University Lost & Found System

A comprehensive web-based Lost and Found Management System for universities built with Flask, HTML, CSS, and SQLite.

Features

Admin Portal

- **User Management**: Add/remove student and staff accounts
- **Analytics Dashboard**: View statistics (total lost items, claimed items, pending verifications, monthly summaries)
- **Post Items**: Create posts for lost/found items on behalf of users
- **View All Reports**: Track all submissions with status monitoring
- **Item Handover**: Match lost and found items and record handovers
- **Database Records**: Maintains complete history even after item resolution

User Portal (Students/Staff)

- **Report Lost Items**: Submit detailed reports of lost items
- **Report Found Items**: Submit details of found items
- **View Own Reports**: Track personal submissions
- **Browse All Posts**: View all active lost and found items
- **Dashboard**: Centralized view of all activities

Username Format

Students

Format: `YEAR + DEPARTMENT + ENROLLMENT_NUMBER`

- Example: `2020ASP001`, `2020ASB025`, `2020ICT150`
- Departments: ASP, ASB, ICT

Staff

Format: `university_email@university.edu`

- Example: `john.doe@university.edu`

Admin

- Username: `admin`
- Default Password: `admin123` (change after first login)

Installation & Setup

Prerequisites

- Python 3.8 or higher
- pip (Python package installer)

Step 1: Install Required Packages

```bash
pip install flask werkzeug
```

Step 2: Project Structure

Create the following directory structure:

```
lost_found_system/
│
├── app.py                    # Main Flask application
├── lost_found.db             # SQLite database (auto-created)
│
├── templates/                # HTML templates
│   ├── base.html
│   ├── login.html
│   ├── admin_dashboard.html
│   ├── admin_users.html
│   ├── add_user.html
│   ├── admin_reports.html
│   ├── post_lost_item.html
│   ├── post_found_item.html
│   ├── user_dashboard.html
│   ├── report_lost.html
│   └── report_found.html
│
└── static/
    └── uploads/              # For future file uploads
```

Step 3: Run the Application

1. Navigate to the project directory:

```bash
cd lost_found_system
```

2. Run the Flask application:

```bash
python app.py
```

3. Open your web browser and go to:

```
http://127.0.0.1:5000
```

First-Time Setup

1. **Login as Admin**
   - Username: `admin`
   - Password: `admin123`
2. **Add Users**
   - Navigate to "Manage Users"
   - Click "Add New User"
   - Fill in the details with proper username format
   - Provide a temporary password
3. **Users Can Login**
   - Students/Staff use their assigned username and password
   - They can immediately start reporting lost/found items

Usage Guide

For Admin

1. **Adding Users**
   - Go to "Manage Users" → "Add New User"

- Enter username following the format rules
- Provide full name, user type, and temporary password
- User will receive credentials to login

2. **Posting Lost Items**
   - Dashboard → "Post Lost Item"
   - Fill in item details
   - Submit to create a public post
3. **Posting Found Items**
   - Dashboard → "Post Found Item"
   - Enter found item details
   - Submit to notify users
4. **Matching & Handover**
   - Go to "View Reports"
   - Click "Match & Handover" on any pending item
   - Enter the matching item ID
   - Confirm to complete handover
   - Both posts are marked as resolved but kept in database
5. **View Analytics**
   - Dashboard shows:
     - Total lost items (pending)
     - Total found items (pending)
     - Total claimed items
     - Pending verifications
     - Monthly summary

For Students/Staff

1. **Report Lost Item**
   - Click "Report Lost" in navigation
   - Fill in detailed information

- Submit report
- View status in dashboard

2. **Report Found Item**
   - Click "Report Found" in navigation
   - Describe the item you found
   - Submit report
   - Admin will match it with lost items
3. **Track Your Reports**
   - Dashboard shows all your submissions
   - Status updates (Pending, Claimed, Resolved)
4. **Browse All Items**
   - View all active lost items
   - View all active found items
   - Find your lost items or claim found ones

Database Schema

users

- id, username, password (hashed), user_type, full_name, email, contact, created_at

lost_items

- id, item_name, category, description, location_lost, date_lost, reporter_id, status, created_at

found_items

- id, item_name, category, description, location_found, date_found, finder_id, status, created_at

claims

- id, lost_item_id, found_item_id, claimed_by, verification_status, claimed_at, resolved_at

Security Features

- Password hashing using Werkzeug security
- Session-based authentication
- Role-based access control (Admin vs User)

- Login required for all views
- Admin-only restricted routes

Item Categories

- Electronics
- Documents
- Accessories
- Books
- Clothing
- Keys
- Others

Status Types

- **Pending**: Item not yet matched/claimed
- **Claimed**: Item matched and verified
- **Resolved**: Item handed over to owner

Customization

Change Secret Key

In `app.py`, modify:

```python
app.secret_key = 'your-secure-secret-key-here'
```

Add More Categories

Edit the category options in form templates

Modify Username Pattern

Update validation logic in the add_user route

Troubleshooting

Database Issues

If you encounter database errors:

```bash
# Delete the database file
rm lost_found.db

# Restart the application (it will recreate the database)
python app.py
```

Port Already in Use

Change the port in `app.py`:

```python
app.run(debug=True, port=5001)
```

Template Not Found

Ensure all HTML files are in the `templates/` folder

Future Enhancements

- Email notifications
- SMS alerts
- Image upload for items
- Advanced search and filtering
- Real-time chat between finder and loser
- Mobile app integration
- QR code generation for items
- Multi-language support

Support

For issues or questions:

1. Check the troubleshooting section
2. Review the Flask documentation: https://flask.palletsprojects.com/
3. Verify your Python and package versions

License

This project is created for educational purposes.

Created for University Lost & Found Management *Developed with Flask, SQLite, HTML, and CSS*