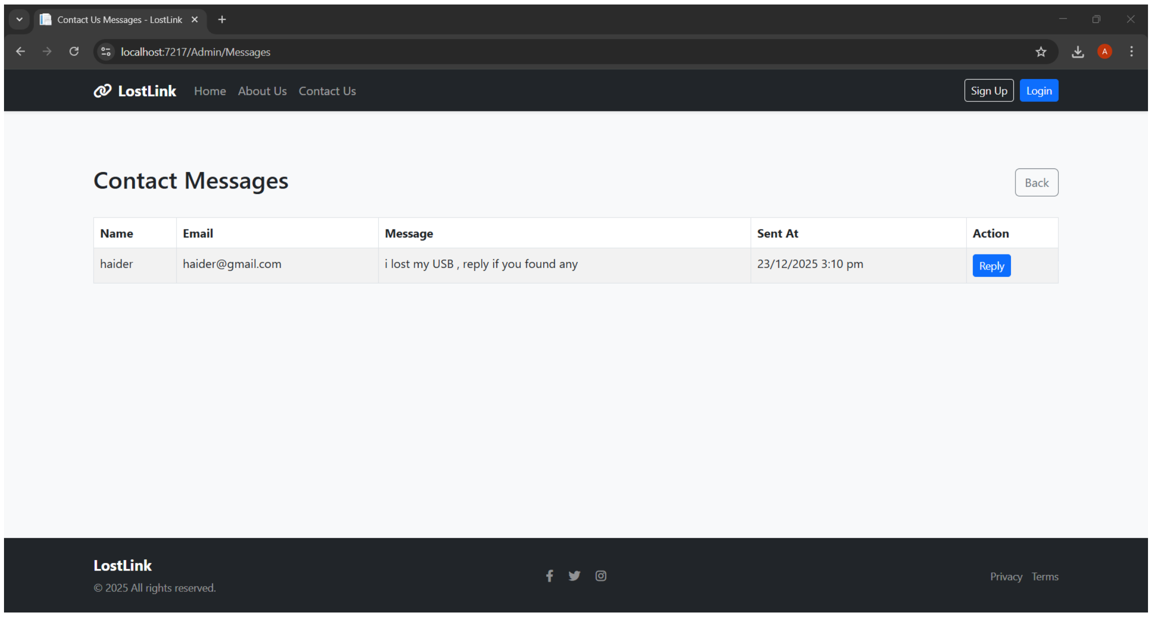


Registered Users

ID	Full Name	Email	Phone	Profile	Actions
3	ashar	ashar@gmail.com	03333333333		Delete
4	abdullah	abdullah@gmail.com	03333333333		Delete

LostLink
© 2025 All rights reserved.
Privacy Terms

5.9. Admin (Contact messages)


Contact Us Messages - LostLink
localhost:7217/Admin/Messages
LostLink
Home
About Us
Contact Us
Sign Up
Login
Back
Reply
LostLink
© 2025 All rights reserved.
Privacy
Terms

Contact Messages

Name	Email	Message	Sent At	Action
haider	haider@gmail.com	i lost my USB , reply if you found any	23/12/2025 3:10 pm	Reply

6. Security Features

- **Authorization:** [Authorize] attributes protect sensitive routes (Profile, Reporting).
- **CSRF Protection:** @Html.AntiForgeryToken() is implemented on all forms.
- **Data Validation:** Strict server-side validation on models prevents invalid data injection.
- **Session Security:** Memory-based session tracking ensures that sessions are invalidated correctly upon server reset.

7. Conclusion

Project demonstrates a practical application of advanced C# and ASP.NET Core concepts. By integrating complex relationships (Users-Items-Claims), secure authentication, and a clear administrative workflow, it provides a scalable solution for campus inventory management.