



LOST AND FOUND PORTAL

DESIGN DOCUMENT

GAURAV MITTAL

RAHUL MITTAL

KAUSHAL YAGNIK

DEEPAK CHAWLA

DESIGN DOCUMENT

PROBLEM DESCRIPTION

- ✚ It is not difficult to observe that the number of items being lost by people in our campus is increasing day by day. Also, in case a person finds something lying around anywhere in the campus, he/she finds it tough to reach the person who might have lost that item. As a result, the item remains lost forever.
- ✚ Also, most of us who have either lost or found something have no other alternative but to send mail to everyone's account regarding the lost entity. Here also, the person face the dilemma of whom to send the mail, whether to btechxx@iitrpr.ac.in which limits him/her to just the B.Tech students or to the Broadcast which takes a long time to reach the recipients.
- ✚ Now, considering the fact that every day, at least one person loses his/her belonging in the campus, we might be having over a hundred mails in our accounts which are of not much use and also, fill up our valuable space.

OBJECTIVE

Looking at the problem at hand, our objective was to develop some platform which can bring these notifications about lost and found items into one place making it convenient for the concerned person to find his/her lost item or reach the person who has lost the item found and at the same time, deal with the big issue of spamming in our campus.

SOLUTION IMPLEMENTED

- ✚ The solution that we have devised is both elegant as well as user-friendly.
- ✚ We created an Intranet based Lost And Found portal to cater the needs of those who either lose something or find something in the campus.
- ✚ This portal will provide the facility of submitting the details regarding a lost or found item by the user in least effort possible, which can be viewed by other users who can then send message to the user concerned, asking about that item.

DESIGN ISSUES

- ✚ We wanted the portal to be as user-friendly as possible, that is, a user can post details regarding an item or even view details about other items in not more than 3 clicks.
- ✚ Security was a huge concern in our portal. Since we are targeting the reduction of spamming through the LNF Portal, we do not want intruders to gain access of features which they are not entitled to in unauthorized way.

- ✚ Since we aimed to install this service for exclusively IIT Ropar, we wanted to design such that only people from IIT Ropar can avail it. For that some kind of verification system needed to be set up in place.
- ✚ The portal being an Intranet service should rely on the internet as less as possible.
- ✚ The portal should provide an efficient search mechanism allowing the user to pin point the item he/she wants in no time.

DESIGN OPTIONS CONSIDERED

FRAMEWORK TO USE

We wanted our design to be such that it can be easily maintained, is reliable and very user friendly. We had the following choice before us which we did not choose:

1. **Using php for server side and a content management system like Drupal:** Drupal did not provide us the amount of abstraction we were looking for like database independent models and view abstraction. Also, we found PHP more like C.
2. **Using JSP to create servlets:** Since we wanted to accelerate our work and save us from compiling the code every time we made any modification, we chose not to use Java for implementation.

Above all, we wanted to use something different from what we have implemented before and also, do it as quickly as possible, in terms of understanding, readability and writability.

What we chose

Python based Django: Django provided us with great flexibility in programming along with excellent documentation and security features which compelled us to use this framework.

DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

Why Django

There were several advantages that compelled us to use this framework:

- 1) The biggest advantage is that it is based on Python which allowed us to do concise and crisp coding. The entire code that powers our portal is spread across just 500 lines which could have been more than 1000 lines in java and over 2000 lines in C/C++.
- 2) Abstraction of database: With Django, we just needed to create our data models in the form of Python classes providing all the attributes and behavior to the individual entities. After this, Django on its own created the database schema in any of the popular RDBMS. We can even view the corresponding SQL code through Django. This allows us to abstract out data models making them independent of a particular database.

- 3) Flexibility of views: Django provided us the feature of views to handle data flow between the server and client effectively.
- 4) Django implements Model-view-controller Pattern allowing us to create reusable modules/apps.

Model-view-controller Pattern

Django enables us to implement MVC pattern in the portal making it both efficient to use and easy to maintain. MVC pattern divides a software application into three connected parts, separating the internal representations of information from the way that information is presented to or accepted from the user.

Why MVC?

MVC pattern helped us to organize our code well leading to a better design and increased maintainability. Also, it allowed us to divide our work efficiently during the development phase.

DESIGN DECISIONS

- 1) To make the portal exclusively for IIT Ropar, we verified a person's email against the list of valid emails before allowing him/her to create an account and use the facility.
- 2) In the registration form, we did not ask for a password. Instead, we send a mail to the email used for registration containing the password, in order to prevent someone from using another person's email to access the portal.
- 3) In order to save the user from the hassles of logging into the portal every time to use it, we created cookies that will remember whether the user is still logged in and will get automatically deleted when the user logs out.
- 4) We created our own registration system instead of using gmail login API, in order to avoid dependence on the internet.
- 5) When a person logs into the portal, he/she should find all the options on a single page which is a very user friendly design.
- 6) When a user posts a found item, he may wish to remain anonymous, in case he/she does not want to be bothered by unnecessary users.
- 7) A user is able to edit his/her details which include name, address and password.
- 8) User can also edit the description of his/her items.
- 9) Cookies are used to authorize the access whenever some GET/POST request is sent to the server.
- 10) For security purposes, any submitted form is validated both on the client side and the server side.

DESIGN DETAILS

DEVELOPMENT ENVIRONMENT

- ❖ The project is entirely developed in Ubuntu 13.10(Raring Ringtail) and made use of free tools and editor
- ❖ Python was installed in the system, particularly Python 2.7.5
- ❖ Django module was installed using pip:
 - **pip install Django==1.6.2**
 - (We used ver. 1.6.2 as it is the latest stable version available.)
- ❖ Default SQLite database is used which is included in Python.
- ❖ Installed the following to successfully run the application:
 - PIL** : **\$ pip install PIL**
 - Whoosh** : **\$ pip install Whoosh**
 - Django-haystack**: **\$ pip install django-haystack**

STARTING THE PROJECT

- 1) After opening the terminal, current working directory was changed to the desired directory.
- 2) The following command was run from the terminal:

django-admin startproject LostNFound_Project

This created the following directory structure inside the current working directory.

```
LostNFound_Project /
  manage.py
  LostNFound_Project /
    __init__.py
    settings.py
    urls.py
    wsgi.py
```

manage.py: All types of Django management commands can be executed using manage.py such as running the server, creating apps, middleware classes, verifying sql models, creating indexes, etc.

settings.py: This file specifies various settings related to the project such as directory path, list of associated apps, date and time formats, various macros, etc.

urls.py: This file is really important as it specifies list of url patterns associated with the project. Each url pattern is a matching between a regular expression and all the further associated urls to be matched with it.

wsgi.py: This file specifies the settings for the wsgi_mod. We did not need to work with this file.

STARTING THE APP

- 1) The directory is changed to LostNFound_Project.
- 2) The following command is run to create a Django app named LostNFound:

python manage.py startapp LostNFound

This will create LostNFound directory with following layout:

```
LostNFound/  
  __init__.py  
  admin.py  
  models.py  
  tests.py  
  views.py
```

STARTING THE SERVER

The development server can be started by running the following command from the LostNFound_Project directory:

Python manage.py runserver

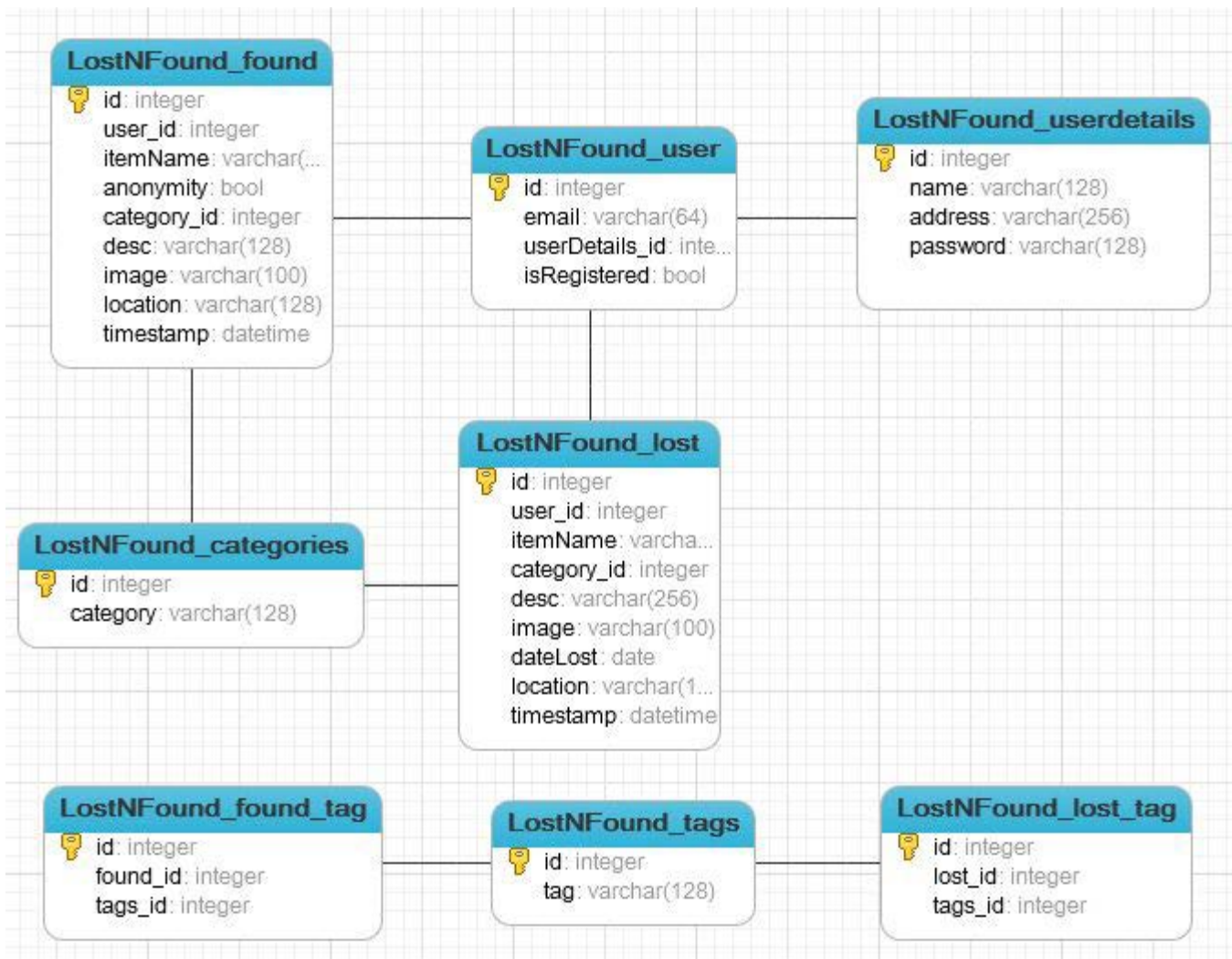
The following output will be displayed on successful starting of the server.

```
E:\4th Semester\CS Project\final\LostNFound_Project>python manage.py runserver  
Validating models...
```

```
0 errors found  
April 30, 2014 - 18:47:42  
Django version 1.6.2, using settings 'LostNFound_Project.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

CREATING THE DATABASE MODEL

- ❖ In Django application, the database model can be created in **models.py**. LostNFound app directory contains the file **models.py** where we specified all the relationship sets and entity sets of the database model as python based classes with required attributes and behavior(methods). The following E-R model summarizes the various features of the database:



- ❖ Run the **syncdb** command to create the model tables in the database:
Python manage.py syncdb
- ❖ Using this code in `models.py`, Django will now create the database schema and a Python database-access API for accessing the various models specified in `models.py`.
- ❖ The SQL code for the created models can be viewed using the following command.
python manage.py sql LostNFound

INCLUDING APP IN THE PROJECT

Django apps are “pluggable”: You can use an app in multiple projects, and you can distribute apps, because they don’t have to be tied to a given Django installation. **LostNFound_Project/settings.py** file is edited by including LostNFound app inside the list of **INSTALLED_APPS**.

```

INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'LostNFound',
)
  
```

)

USING THE ADMIN

- ❖ The admin for the project can be accessed by the url: localhost:8000/admin.
- ❖ Various models can be edited in the admin page. But before this, the model names need to be specified in the **LostNFound_Project/LostNFound/admin.py** as given below:

```
from django.contrib import admin
from LostNFound.models import Tags, Categories, UserDetails, User, Lost, Found

admin.site.register(Tags)
admin.site.register(Categories)
admin.site.register(UserDetails)
admin.site.register(User)
admin.site.register(Lost)
admin.site.register(Found)
```

WRITING DJANGO VIEWS

- ❖ **Views** are the most important part of a Django application. Each View is associated with one or more template web pages and helps in serving a specific function. Basically, they are Django's way of returning an HttpResponse to the user by rendering some HTML webpage after processing some kind of GET/POST request.
- ❖ **Views** in Django for the app LostNFound are specified in LostNFound_Project/LostNFound/ views.py.
- ❖ Each view deals with some part of the functionality of the project such as registering the user, displaying the list of items and processing search query during browsing.
- ❖ The entire working part of the application for the server side is specified in **views.py**.

SPECIFYING THE TEMPLATE WEBPAGES

- ❖ In Django application, the client side template web pages are specified in the **template directory** present in the app directory.
- ❖ All the HTML based templates are specified in LostNFound/templates/LostNFound.
- ❖ The template contain HTML5 code combined with python and javascript code. Python code dynamically renders content into the pages.
- ❖ Javascript helps in client side validation for various form present in the website.
- ❖ Related CSS files are also specified along with the templates.

SETTING UP THE URLS

- ❖ All the urls associated with the LostNFound app are specified in the file **LostNFound/urls.py**.
- ❖ The urls are specified in the form of a list of url patterns. When specifying the url in the web browser, if the specified url matches any of the set url patterns, appropriate template is loaded and view is called. Otherwise, an error is thrown.
- ❖ Each url pattern contains a regular expression to be matched, the name of that particular pattern and a mapping to the associated view to be called in case that url is invoked.

DEFINING FORMS

- ❖ Django provides a very useful functionality of creating form objects inside an app. These form objects are essentially Python based classes with associated attributes and behavior. These forms are highly versatile and have the following features:
 - Validating the POST data.
 - Creating a form widget for the HTML page.
 - Generating errors when the form filled in invalid.
 - Preventing attacks related to SQL and script injection.
- ❖ The forms related to the LostNFound Portal are specified inside **LostNFound/forms.py** file as Python classes.
- ❖ To use these forms, in the views.py, whenever some POST data is received by some view object, that data can be passed to an instance of a form object created for validating that received data. Before any further processing, the validity of the form data is checked and only after successful validation, the data is sent for further processing.

CREATING THE SEARCH MECHANISM

- ❖ For the LostNFound application, to enable the feature of browsing items based on some query based on text input, category and relevance, and also, to match the most relevant found items to a particular lost item and vice versa, a **robust and efficient search mechanism** was required to be put in place.
- ❖ For this, Python based APIs **Whoosh** and **Django-Haystack** are used. Django-Haystack wraps around Whoosh and provides the functionality of search inside the database by creating indexes over it using specific models mentioned in the file **LostNFound/searchindexes.py**.
- ❖ These apps are included in the list of INSTALLEDAPPS inside LostNFound_Project/settings.py.
- ❖ The indexes for the LostNFound database are specified in the folder **LostNFound/templates/search/indexes/LostNFound/**.
- ❖ To use this API, proper functions are called from a view in views.py over the query received from the user.
- ❖ The indexes need to be periodically updated to keep the search fresh.

COOKIES

- ❖ To validate a user and to ensure only authorized access is permitted inside the portal, cookies are used.
- ❖ Another advantage of using cookies is to save the user from logging into the portal every time he/she tries to visit it.
- ❖ Cookies are enabled by including appropriate classes in the list of MiddleWare classes inside LostNFound_Project/settings.py.

HANDLING IMAGES

- ❖ Django has a field named ImageField - a wrapper for the Python library - PIL - Python Imaging Library.
- ❖ To upload images we transferred the file data from request.FILES to the necessary ImageField, which gets stored at path defined in settings.MEDIA_ROOT.
- ❖ Also, Django does not overwrite images which have to be handled manually.

BUGS ENCOUNTERED

- django-haystack creates a cache every time we call its function SearchQuerySet().
- During our beta testing phase, we found this feature is causing duplicates to persist in the list which needed to be removed manually.

WORKING OF THE SOLUTION

- As soon as a person arrives at the portal, he/she will be asked either to login using his email and password or to register if not done yet. Registration is possible only through a valid iitrpr gmail account.
- Proper client side and server side form validation will be experienced by the user at every step to ensure only authorized access is permitted.
- After the person has registered successfully, he/she will receive a mail on the email used for registration containing the password for login.
- The person, after procuring the password, is ready to login into the portal.
- When a person logs into the portal, cookies are saved on the client browser which ensure that the person is saved from logging in every time he/she wishes to use the portal.
- After the person has successfully logged into the portal, he will be taken to the home page where he will find various useful links:

I HAVE LOST SOMETHING

The screenshot displays the 'LOST and FOUND portal' interface. At the top, a blue header contains the text 'LOST and FOUND portal' and 'Welcome, Kaushal'. Below the header, a navigation bar includes links for 'Home', 'My Profile', and 'Logout'. The main content area features a form titled 'Enter the details of the Lost Item'. The form fields are as follows:

- Category ***: A dropdown menu with 'Gadget' selected.
- Item Name ***: A text input field containing 'ipod Nano'.
- Place Lost ***: A text input field containing 'Cafeteria'.
- Date lost ***: A date picker showing '2014-04-25'.
- Tags**: A list of tags including 'HDFC', 'phone', 'Earring', and 'sports'. The 'ipod' tag is selected.
- Description**: A text input field containing 'Orange color' and 'Very Slim'.
- Image**: A 'Browse...' button next to the filename 'IMG_1352-500x375.jpg'.

A 'SUBMIT' button is located at the bottom of the form. A note at the bottom of the form states: 'Fields marked with * are mandatory.' The footer of the page reads: 'Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg'.

- When a user clicks on this link, a form will load where the user can specify the details regarding the item he has lost such as item name, category, description, tags (important points about the item), image and date lost. Some of them are required denoted by an asterisk (*).
- After successful submission, the user will be redirected to the page of the recently created lost item.



Item Name : Ipod nano
 Lost by : Kaushal Yagnik
 Category : Gadget
 Place lost : Cafeteria
 Date lost : 04/25/2014
 Tags :
 Description : Orange color Very Slim

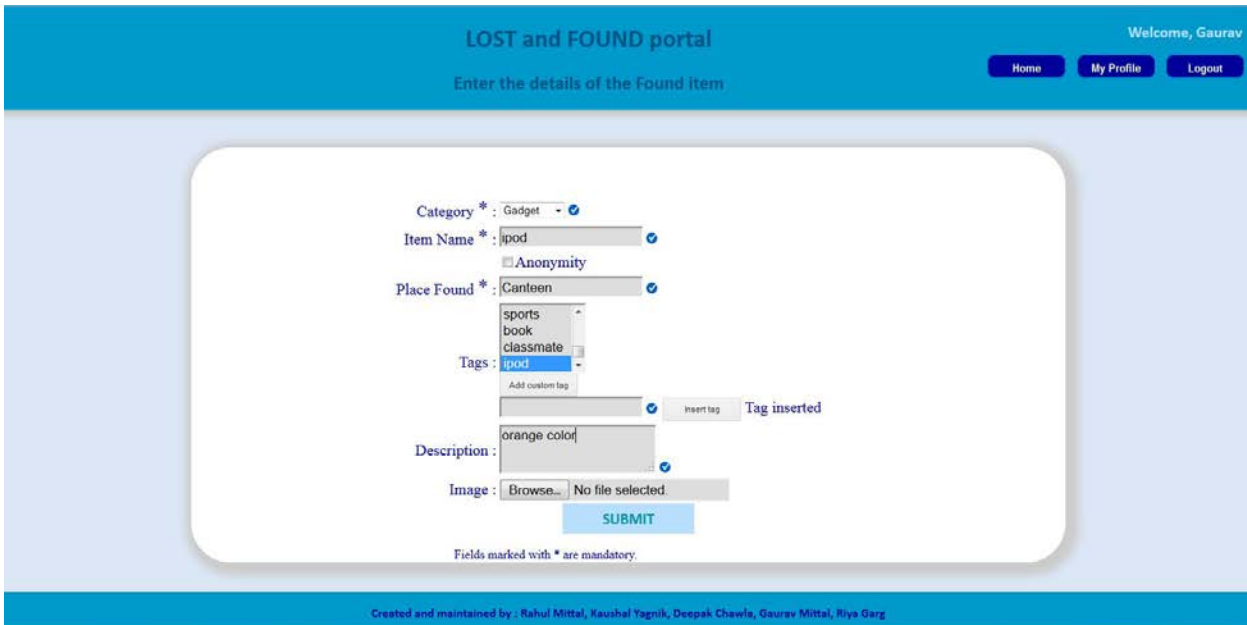
[Edit item detail](#) [Delete item](#)

Items Relevant to this item

Item Name	Category	Description
ipod	Gadget	

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

I HAVE FOUND SOMETHING



LOST and FOUND portal
 Welcome, Gaurav
[Home](#) [My Profile](#) [Logout](#)

Enter the details of the Found item

Category * : Gadget
 Item Name * : ipod
☐ Anonymity
 Place Found * : Canteen
 Tags : sports, book, classmate, ipod
 Add custom tag
 Description : orange color
 Image : Browse... No file selected.
 SUBMIT

Fields marked with * are mandatory.

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this link, a form will load where the user can specify the details regarding the item he has found such as item name, category, description, tags (important points about the item), image and date lost. Some of them are required denoted by an asterisk (*). He can also specify to remain anonymous.
- After successful submission, the user will be redirected to the page of the recently created lost item.

- Also, if any lost item existing in the database matches this found item, a mail will be sent to the user corresponding to the best matched item notifying him/her about the newly created found item.

My

LOST

ITEMS

LOST and FOUND portal

Welcome, Kaushal Yagnik

Home My Profile Logout

My Lost items

Image	Item Name	Category	Description	Change..
	Ipod nano	Gadget	Orange color Very Slim	Edit Delete

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this link, he/she will be taken to a page listing the items lost by him/her.
- The details of a particular object can be seen by clicking on its name which will display a page containing the items details and options to edit and delete it.

My FOUND ITEMS

LOST and FOUND portal

Welcome, Gaurav

Home My Profile Logout

My Found items

Image	Item Name	Category	Description	Change..
	Ear ring	Jewellery	Gold valuable	Edit Delete
	Mobile phone	Gadget	Brand new smartphone	Edit Delete
	Copy	Textbook	white copy.	Edit Delete
	Ipod	Gadget		Edit Delete

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this link, he/she will be taken to a page listing the items found by him/her.

- The details of a particular object can be seen by clicking on its name which will display a page containing the items details and options to edit and delete it.

BROWSE ALL LOST ITEMS

LOST and FOUND portal

Welcome, Gaurav

[Home](#)
[My Profile](#)
[Logout](#)

Browse items

Search Query:

Category: All

Relevance: Relevance

Filter

Item Name	Category	Description
Id card	ID Card	Has my photo on it !
Samsung mobile phone	Gadget	A 757 wave model
Watch	Gadget	
Book	Textbook	
Ipod nano	Gadget	Orange color Very Slim
Basketball	Sports	lost it somewhere in the court

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this link, he/she will be taken to a search page where the user can browse through the list of all the lost items present in the database.
- The user can filter the items according to category, pass strings to search for items in the database and also, order the items according to relevance, name or date lost.
- The user can view the details of a particular object by clicking on its name.

BROWSE ALL FOUND ITEMS

LOST and FOUND portal

Welcome, Gaurav

[Home](#)
[My Profile](#)
[Logout](#)

Browse items

Search Query:

Category: All

Relevance: Relevance

Filter

Item Name	Category	Description
Laptop charger	Gadget	65W charger found in MP Hall
Credit card	Money	Credit Card Found. Please verify to claim
Ear ring	Jewelery	Gold valuable
Mobile phone	Gadget	Brand new smartphone
Copy	Textbook	white copy.
Ipod	Gadget	
Del toro	Textbook	AC no 22551

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this link, he/she will be taken to a search page where the user can browse through the list of all the found items present in the database.
- The user can filter the items according to category, pass strings to search for items in the database and also, order the items according to relevance, name or date lost.
- The user can view the details of a particular object by clicking on its name.

ITEM DESCRIPTION PAGE



Item Name : Ipod nano
 Lost by : Kaushal Yagnik
 Category : Gadget
 Place lost : Cafeteria
 Date lost : 04/25/2014
 Tags :
 Description : Orange color Very Slim

[Edit item detail](#) [Delete item](#)

Items Relevant to this item

Item Name	Category	Description
ipod	Gadget	

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on the name of an item anywhere in the portal, he will be shown the description page for that item.
- This page displays all sorts of information related to the item such as related user, image, description, tags, anonymity (in case of found item), date lost (in case of lost item) and category.
- Along with these, if the item was posted by the same user, he will get links to either edit the details for that item or even delete that item altogether.
- If the item was posted by some other user, the user visiting the item will get a text box which can be used to send mail to the concerned user.

EDIT PROFILE

LOST and FOUND portal

Welcome, Gaurav

HomeLogout

Edit..

Email ID : gauravmi@iitrpr.ac.in
Name : Gaurav
Address : Room: 210 Jupiter House

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

LOST and FOUND portal

Welcome, Gaurav

HomeLogout

Email ID : gauravmi@iitrpr.ac.in
Name : Gaurav
Address : Room: 210 Jupiter House
Old Password :
New Password :
Confirm Password :
ApplyCancel

All the fields in this form are mandatory

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- This page provides the user the functionality to edit his/her name, address and even password (only after providing his/her old password).

LOGOUT

LOST and FOUND portal

Login to continue

Successfully Logged Out

Email ID :

Password :

LOGIN

Are you a new user? [REGISTER](#) your email here...

Created and maintained by : Rahul Mittal, Kaushal Yagnik, Deepak Chawla, Gaurav Mittal, Riya Garg

- When a user clicks on this button, he will be logged out of the portal and all cookies will be deleted automatically without him/her knowing about it.