VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAUM-590 014

A Mini-Project Report
on

"HOUSE INVENTORY MANAGEMENT SYSTEM"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

TAYSEER MOHAMMED FAROOQ, 1VE20CS159
CHAUHAN YASHWANTSINGH, 1VE20CS031

Under the Guidance of

Ms. Veda N
Assistant Professor
Department of Computer Science and Engineering
Sri Venkateshwara College of Engineering, Bangalore-562 157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SRI VENKATESHWARA COLLEGE OF ENGINEERING
BANGALORE - 562 157

2022-2023

SRI VENKATESHWARA COLLEGE OF ENGINEERING,
Vidyanagar, Bangalore – 562 157
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Mini-Project entitled **“House Inventory Management System”** carried out by **Mr. Tayseer Mohammed Farooq - USN 1VE20CS159** and **Mr. Chauhan Yashwantsingh – USN 1VE20CS031** of V Semester students of Sri Venkateshwara College of Engineering, in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of **Visvesvaraya Technological University, Belgaum** during the academic year 2022-2023. The Mini-Project report has been approved as it satisfies the academic requirements in respect of DBMS Laboratory with Mini-Project (18CSL58) work prescribed for the said Degree.

Signature of the Guide	**Signature of the HOD**
Ms. Veda N	**Dr. S. C. Lingareddy**
Asst. Professor, Dept. of CS&E	HOD, Dept. of CS&E
SVCE, Bangalore	SVCE, Bangalore

Name of the Examiners: **Signature with Date**

1.

2.

ACKNOWLEDGMENT

It gives us immense pleasure to express our sincere gratitude to the management of **Sri Venkateshwara College of Engineering, Bengaluru** for providing the opportunity and the resources to accomplish our project work in their premises.

We would also like to convey our regards to **Dr. Nageshwara Guptha, Principal, SVCE** for providing us with the infrastructure and facilities needed to build our project.

Heartfelt and sincere thanks to **Dr. Sanjeev Lingareddy, HOD, Dept. of CSE**, for his suggestions, constant support and encouragement.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Ms. Veda N, Assistant Professor, Dept. of CSE**, for her invaluable help and guidance.

TAYSEER MOHAMMED FAROOQ	**–**	**1VE20CS159**
CHAUHAN YASHWANTSINGH	**–**	**1VE20CS031**

ABSTRACT

House Inventory Management System is a software which is helpful for anyone who wants to manage his/her items in home. It is time saving and scalable. Our House Inventory Management System deals with various managing activities related to the household items. In the software we can register as a user. User can add, modify, delete, browse and print the items. User can easily search his/her items location.

CONTENTS

CHAPTER No. PAGE No.

1. INTRODUCTION…………………………………………..1

1.1 OBJECTIVES

1.2 LIMITATIONS

2. STUDY OF EXISTING SYSTEM……………………………2 - 3

2.1 A CASE STUDY ON HIMS 2

2.2 PROPOSED SYSTEM 3

3. DATABASE DESIGN…………………………………………4 - 13

3.1 SOFTWARE REQUIREMENT SPECIFICATION 4

3.1.1 COLLECTION OF REQUIREMENTS

3.1.2 SOFTWARE AND HARDWARE REQUIREMENTS

3.2 CONCEPTUAL DESIGN 5 - 6

3.2.1 ER DIAGRAM 5

3.2.2 DATABASE SCHEMA DIAGRAM 6

4. IMPLEMENTATION…………………………………………………7 - 13

4.1 FRONT END 7 - 9

4.2 BACK END 9 - 12

4.3 TRIGGER 13

4.4 STORED PROCEDURE 13

5. USER INTERFACES…………………………………………14 - 17

6. CONCLUSIONS AND FUTURE ENHANCEMENT………18

6.1 CONCLUSIONS

6.2 FUTURE ENHANCEMENT

REFERENCES…………………………………………………...19

LIST OF FIGURES

FIGURE-NO	NAME	PG-NO
3.1	ER DIAGRAM	5
3.2	DATABASE SCHEMA	6

LIST OF TABLES

TABLE-NO	NAME	PG-NO
4.1	INVENTORY_LOCATION	10
4.2	INVENTORY_ITEMS	10

CHAPTER-1

INTRODUCTION

1.1 OBJECTIVES:

- The main objective of the project is to design and develop a user friendly-system
- Easy to use and an efficient computerized system.
- To develop an accurate and flexible system, it will eliminate data redundancy
- To study the functioning of House Inventory Management System.
- To make a software fast in processing, with good user interface.
- To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
- To provide synchronized and centralized farmer and seller database.
- Computerization can be helpful as a means of saving time and money.
- To provide better Graphical User Interface (GUI).
- Less chances of information leakage.
- Provides Security to the data by using login and password method.
- To provide immediate storage and retrieval of data and information.
- Reducing paperwork.

1.2 LIMITATIONS:

- It is very time-consuming process.
- Dynamic storage space is required for the maintainability.
- Less Reliable use of papers for storing valuable data information is not at all reliable.

CHAPTER-2

STUDY OF EXISTING SYSTEM

2.1 CASE STUDY

This House inventory management system is a unique idea which we have implemented to search any kind of items stored in home. Generally, we intend to forget the location of the things which we have kept for long term use. For example Suitcases, Old Blankets, etc. So, this is where our idea of house inventory database comes into picture. Our database does the following things:

A database can be used to store information about each item, including its location, description, and any other relevant information.

A user interface can be developed to allow users to view the current location of each item and update information about the items as needed.

The system can also be integrated with a mobile app, which would allow users to view the location of items on their mobile devices and update information remotely.

The system can be tested by simulating scenarios where the user is searching for a particular item, and check if the system is able to provide accurate location and details of the item in real-time.

Overall, a house inventory system with item location tracking can help users keep track of their possessions and make it easier to find lost or misplaced items.

2.2 PROPOSED SYSTEM

A proposed system for a house inventory management system would involve the following key components:

Data entry: Users would be able to input information about their belongings, including item name, location, date of acquisition, value, and any relevant notes or photos.

Database: All of the information entered would be stored in a centralized database, allowing users to easily access and update their inventory.

Search and filter: Users would be able to search and filter their inventory by various criteria, such as location, value, or date of acquisition.

Access Control: Users would be able to share their inventory with their family members, insurance company, or other trusted parties with access control.

Mobile Compatibility: The system would be available on mobile devices, so users can access their inventory from anywhere.

Cloud-based: The system would be cloud-based, so users can access their inventory from any device with an internet connection.

CHAPTER-3

DATABASE DESIGN

3.1 SOFTWARE REQUIREMENT SPECIFICATION

3.1.1 SOFTWARE REQUIREMENTS:

Frontend - HTML, CSS, Bootstrap

Backend - SQLite3, Python, DB Browser

Framework - Django

- Operating System: Mac OS, Windows 11
- Google Chrome
- Workspace editor: Visual Studio Code

3.1.2 HARDWARE REQUIREMENTS:

- Computer with a 1.1 GHz or faster processor
- Minimum 2GB of RAM or more
- 2.5 GB of available hard-disk space
- 5400 RPM hard drive
- 1366 × 768 or higher-resolution display

3.2 CONCEPTUAL DESIGN:

3.2.1 E-R DIAGRAM:

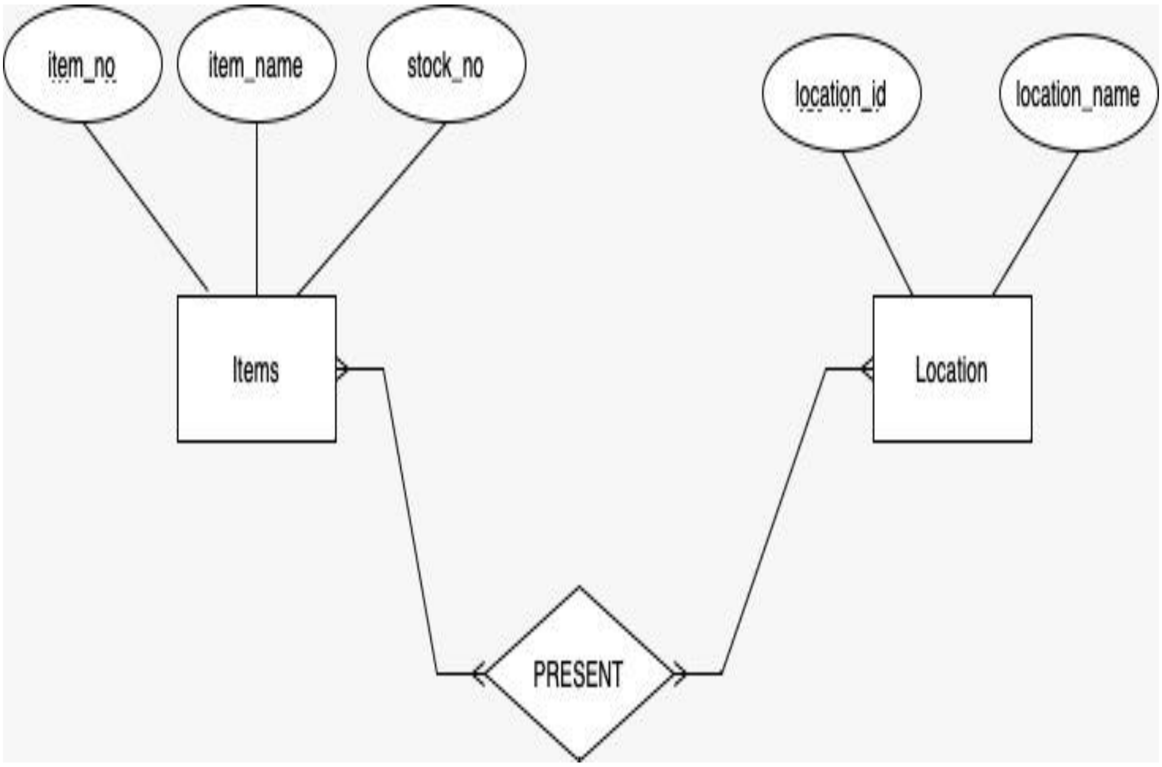


Figure – 3.1 ER DIAGRAM

3.2.2 DATABASE SCHEMA DIAGRAM:

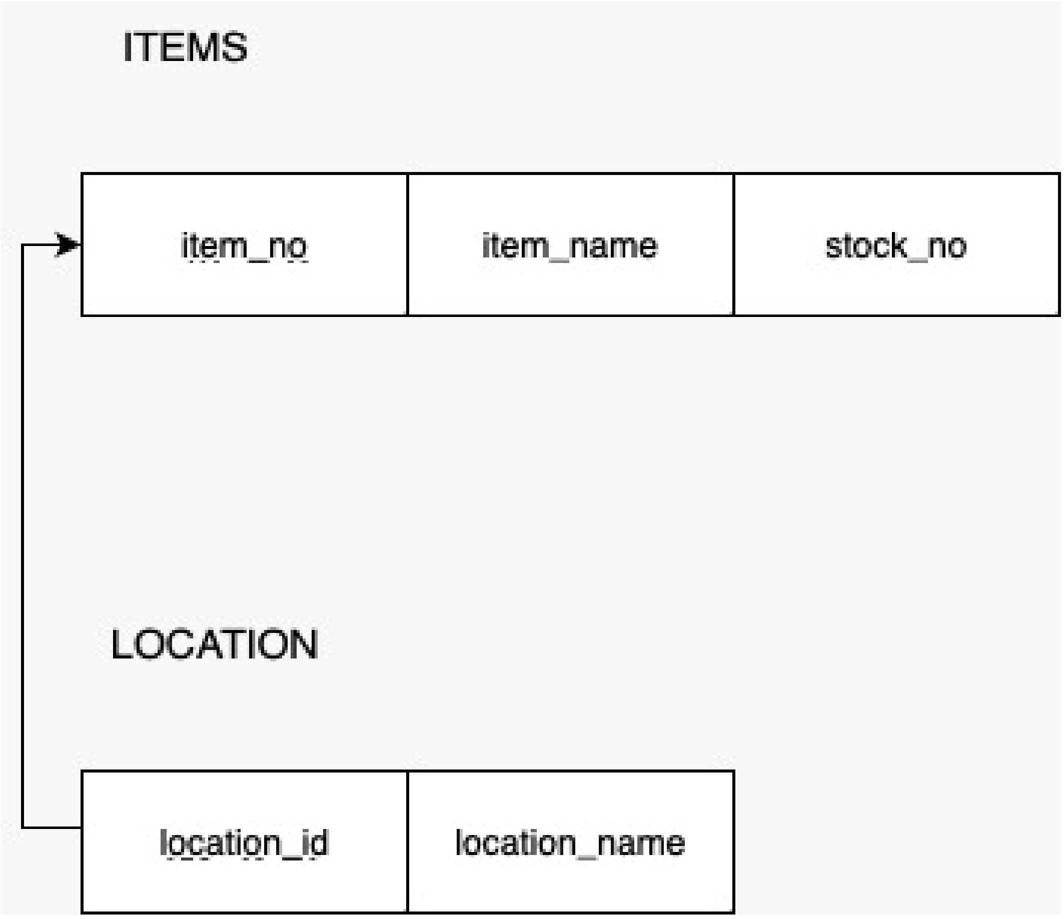


Figure – 3.2 DATABASE SCHEMA

CHAPTER – 4

IMPLEMENTAION

The implementation of this house inventory management of the language starts with creating a virtual environment which is used to Execute the program only in the preferred directory. Next, we have to install Django in our virtual network and configure all the models and templates using HTML & CSS respectively.

4.1 FRONT END

Django HTML:

```
{% extends 'inventory/base.html' %}
{% block content %}
  <h1> Welcome to Home Inventory</h1>
  <table class="table">
    <thead>
      <tr>
        <th>Item name</th>
        <th>Stock number</th>
        <th>Location</th>
      </tr>
    </thead>
    <tbody>
      {% for i in items  %}
      <tr>
        <td>{{ i.item_name}} </td>
        <td>{{ i.stock_no}}</td>
```

```
            <td>{{ i.location_id}}</td>


            {% endfor %}
          </tr>
        </tbody>
    </table>
{% endblock  %}
```

Bootstrap:

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <title>House Inventory</title>
  </head>
  <body>
    {% block  content %}
```

```
   {% endblock  %}

   <!-- Optional JavaScript -->

   <!-- jQuery first, then Popper.js, then Bootstrap JS -->

   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>

   <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1
" crossorigin="anonymous"></script>

   <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

  </body>

</html>
```

4.2 BACK END (SQLite3)

SQLite3:

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

BACKEND WITH SQLite 3 CODE

```
CREATE TABLE "inventory_location" (

        "id"     integer NOT NULL,
        "name"varchar(25) NOT NULL,

        PRIMARY KEY("id")

);
```

Inventory_location Table:

ID	NAME
1	Class Room
2	Basement Room

Table – 4.1 INVENTORY_LOCATION

```
CREATE TABLE "inventory_items" (

        "id"     integer NOT NULL,

        "item_name"   varchar(255) NOT NULL,

        "stock_no"      integer NOT NULL,

        "location_id"   bigint NOT NULL,

        PRIMARY KEY("id"),

        FOREIGN KEY("location_id") REFERENCES "inventory_location"("id")
DEFERRABLE INITIALLY DEFERRED

);
```

Inventory_items Table:

Item_name	Stock_no	Location_id
Shirt	3	1
Suitcase	2	2

Table – 4.2 INVENTORY_ITEMS

```
CREATE TABLE "django_migrations" (

        "id"     integer NOT NULL,

        "app"   varchar(255) NOT NULL,
        "name"varchar(255) NOT NULL,
        "applied"        datetime NOT NULL,
        PRIMARY KEY("id" AUTOINCREMENT)
);
```

BACKEND WITH PYTHON (models.py)

```
from django.db import models
class Location(models.Model):
    name = models.CharField(max_length=25)
       def _str_(self):
        return self.name
class Items(models.Model):
    item_name = models.CharField(max_length=255)
    stock_no= models.IntegerField()
    location = models.ForeignKey(Location, on_delete=models.CASCADE)
```

BACKEND WITH PYTHON (migrations/0001_initial.py)

```
from django.db import migrations, models
import django.db.models.deletion
class Migration(migrations.Migration):
    initial = True
    dependencies = [
    ]
    operations = [
        migrations.CreateModel(
            name='Location',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=25)),
            ],
        ),
        migrations.CreateModel(
            name='Items',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('item_name', models.CharField(max_length=255)),
                ('stock_no', models.IntegerField(max_length=10)),
                ('location',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='inventory.location')),
            ],
        ),
    ]
```

4.3 TRIGGERS

It is the special kind of stored procedure that automatically executes when an event occurs in the database.

Triggers used:

1: Trigger name: on delete

Table: Items

Time: after

Event: delete

Definition: DELETE FROM items WHERE ITEM_NAME = ‘$item_name’

4.4 STORED PROCEDURE

Routine name: proc

Type: procedure

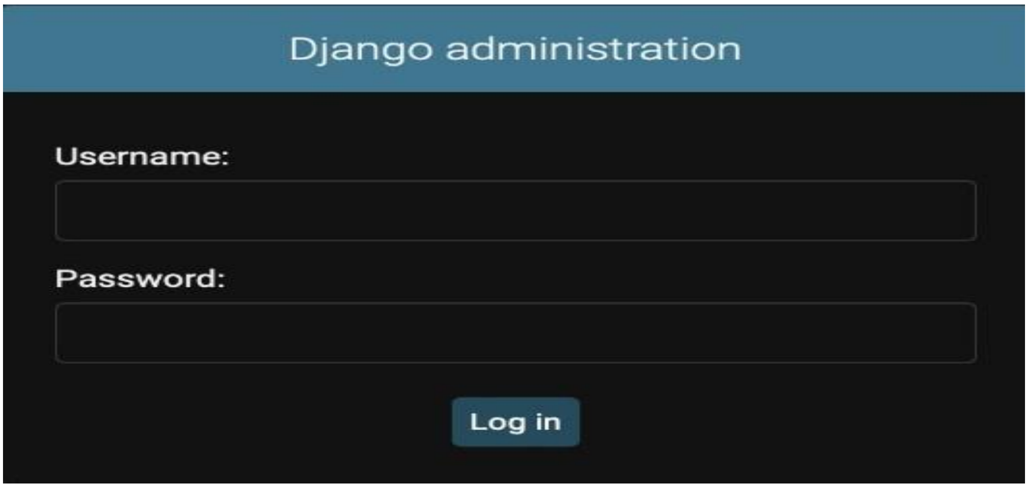
Definition: Select * from inventory;

CHAPTER-5

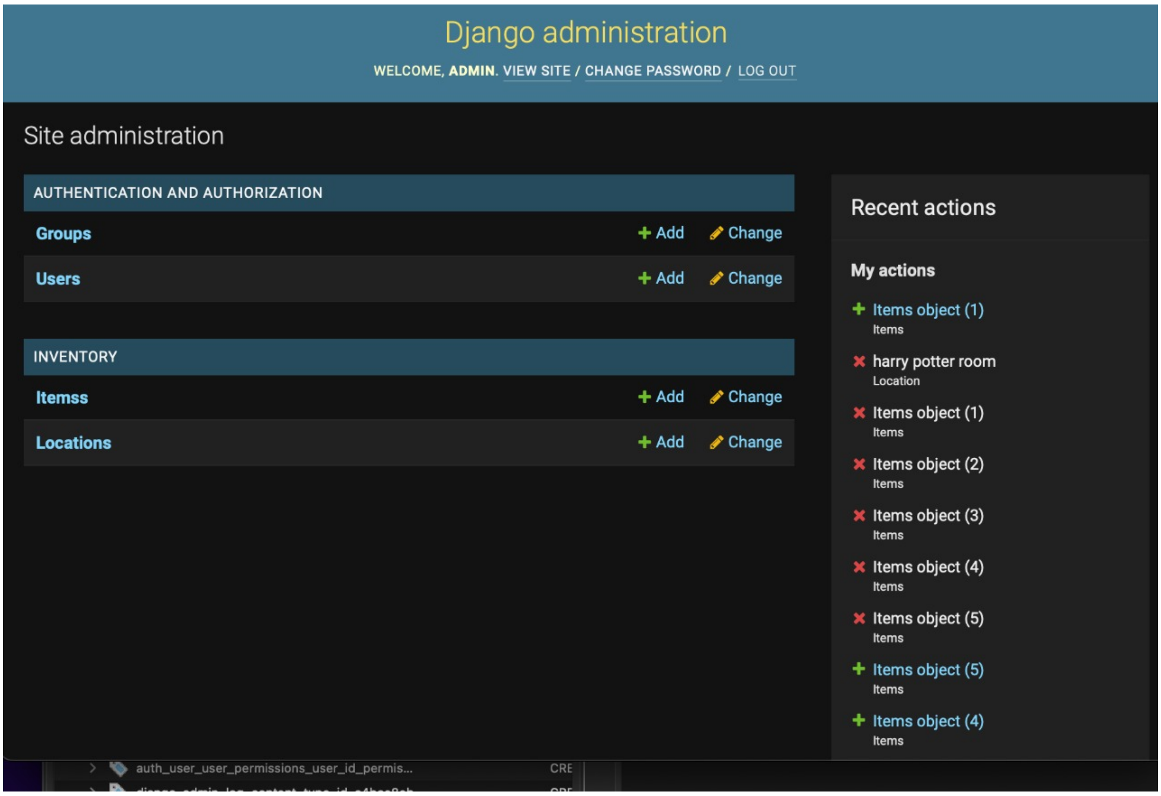
USER INTERFACE

SCREEN SHOTS

LOGIN PAGE:



HOME PAGE:



Django administration

WELCOME, **ADMIN**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	✎ Change
Users	+ Add	✎ Change

INVENTORY		
Itemss	+ Add	✎ Change
Locations	+ Add	✎ Change

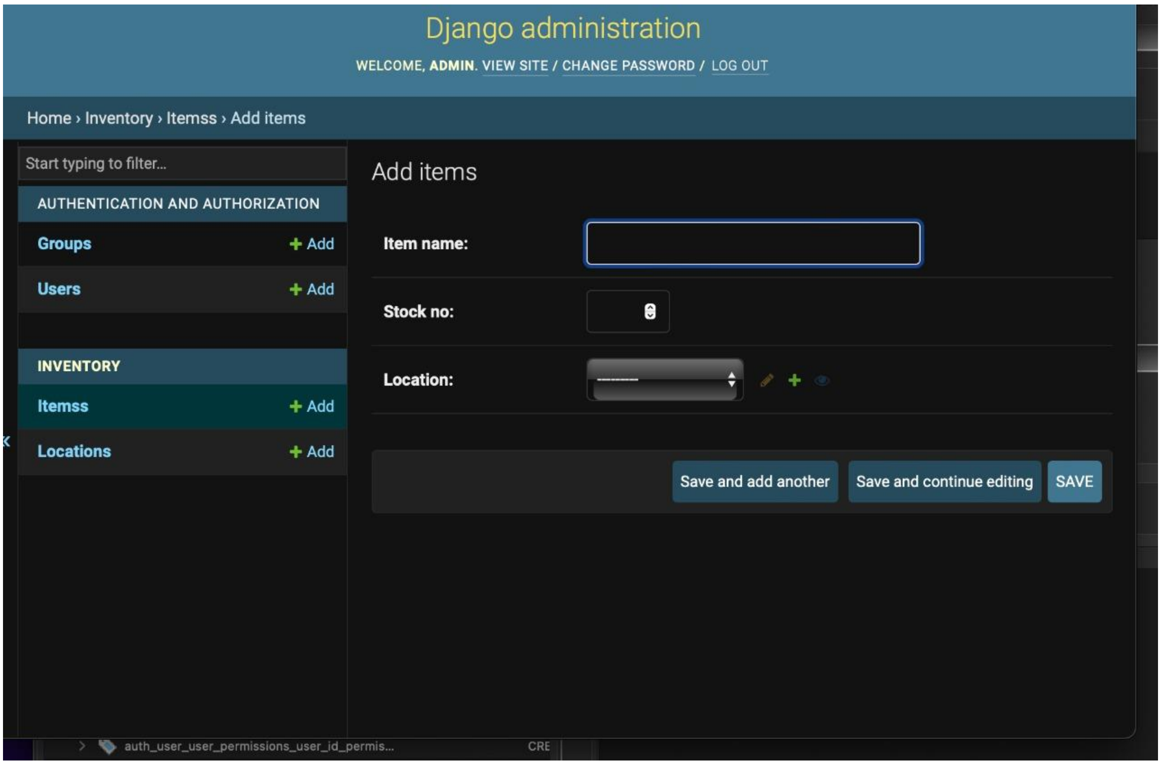
Recent actions

My actions

- + Items object (1)
Items
- ✖ harry potter room
Location
- ✖ Items object (1)
Items
- ✖ Items object (2)
Items
- ✖ Items object (3)
Items
- ✖ Items object (4)
Items
- ✖ Items object (5)
Items
- + Items object (5)
Items
- + Items object (4)
Items

auth_user_user_permissions_user_id_permis... CRE

ADD ITEMS PAGE:

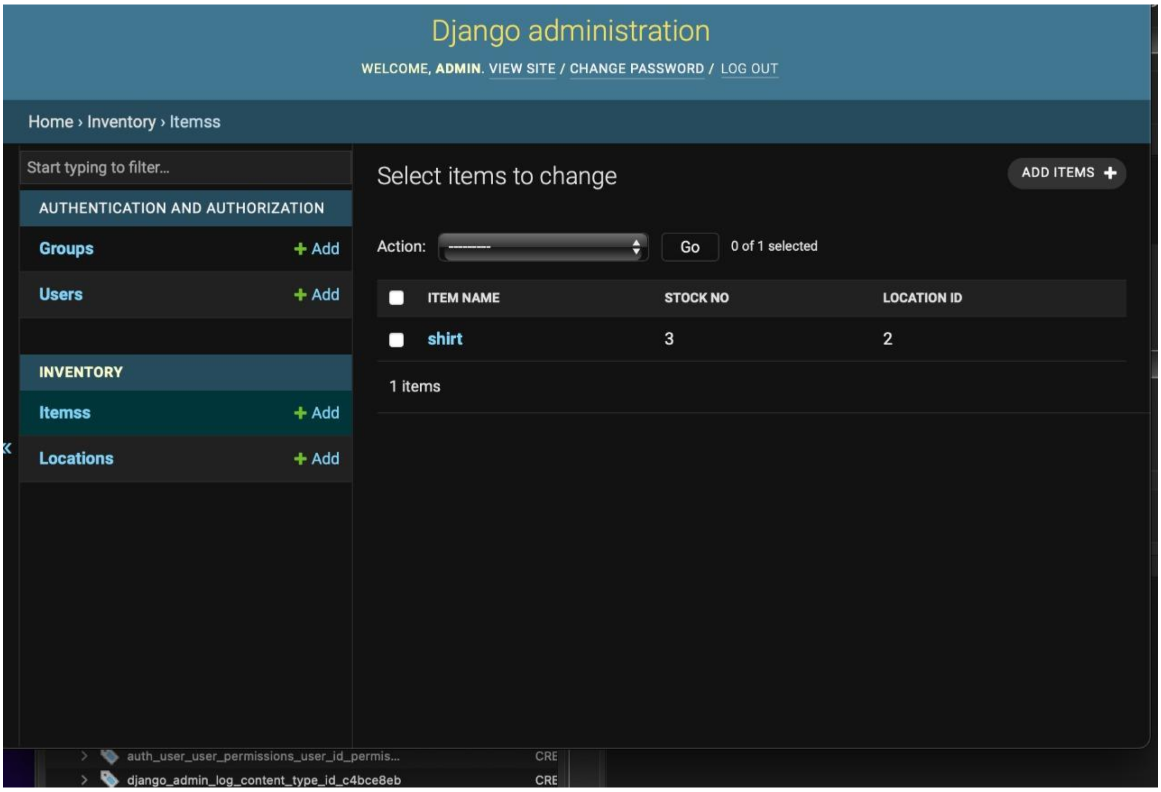

Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Inventory › Itemss › Add items
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
INVENTORY
Itemss + Add
Locations + Add
Save and add another
Save and continue editing
SAVE
auth_user_user_permissions_user_id_permis...
CRE

Add items

Item name:

Stock no:

Location: ---------


Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Inventory › Itemss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
INVENTORY
Itemss + Add
Locations + Add
ADD ITEMS +
Go
auth_user_user_permissions_user_id_permis...
CRE
django_admin_log_content_type_id_c4bce8eb
CRE

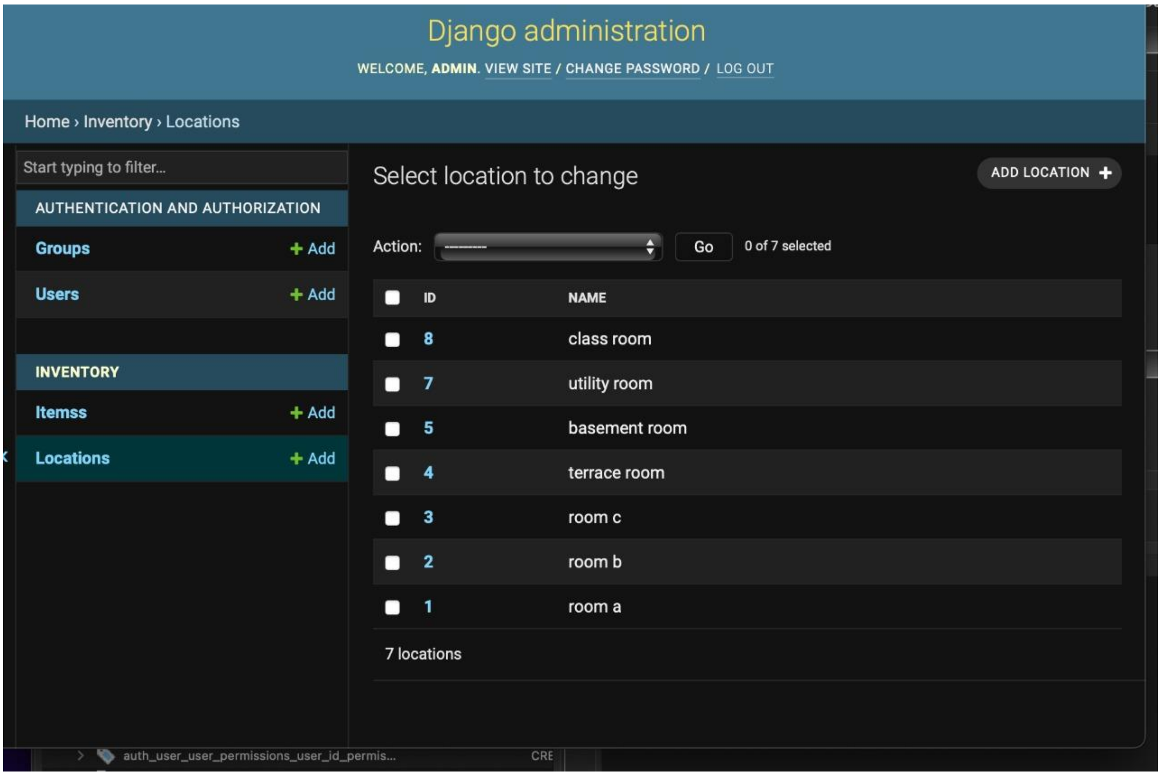
Select items to change

Action: ---------   0 of 1 selected

□	ITEM NAME	STOCK NO	LOCATION ID
□	shirt	3	2

1 items

ADD LOCATION PAGE:

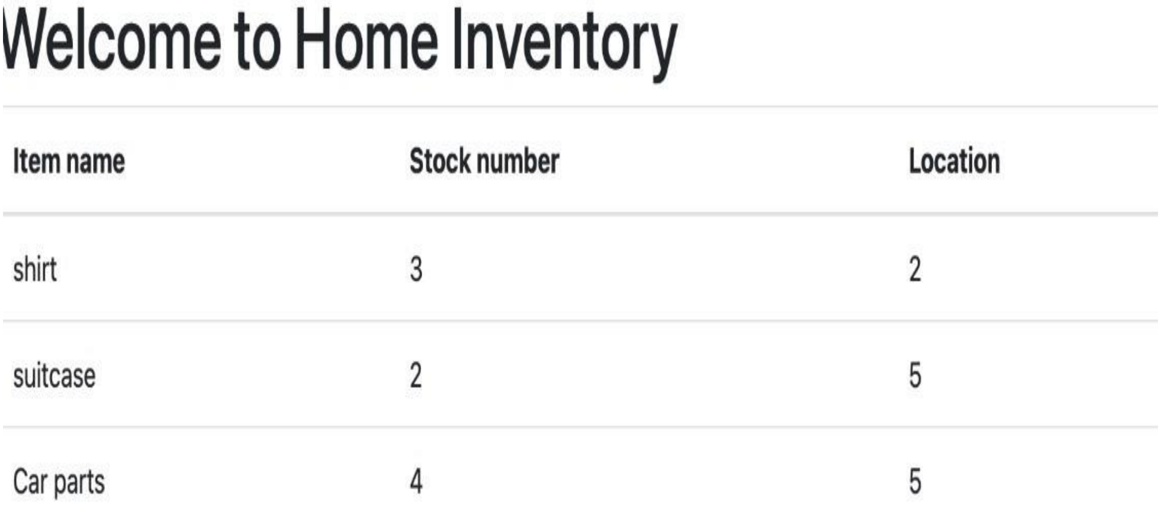

Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Inventory › Locations
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
INVENTORY
Itemss + Add
Locations + Add
ADD LOCATION +
Action: ---------
Go
0 of 7 selected
auth_user_user_permissions_user_id_permis...
CRE

Select location to change

□	ID	NAME
□	8	class room
□	7	utility room
□	5	basement room
□	4	terrace room
□	3	room c
□	2	room b
□	1	room a

7 locations

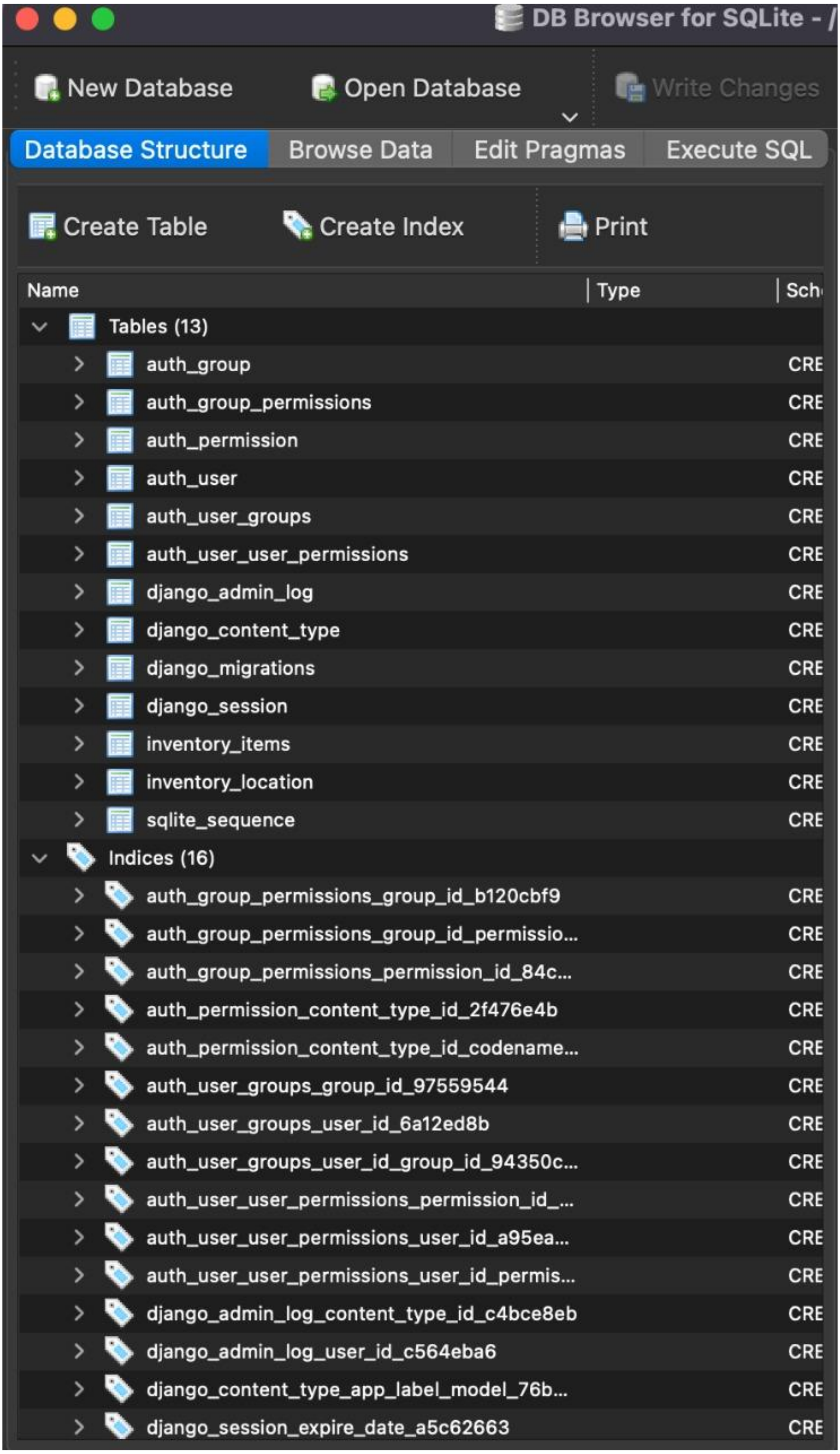
HOME INVENTORY PAGE:



Welcome to Home Inventory

Item name	Stock number	Location
shirt	3	2
suitcase	2	5
Car parts	4	5

DATABASE STRUCTURE:



Name	Type	Sch
Tables (13)		
auth_group		CRE
auth_group_permissions		CRE
auth_permission		CRE
auth_user		CRE
auth_user_groups		CRE
auth_user_user_permissions		CRE
django_admin_log		CRE
django_content_type		CRE
django_migrations		CRE
django_session		CRE
inventory_items		CRE
inventory_location		CRE
sqlite_sequence		CRE
Indices (16)		
auth_group_permissions_group_id_b120cbf9		CRE
auth_group_permissions_group_id_permissio...		CRE
auth_group_permissions_permission_id_84c...		CRE
auth_permission_content_type_id_2f476e4b		CRE
auth_permission_content_type_id_codename...		CRE
auth_user_groups_group_id_97559544		CRE
auth_user_groups_user_id_6a12ed8b		CRE
auth_user_groups_user_id_group_id_94350c...		CRE
auth_user_user_permissions_permission_id_...		CRE
auth_user_user_permissions_user_id_a95ea...		CRE
auth_user_user_permissions_user_id_permis...		CRE
django_admin_log_content_type_id_c4bce8eb		CRE
django_admin_log_user_id_c564eba6		CRE
django_content_type_app_label_model_76b...		CRE
django_session_expire_date_a5c62663		CRE

CHAPTER-6

CONCLUSION AND FUTURE ENCHANCEMENT

6.1 CONCLUSION:

HOUSE INVENTORY MANAGEMENT SYSTEM successfully implemented based on online data filling which helps us in administrating the data user for managing the tasks performed in students. The project successfully used various functionalities of Bootstrap and Python- Django and also create the fully functional database management system for any portals.

Using SQLite 3 as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the SQLite3 database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQLite. We can also use DB browser to easily access our database and customize it

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “House Inventory Management System” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

- The planning that goes into implementing a project.
- The importance of proper planning and an organized methodology.
- The key element of team spirit and co-ordination in a successful project.

6.2 FUTURE ENHANCEMENT

- Enhanced database storage facility
- Enhanced user-friendly GUI
- More Details to the storage area
- online feedbacks forms

REFERENCES

- https://www.youtube.com
- https://www.google.com
- http://www.getbootstrap.com