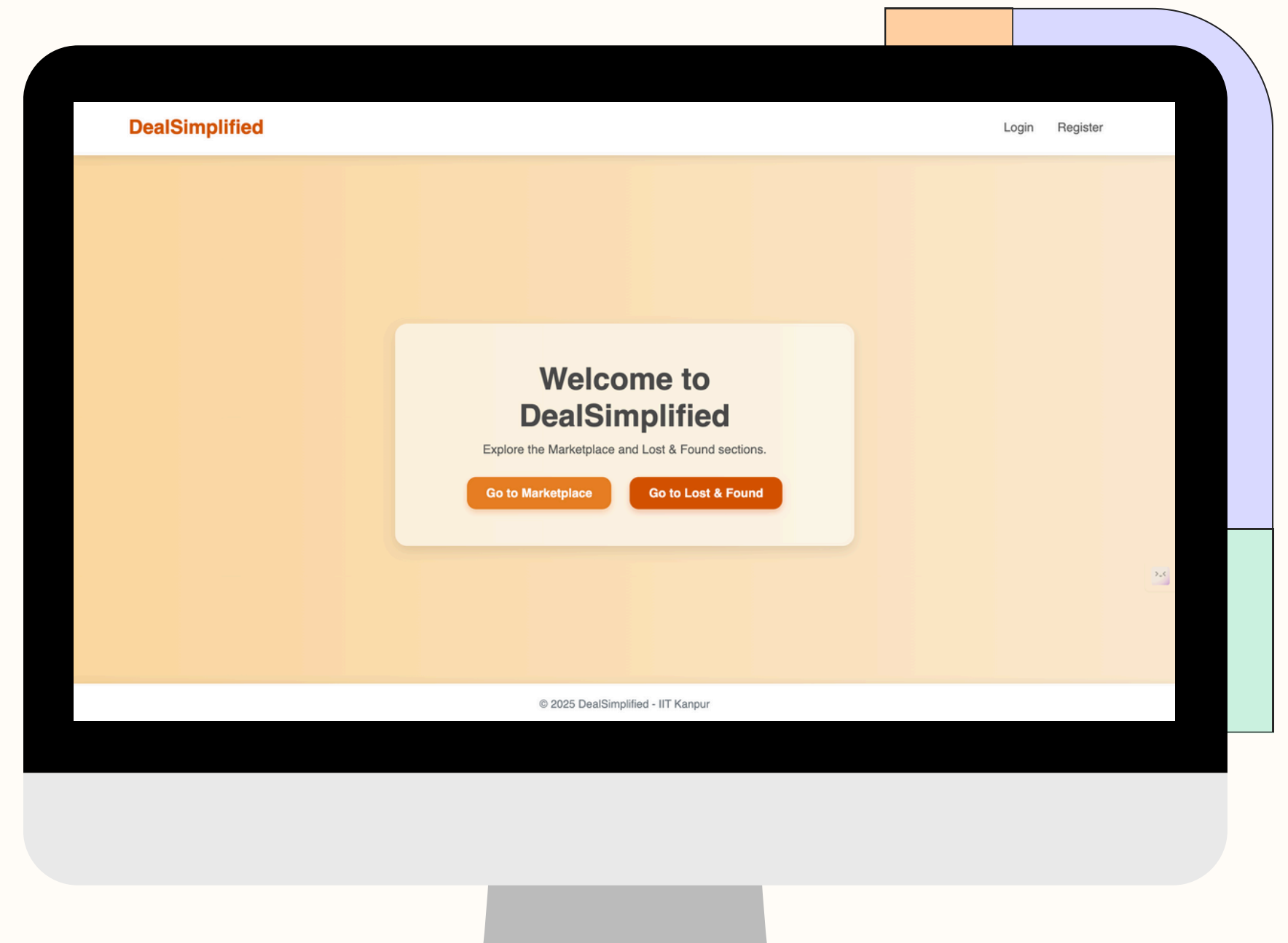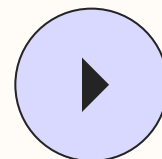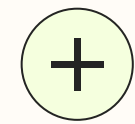Kasukabe Defence Group

DealSimplified

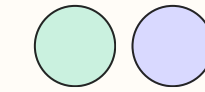An innovative platform designed to address two key aspects of campus life at IIT Kanpur: managing lost and found items and facilitating an in-campus marketplace for buying and selling pre-owned goods.

Github Link

DealSimplified

Login   Register

Welcome to DealSimplified

Explore the Marketplace and Lost & Found sections.

Go to Marketplace   Go to Lost & Found

© 2025 DealSimplified - IIT Kanpur

Need For Software

🔍 Fragmented Recovery Channels

Lost and found processes are often manual, uncoordinated, and dependent on word-of-mouth or physical notice boards. There is a lack of a centralized system to report and recover lost items effectively.

🧾 Informal Marketplace Issues

College and community marketplaces often operate on informal WhatsApp groups or unmoderated platforms, resulting in miscommunication, scams, or missed opportunities due to clutter and lack of categorization.

🧠 Cognitive Overload & Delays

Users often forget what they've lost, where they posted about it, or whom they contacted. A structured platform that tracks lost items, found reports, and product listings with user verification simplifies the process significantly.

🔒 Security & Trust Deficits

Lack of authentication in peer-to-peer exchanges leads to untrustworthy transactions. A platform with verified users, optional ratings, and built-in chat boosts confidence in engagement.

📲 Convenience & Digital Accessibility

Students and residents increasingly prefer seamless, mobile-friendly, and self-service tools. DealSimplified provides a one-stop solution that meets these expectations with ease.

Lost N Found & Marketplace

Potential Users

🎒 Students & Campus Communities

Ideal for college campuses where students frequently lose everyday items like ID cards, chargers, or books, and regularly buy/sell essentials during semester transitions. This platform fixes these issues by structured listings and easy search.

🧍 General Public

Any community, housing society, or institution can use DealSimplified for managing their lost & found or internal marketplace efficiently.

🏫 Institutional Admins / Campus Authorities

Can use the platform to monitor the lost & found activity, maintain order, and resolve ownership disputes. Marketplace moderation and item verification can also be integrated.

🛍️ Buyers & Sellers (Marketplace Users)

Anyone looking to buy or sell products like bicycles, electronics, books, etc., within a trusted local community can benefit from a clean and secure platform.

👨‍💻 Developers & Tech Clubs

As an extensible, well-documented platform, DealSimplified can be adopted or customized by tech clubs or student groups for deployment across campuses or institutions.

Lost N Found & Marketplace
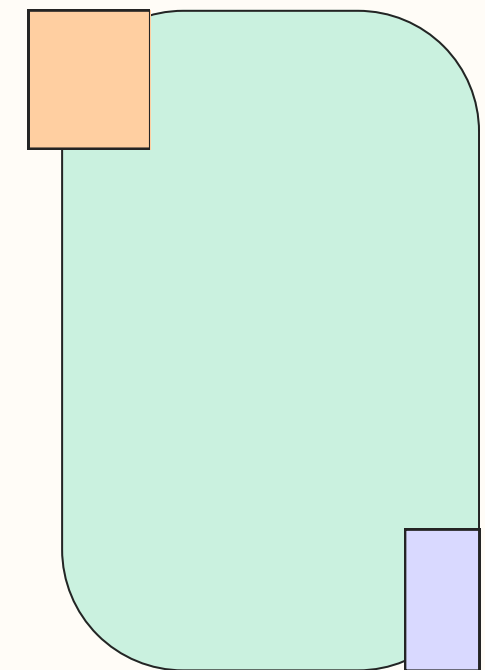
Major Requirements

Functional Requirements

Item Upload: Users can upload details of lost or found items with images, descriptions, category tags, location, and contact info.

Search Functionality: Users can search lost or found items using filters such as category, location, date, and keywords.

Claim Process: Users can initiate a claim for a found item by submitting proof or contacting the uploader.

Notification System: Notify users via email/notifications if an item matching their posted lost/found item is listed.

Chat/Contact Option: Secure contact or in-platform chat for users to communicate about items.

Product Listings: Users can upload products for sale with title, images, specs, price, and description.

Product Categories: Listings are organized into categories (Mobiles, Laptops, Accessories, etc.).

Major Requirements

Functional Requirements

Wishlist/Save Items: Users can save items to view or purchase later.

User Registration & Login: Users can register and log in via email/password.

Smart Match for Lost & Found: Items suggestions based on descriptions, location, and category similarity.

Report Lost Item: Users can post a Lost Item with description, image, and location.

Report Found Item: Users can post a found Item with description, image, and location.

Cart section: Users can temporarily store marketplace items they intend to purchase before proceeding to checkout. Items in the cart remain until removed or purchased.

Non-Functional Requirements

All users must create an account with valid credentials to use the platform's features like posting or claiming items.

The platform ensures that sensitive data such as personal info, chat history, and listing details are protected using secure encryption and user-specific access.

Only authenticated users can access detailed item history, chat threads, or make claims on LostNFound listings.

The system is designed to be highly scalable, allowing for seamless handling of increased user activity and item posts during peak times like college fests or campus drives.

The site is optimized to provide fast loading times (under 3 seconds) even with high traffic and image-heavy listings.

Key Design and Implementation Decisions

Design

A major focus during design was establishing a software framework tailored to user requirements.

It was essential to adopt an object-relational model.

Detailed analysis of class diagrams, use cases, and sequence diagrams guided the decision-making process.

Frontend and backend components were conceptualized separately and later merged.

A top-down architecture was followed, beginning with the user interface and ending with the database layer.

Separate models were created to represent different users, entities, and classes.

The system was built to be adaptable and scalable, combining HTML, CSS, and Python technologies.

Django simplified the integration of object-relational mapping (ORM).

Key Design and Implementation Decisions

Implementation

Project Unit: This refers to the part of the application that users interact with directly. In our case, it's a web-based interface built using HTML, CSS, and JavaScript, which operates within a browser.

Logic Unit: This layer contains the backend logic written in Python. It processes user inputs and coordinates the application's responses, acting as the core driver behind the functionality presented in the project unit.

Framework: Django was selected for its strong security mechanisms, rapid development tools, and built-in administrative interface. Its extensive and active community also offers robust support and abundant resources.

Data Unit: This layer handles data storage using PostgreSQL, chosen for its open-source nature, powerful features, ease of integration, and platform independence. The application was deployed using PythonAnywhere.

Demonstration of the Software

Integration of UPI in the for direct payments -
To streamline transactions on the in-campus marketplace, we plan to integrate UPI (Unified Payments Interface) into DealSimplified. This will allow users to make and receive payments directly through the platform without the need for external apps.

Mobile app deployment -
Since students mostly access platforms via mobile phones, we are working towards launching a fully functional mobile application for both Android and iOS. The app will offer push notifications for updates (e.g., if a lost item is reported or a listed product gets a response), easy image uploading for listings, and smooth integration with mobile payment systems

Admin Dashboard and Moderation Tools -
We plan to build an internal admin dashboard to monitor content quality, handle reported items/posts, and manage user verification. This will help prevent misuse, spam, and fake listings. Admins will be able to view flagged content, suspend accounts, and get analytics on platform usage.

QR Code-Based Item Tagging for Lost and Found -
Provide QR codes that users can stick on their valuables. If someone finds an item, scanning the QR code will show contact info (if shared) or let them submit a "found" report easily

Effective communication is vital in such a large group to ensure everyone is on the same page regarding tasks, deadlines, and project goals. Learning to collaborate efficiently with a large team, dividing tasks, and coordinating efforts to achieve a common objective.

Enhancing technical skills by learning from teammates, sharing knowledge, and solving complex problems together. Developing problem-solving abilities through encountering and resolving various technical challenges and issues that arise during the project.

Learning to manage time effectively to meet deadlines and deliverables, especially when working in a team with multiple dependencies. Embracing feedback from teammates and stakeholders to iterate and improve the web app continuously.

Recognizing the significance of documenting processes and decisions made throughout the project for future reference and maintenance. It supports the long term success of software development projects.

Thank You