

LABORATORY INVENTORY MANAGEMENT SYSTEM

at

Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and
Engineering

By

VEERAPUNENI SAI VENKATA SIVA NANDA
Reg.No.37110814



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU
APRIL 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of VEERAPUNENI SAI VENKATA SIVA NANDA (Reg.No.37110814) who carried out the project entitled “LABORATORY INVENTORY MANAGEMENT SYSTEM” under my supervision from October 2020 to March 2021.

Internal Guide
. Mrs.M.D.ANTO PRAVEENA M.C.A.,M.E.,(Ph.D)

Head of the Department
Dr.S.VIGNESHWARI M.E.,Ph.D.,
Dr.L.LAKSHMANAN M.E.,Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I VEERAPUNENI SAI VENKATA SIVA NANDA “LABORATORY INVENTORY MANAGEMENT SYSTEM” done by me under the guidance of Mrs.M.D.ANTO PRAVEENA M.E., (Ph.D). is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to I would like to express my sincere and deep sense of gratitude to my Internal Project Supervisors Dr. Albert Mayan M.E., Ph.D. and Dr. S. Murugan M.E., Ph.D. for their valuable guidance, suggestions and constant paved way for the successful completion of my project work.

I would like to express my sincere and deep sense of gratitude to my Project Guide Mrs.M.D.ANTO PRAVEENA M.E., (Ph.D.), for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

NAME: ARJUN DAVE

**SATHYABAMA**
INSTITUTE OF SCIENCE AND TECHNOLOGY
CREATING THE 21ST CENTURY UNIVERSITY
Established with Deem'ty by MHRD | ISO 9001:2015 | Approved by AICTE

MENU

- Give Licence
- Add Tools
- All Tools
- Calendar
- View Bookings

INCHARGE ID: 123
Aditi

Incharge Dashboard

Welcome Aditi



Atomic Force Microscope (Afm)
Resource Id: 1550278825



Ball Milling
Resource Id: 3180596424
Status: Unavailable



Flash Chromatography
Resource Id: 638863853
Status: Available

IN HOUSE PROJECT

PROJECT SUPERVISORS

Dr. S. Murugan M.E., Ph.D.

Dr. J. Albert Mayan M.E., Ph.D.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	8
	LIST OF FIGURES	9
	LIST OF TABLES	10
	LIST OF ABBREVIATIONS	11
1	INTRODUCTION	12
	1.1 OUTLINE OF THE PROJECT	12
	1.2 TECHNOLOGIES USED	13
2	METHODOLOGY	15
	2.1 LITERATURE REVIEW	15
	2.2 DATA COLLECTION	15
	2.3 BUILD THE DATABASE	16
	2.4 CREATE THE SUITABLE GUI WITH CODING	16
	2.5 SOFTWARE TESTING	16
	2.6 AIM & SCOPE	16
3	MATERIALS, METHODS AND ALGORITHMS USED	18
	3.1 HARDWARE REQUIREMENTS	18
	3.2 SOFTWARE REQUIREMENTS	18
	3.3 SOFTWARE DESCRIPTION	18
	3.4 ARCHITECTURAL DIAGRAM	21
	3.5 PROPOSED SYSTEM	22
	3.6 PROJECT DESCRIPTION	23
	3.7 THE CODE DESIGN	23

4	RESULTS AND DISCUSSION	29
	4.1 RESULTS	29
	4.2 DISCUSSION	29
5	CONCLUSION AND FUTURE WORKS	30
	5.1 CONCLUSION	30
	5.2 FUTURE WORKS	30
	5.3 REFERENCES	31

ABSTRACT

LIMS Application is Laboratory Inventory Management System which automates the process of booking lab equipment for research purpose. This system is built on the Django framework and uses MySQL for the database.

Getting the instrument(s) for our experiment is often a tedious process also at times it leads to delay in the experiments because there is a lot of communication gap between the lender and borrower. So, the Laboratory Inventory Management System helps us to make easy, the process of booking the instrument through a web interface which allows us to see if the required instrument is already booked at the requested time slot, is under maintenance or not functioning.

This makes processes more economical and affordable for research faculties and students in the long run. It also frees the faculties from performing banal tasks day in and day out. Automated processes are not prone to errors and even if an error is identified rectification is easy and can be applied system wide without any delay.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.1	METHODOLOGY	15
3.1	DJANGO MVT ARCHITECTURE	19
3.2	EXAMPLE MODEL CODE	20
3.3	EXAMPLE VIEW FUNCTION CODE	20
3.4	EXAMPLE TEMPLATE CODE	21
3.5	SQL TABLE	21
3.6	ARCHITECTURE DIAGRAM	23

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
3.1	PAGE, URL & FUNCTION TABLE	23

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
LIMS	Laboratory Inventory Management System
MVT	Model View Template
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
GUI	Graphical User Interface

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

- Lab inventory management is an important, yet mundane task in every laboratory, that is often overlooked or neglected. Managing lab inventory effectively can be significant in reducing substantial costs and saving time, enabling you to run your lab efficiently and increase productivity.
- Many labs still choose to run their inventory on paper, or spreadsheets, making it almost impossible to properly track the stock levels or material location. Many times, this results in loss of expensive reagents, antibodies, cell lines etc.
- Using web-based lab inventory management software can help laboratories manage their inventory in a productive and cost-effective manner. The advantages of such a system over using paper are clear – You can gain control and keep track of everything in one place, manage your stock levels and new orders, create stock alerts, easily share inventory locations with all lab members- so everyone is up to date. Lab Inventory System is a full-featured solution for research laboratories, designed to enable organizations to perform effective inventory monitoring, setting and assessing stock levels, up-to-date inventory management, for the instruments of the laboratory.
- The management of laboratory instruments and the associated analysis and reporting are time-consuming manual processes. At times, we are also unable to complete most of our experiments on time due to most of the instruments missing.
- A Lab Inventory Management System tends to have a base set of functionalities that defines it. That functionality can roughly be divided into four laboratory processing phases, with numerous software functions falling under each:
 - i. To have the details of the instruments available
 - ii. the reception and login of users for the instruments,

- iii. the assignment, scheduling, and tracking of the instruments,
- iv. the inspection, approval, and accepting of the instruments for other orders.

It is a more traditional client/server architecture, with some of the system residing on the computer or workstation of the user (the client) and the rest on the server. This makes processes more economical and affordable for research faculties and students in the long run. It also frees the faculties from performing banal tasks day in and day out. Automated processes are not prone to errors and even if an error is identified rectification is easy and can be applied system wide without any delay.

1.2 TECHNOLOGIES USED

This project is built using

- ☐ Django 2.2.5
- ☐ MySQL 0.9.3
- ☐ HTML 5
- ☐ CSS 3
- ☐ JavaScript ES6 and
- ☐ Bootstrap 4

CHAPTER 2

METHODOLOGY

The main methodology is divided into various parts. The parts are literature review, data collection; build the database, creating the suitable Graphical User Interface (GUI) complete with coding and software testing.

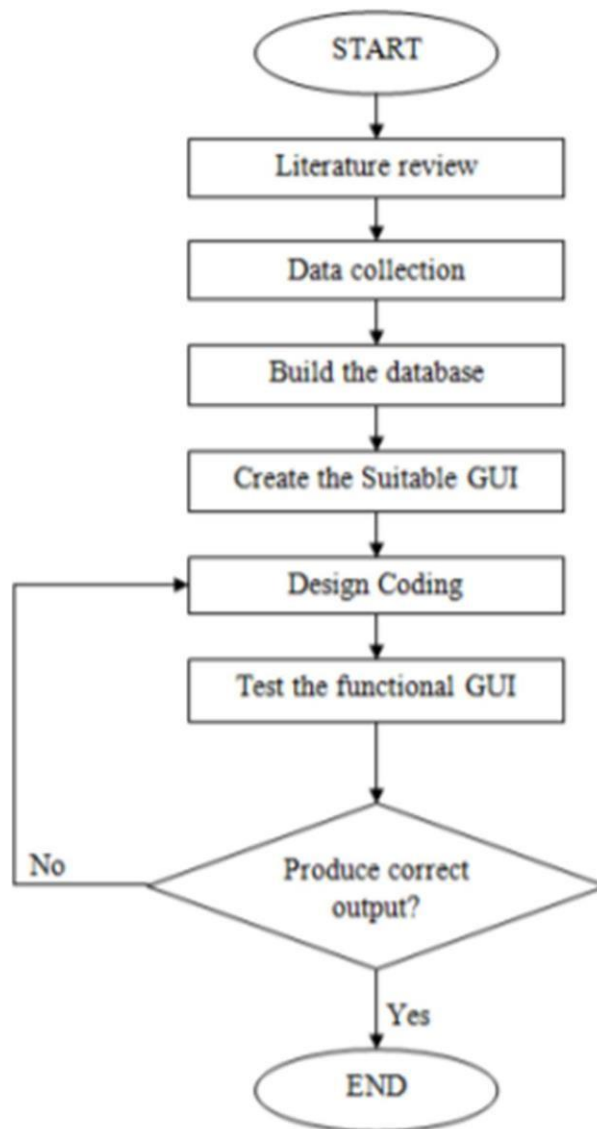


Fig 2.1: Methodology

2.1 LITERATURE REVIEW

- The project begins with literature review where the overview of the project needs to be reviewed. The literature review is done by finding out numbers of laboratory are involved in Lab Inventory Management.
- All this information must be known in order to provide information inside the software that is useful in arranging the database.
- The project has been divided into 2 modules:
 - o In-charge View
 - o User View

2.1.1 INCHARGE VIEW

All the instruments added and available on the LIMS is being taken care by the In-charge. The license is given by the In-charge. After getting the license, the user books the tool. The In-charge is the one who restricts the usage time

The dashboard displays a navigation pane on the left, and a contents pane on the right. The navigation pane shows the same hierarchy as a directory or subdirectory in the navigation pane, the data on the tabs relates to the agreements contained in the selected directory and all its subdirectories.

2.1.2 USER VIEW

The User books requests for the license, they will be able to book the tool only when the license is issued. Two users can't book the same instrument for same date and time. The user who has set a password and logged in at least once gets a user dashboard view. Your user dashboard group is meant to serve as your individual workspace.

2.2 DATA COLLECTION

- The data collection is done by obtaining how much equipment and component is inside the selected laboratory.

23 BUILD THE DATABASE

- Database is required to improve the functionality of this project. Therefore, the first stage of software development is to create the database. Database is a body of information made up by related pieces of data organized so that they can be easily manipulated by computer.
- As for software development, it is important to have the database so that all the information need will be saved for future used.

24 CREATE THE SUITABLE GUI WITH CODING

- Graphical User Interface (GUI) is a way of interfacing with a computer using pictures and other visual elements displayed on a computer screen. The pictures and buttons are used to control many functions. GUI implementation is important to make user easier to understand what they have to do in order to use the software.
- The GUI must be user friendly and easy to understand. In order to make the GUI function, coding be inserted to make the software work properly. The GUI for this problem is design using Bootstrap.

25 SOFTWARE TESTING

- Software testing is the process to measure the quality of developed computer software. The main intention is to determine whether the software meets its required results. It is used to finding, reduce and maybe can eliminate errors.

26 AIM & SCOPE

2.6.1 AIM

- A Laboratory Information Management System (LIMS) software allows you to effectively manage samples and associated data. LIMS can automate workflows, integrate instruments, and manage samples and associated information.

- Project for a system that is most adept at sending laboratory test orders to lab instruments and tracking those orders typically to a searchable database.

2.6.2 SCOPE

- A clearly-defined scope is set by the team which allows LIMS to accomplish its program goals while bound by a finite set of resources and an agreed-upon time frame. This project is for the creation of an efficient web application which facilitates laboratory instruments among students/ researchers.
- The user interface is be designed as part of the project where the users who has the license are allowed to order the desirable instruments depending on their availability. The incharge gives the license to the normal users and they are the one who lends the laboratory instrument to the normal users. Incharge can accept the request issued and can allow that specific user to use the instrument for that specific period of time.
- The most important aspect for data inventory is the safety. Thus, this project will improve data security. Using this invented LIMS database, all data about equipment and component record will be kept in a database. By using this database concept, the problems such as data loss and damage can be avoided.
- Capture the high-level requirements for the Laboratory Information Management Systems for the various types of laboratories in the provided / available lab.
- Provide the tools needed to select and modify as needed a specific solution(s) to meet user requirements.
- Identify detailed operating procedures of laboratories.
- LIMS helps eliminate errors by utilizing digital tracking through barcodes and consistent naming conventions.

CHAPTER 3

MATERIALS, METHODS AND ALGORITHMS USED

3.1 HARDWARE REQUIREMENTS

- ☐ PC
- ☐ RAM: 4 GB
- ☐ PROCESSOR: i3, i5 or equivalent

3.2 SOFTWARE REQUIREMENTS

- ☐ OS: Windows, Linux
- ☐ Visual Studio Code (code-editor)
- ☐ Python (Language)

3.3 SOFTWARE DESCRIPTION

- ☐ The purpose of the Software Requirement Specification is to produce the
- ☐ specification of the analysis task and also to establish complete information about
- ☐ the requirement, behavior and also the other constraint like functional performance
- ☐ and so on. The main aim of the Software Requirement Specification is to completely
- ☐ specify the technical requirements for the software product in a concise and in unambiguous manner.

3.3.1 DJANGO

- o The Django framework allows for most of the backend code and also provides.
- o Jinja templating which allows us to make the webpage dynamic. MySQL is used for the database handling tasks. It is a Structured Query Language which is connected to Python Django backend.

- o Django follows the MVT Architecture. Here, “M” stands for Model, “V” stands for View and “T” stands for Template.

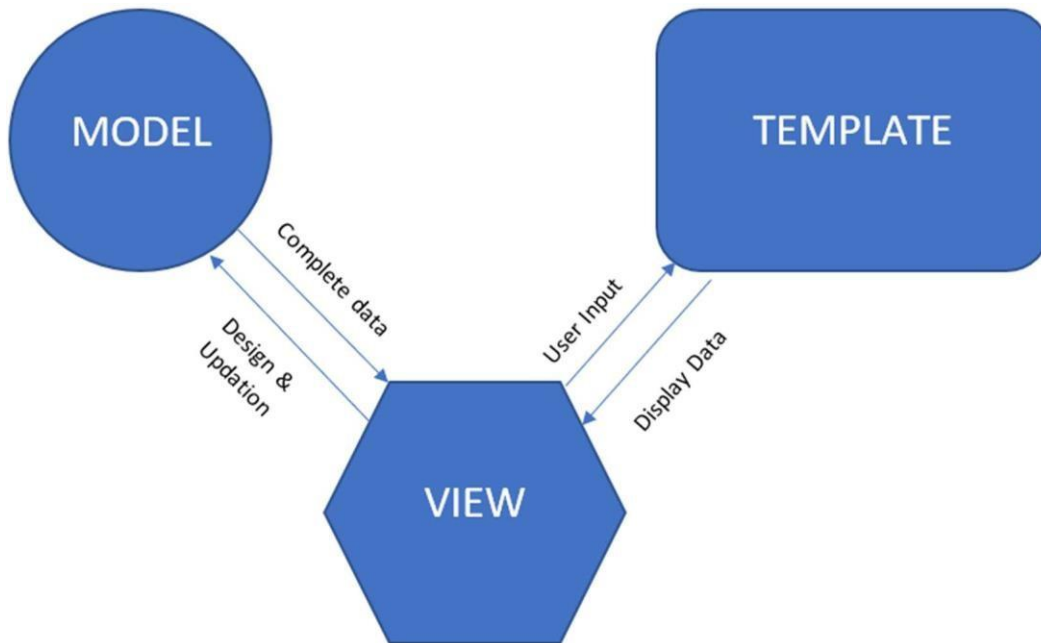


Fig 3.1:Django MVT Architecture

- o Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

```
from django.db import models

class Student(models.Model):
    student_id = models.CharField(max_length=15, primary_key=True)
    name = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    Class = models.CharField(max_length=3)
    dob = models.DateField()
```

Fig 3.2: Example Model Case

- o View: The View is the user interface (i.e.) what you see in your browser when

you render a website. It is represented by HTML / CSS / JavaScript and Jinja files. The view also has most of the logic behind our application. It is responsible for the dynamic data that appears on the webpage. It uses the `models.py` to extract the data from the database. It has many user-defined functions which take in a Web Request (directed from `urls.py`) and return a Web Response (could be a HTTP response, contents of a webpage, 404 Error etc....) that the browser can display.

```
def view_function(request):
    # Various logic needed for the application
    # Database calls, data manipulation etc...
    value = "Can be any type of data"
    value2 = {
        'reg_no': 3982900,
        'name': "Surya",
    }
    context = {
        'key': value,
        'key2': value2,
    }

    return render(request, "htmlFile.html", context)
```

Fig3.3:Example View Case

- o Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted (i.e., Jinja Templating).

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>

    <h1>Basic HTML static content</h1>
    <!-- Jinja Templating can be used to show dynamic data-->
    {{ key }}
    <!-- it can also be used for if statements, for loops etc... -->
    {% if key == 'Expected String' %}
        <h2>Show this</h2>
    {% endif %}
</body>
</html>
```

Fig 3.4: Example Template Code

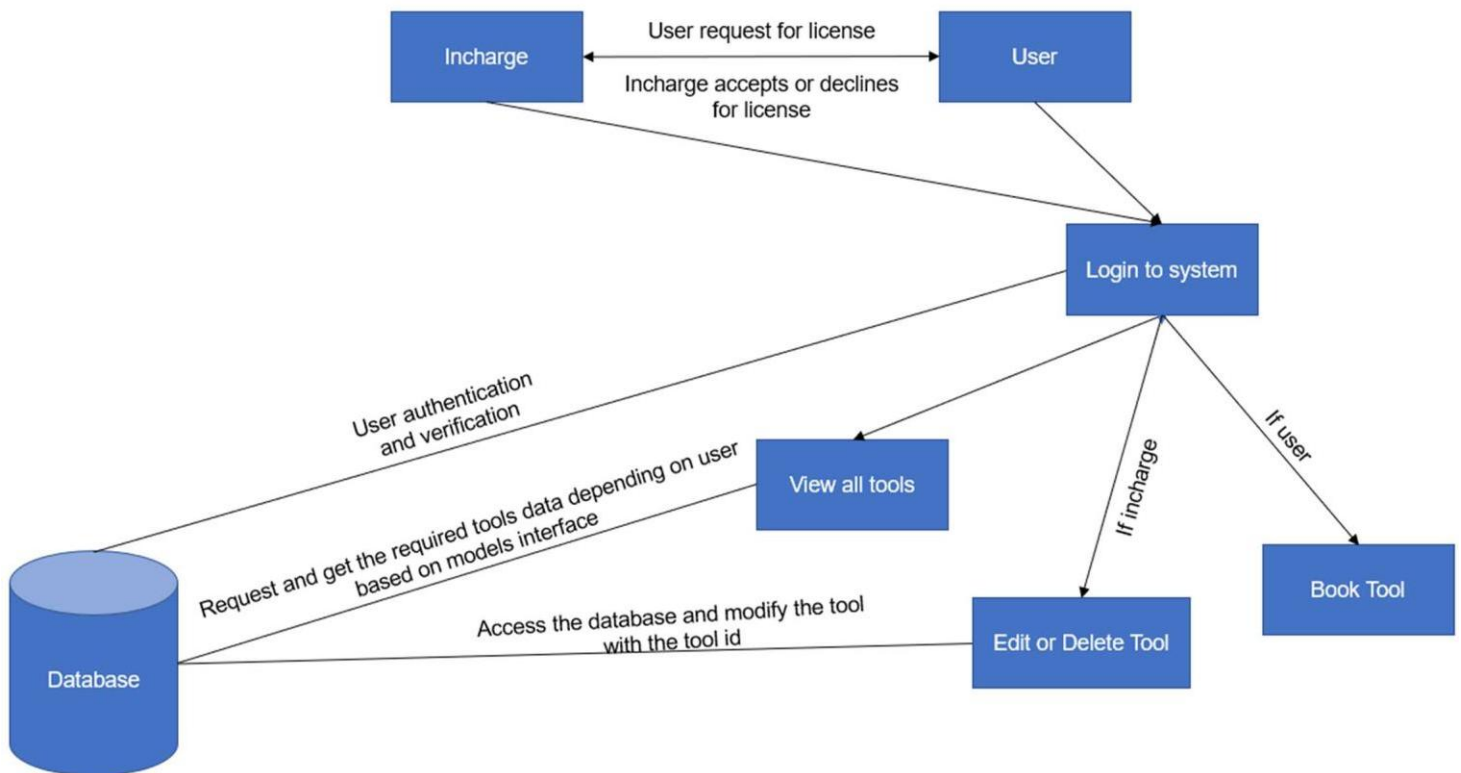
3.3.2 MySQL

MySQL is an open-source DBMS (DataBase Management System) which is built, supported and distributed by MySQL (ORACLE). It is a relational database management system based on SQL (Structured Query Language). The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

	tool_id	tool_name	department	controlled_by_id	lab_name	status	description	picture	created_at	updated_at	price	location	make	model
☐ Edit ☐ Copy ☐ Delete	1866909674	Trial	CSE	1234	Computer Science	Active	Isis	media/39110037_Akshay%20J.pdf	2021-08-21 07:21:42.939176	2021-08-21 07:21:42.954589	0.00	Sk	1	Dikd
☐ Edit ☐ Copy ☐ Delete	3124082068	Thermal Evaporator	CSE	1234	Computer Science	Active	gg	media/Org_wm9He6O.jpg	2021-08-21 07:52:26.631305	2021-08-21 07:52:26.635326	0.00	Chemistry Lab	Chem Co.	12A4D
☐ Edit ☐ Copy ☐ Delete	4065198048	Burette	CSE	1234	Computer Science	Inactive	test22epkkjhn	media/928e2462-01c9-4940-8ff8-7eebefd1cedd_EGihWuO...	2021-07-26 12:21:04.234066	2021-07-26 12:21:04.238055	0.00	Chemistry Lab	Chem Co.	Glass

Fig 3.5:Example SQL code

34 ARCHITECTURAL DIAGRAM



3.5 PROPOSED SYSTEM

The objective of the system is to make the lab instruments easily available for the students and other people. To design an efficient Laboratory management system that can ease the management, availability, and distribution of instruments for the people who need it, we need to follow various steps:

- i. We need to update the web page on the basis of the availability and unavailability of the instruments.
- ii. Assign and make the instruments available for the users who applied.
- iii. To issue an instrument the user will have to apply for a license and only after its approval the instrument will be booked under the user's name.
- iv. After the tenure for the assigning of the instruments is over, we can make the instruments available again and then assign the particular instrument to the other user.

3.5.1 ADVANTAGES OF PROPOSED SYSTEM

- i. Booking of Equipments at an optimum amount for students, faculties and also researchers.
- ii. Incharge handling this web application can book equipments for lab usage.
- iii. This application makes equipments in reach to other colleges as well.
- iv. Saves time.
- v. Booking is only possible for those who have license applied which makes it secured.

3.5.2 DISADVANTAGES OF PROPOSED SYSTEM

- i. Time to adapt to a new system, when starting up a computer system, it may seem inconvenient and unwieldy to laboratory staff

3.5.3 APPLICATIONS OF PROPOSED SYSTEM

- i. To have a record of the usage of equipments.
- ii. To share the instrument among the students/researchers to help them study their experiments.

36 PROJECT DESCRIPTION

- ☐ This project is mainly focused on lending the laboratory instruments among students/ researchers.
- ☐ The LIMS has two dashboards, one is for the user and the other one is for the incharge. The admin is the super user and he is the one that gives the access/license for the Incharge. The incharge gives the license to the normal users and he is the one who lends the laboratory instrument to the normal users.
- ☐ The users who have the license are only allowed to order the instruments. The incharge is also the user since he is the one who lends his instruments. The user can order the required instrument based on their availability.
- ☐ The user selects the time period he requires the instrument. The incharge can accept that request and can allow that specific user to use the instrument for that specific period of time.

37 THE CODE DESIGN

The templates are stored in the templates folder. Depending on the url request in the browser a specific function is called which is in the views.py file. The view function renders a specific template from the templates folder. The templates for user pages are in html2 folder and for the incharge pages are in html1. There are base.html files which are extended by the other HTML files when called.

Page	URLs	View Function
Login Page	/	index
	/login	login
Home page	/incharge_success	inchargeSuccess
	/user_success	userSuccess
Create User	/new_user	create_user
	/save_user	save_user
Adding tool page	/add_tool	addTool
	/save_tool	saveTool
All Tools page	/view_tools	viewTools
	/tool_active/<tool_id>	toolActive
	/tool_inactive/<tool_id>	toolInactive
Edit tool data	/edit_tool/<tool_id>	editTool
	/save_edit_tool/<tool_id>	saveEditTool
User Profile Page	/user_profile/<user_id>	userProfile
Calendar Page	/calendar	calendar
	/user_calendar	userCalendar
	/tool_calendar/<tool_id>	toolCalendar

Table 3.1

3.7.1 LOGIN PAGE:

The '/' URL calls "index" function in views.py file and renders the template from "templates/html1/login.html" as requested showing the login page and empty spaces to add credentials to get authentication to the web application.

On form submission, it redirects to "/login" with HTTP method POST which calls login function in views.py file and traverses its conditions by checking if the person logging in is an incharge or a user. If it is an incharge then it calls 'inchargeSuccess' function. If it is a user then it calls 'userSuccess' function. This Authentication is done by checking if the account details match with the already existing details in the database. The function redirects to the login page if the person is an invalid user.

3.7.2 HOME PAGE:

Incharge:

After signing in from `"/login"` URL, the page is redirected to `"/incharge_success"` URL in `urls.py`. Then `"inchargeSuccess"` function inside `views.py` file is called. In `views.py`, `"inchargeSuccess"` function is defined with `incharge_id` and `username` which is displayed in toggle bar. We get an array of tools from the database. Incharge has access to change the status of any particular instrument that he created. In the context part, tools are displayed with resource id and its availability status. The `"inchargeSuccess"` function will render the template which in the location `"templates/html1/incharge_success.html"`.

User:

Similarly, after signing in from `"/login"` URL, the page is redirected to `"/user_success"` which calls the `"userSuccess"` function inside `views.py` file. The `"userSuccess"` function is defined with `user_id` and `username` which is displayed in toggle bar. We get an array of tools from the database. In the context part, tools are displayed with resource id and its availability status. User have to click on `"Book Tool"` button to book a tool it will redirect to `"/book_tool"` URL. The `"userSuccess"` function renders that template located in `"templates/html2/view_tools.html"`.

3.7.3 CREATE USER:

URL: `/new_user`

This URL will call the `"create_user"` function in the `views.py` file which in turn simply renders the template from location `"templates/html2/create_user.html"`

The template has a form which on submission will redirect to `"/save_user"` with a HTTP POST method which then makes a database call to create a new user in the `"User"` table. After creating the user it redirects to the login page.

3.7.4 ADDING TOOL PAGE:

The `"/add_tool"` URL calls the `"addTool"` function in the `views.py` file. The function fetches the incharge data of the currently logged in incharge and renders the template located in `"templates/html1/add_tool"` and pre-fills the incharge details in the form. The

add tool page has a form which on submission redirects to “/save_tool” URL with HTTP method POST which calls the “saveTool” function in views.py file.

The “saveTool” function gets the data from the form and makes a database call to create a row in the Tools table with the data received form data. It then redirects to “addTool” page.

3.7.5 ALL TOOLS PAGE

URL: “/view_tools”

The “view_tools” URL calls the “viewTools” function in views.py where it checks the incharge_id and filters and gets an array of tools added by that incharge from the database.

The status of the instrument can be switched between active and inactive based on their availability and usage. When the incharge changes the status from inactive to active the URL is redirected “/tool_active/<tool_id>” which calls the “toolActive” function, there it checks the tool_id and updates the status to active by performing an update operation on the Tool table in the row with the tool_id. If the the status is updated to active it displays a message saying "Tool (tool_id) status is changed to Active". If it wasn't able to update then an error message is displayed saying "Failed to change status of the Tool (tool_id)" and then redirects back to the All Tools page. In the same way when the incharge changes the status from active to inactive the URL is redirected to “/tool_inactive/<tool_id>” URL which calls the “toolInactive” function, and the same process is executed for tool Inactive as tool Active.

3.7.6 EDIT TOOL DATA:

URL: “edit_tool/<tool_id>”

The “edit_tool/<tool_id>” calls the “editTool” function with an attribute “tool_id” which indicates the tool_id of the requested tool. The functions fetches the tool data from the database using this tool_id and then renders it on the template through the context variable. This function renders the template located in “templates/html1/edit_tool.html”. The rendered template has a form to edit the data of the requested tool. On clicking the “save tool” button it redirects to “save_edit_tool” URL with a HTTP POST method which calls the “saveEditTool” function. This function gets the edited data from the form

and performs a update operation on “Tools” table in the row where the tool with the requested tool_id belong. This function then calls the” toolData” function sending the request variable and tool_id as attributes and this function renders the template located in “templates/ html1/tool_data.html”.

3.7.7 USER PROFILE PAGE:

The 'user_profile' URL calls the “userProfile” function in views.py with a parameter “user_id” which gets the data of the User table with the user_id. We can the check if that user has license or not. If that user has license, the user details then the license is displayed in the User License field if the user doesn't have a license, then None is displayed in that field and is redirected to the user_profile URL. The “userProfile” function then renders the template in folder “templates/ html1/user_profile.html”.

3.7.8 CALENDAR PAGE

Calendar page:

"/calendar" URL calls the “calendar” function in views.py file. It will display start and end date and time of tool issued by the user department wise. It filters the orders by following:

- ☐ request_id
- ☐ requested_tool_id
- ☐ from_date
- ☐ to_date
- ☐ tool_name , lab_name

User Calendar Page:

"/user_calendar" URL calls “userCalendar” function in views.py file. It will display start and end date and time of tool issued by the user in their account. It filters the orders by following:

- ☐ user_id
- ☐ request_id
- ☐ requested_tool_id

- ☐ from_date
- ☐ to_date
- ☐ tool_name , lab_name

Tool Calendar Page:

"/tool_calendar" URL calls "toolCalendar" function in views.py file. It will display start and end date and time of the tools issued. It filters the tools by following:

- ☐ request_id
- ☐ requested_tool_id
- ☐ from_date
- ☐ to_date
- ☐ tool_name , lab_name

CHAPTER 4

RESULTS AND DISCUSSION

4.1 RESULTS

We have achieved our objective by creating a web application using the Django Framework for backend and HTML, CSS, JavaScript and Bootstrap combined for the frontend. We have an application in which a user can request for a particular license by the respective incharge. The incharge then accepts or denies that request.

After the user gets a license, he can book the particular tool for research purpose for a given period of time. The incharge can add, modify and delete tools under his control.

The tools have two modes,

1. Active: The tool is available for booking and is in working condition.
2. Inactive: The tool is not available for some unexpected reasons, maybe it is under maintenance or having some problems.

The project is hosted on <https://sathyabamabeta.pythonanywhere.com>

4.2 DISCUSSION

The performance of the proposed project depends mainly on the reserving of laboratory instruments. This helps many research faculties and students in their projects and experiments, this makes it possible for them to use the instruments for a specific period of time and return it back to its incharge. Instead of buying the instrument which could cost a large amount of money the user will be able to borrow the same instrument for a reasonable price and a limited period of time. This not only helps in building a user and incharge relationship but also would help in building a healthy community. It is merely like lending a book from a library where you lend the book for a particular period of time, read it and then return it back to the librarian. Same way instead lending a book a student/researcher lends a lab instrument, use it and then return it back when its period is over.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

This project mainly focuses on easily availing lab equipment for users and incharges. They can check if that particular instrument is available or not. Users can view and book the instrument for their choice of time and date. They have to get license from incharge for that particular lab before issuing their desired equipment. For Incharges, they have to approve license of the user by checking its validity. The particular instrument that user wishes to issue will be booked after checking it is already booked for that slot or not. Incharges can review history of users who borrowed the equipment. Users can view their license history of the instruments they already borrowed. By this project, we can help Users as well as Incharges to make use of equipment for their betterment.

5.2 FUTURE WORKS

In the upcoming works of this project in order to ease the process of reserving with making the utility of the equipment to its maximum. An Email system that facilitates users with notifications of their desired equipment. This interface will be designed as part of the project where the users who have the license are allowed to order the desirable equipment depending on their availability.

The bell prompt will be showing notifications within the web application when equipment is available, under maintenance, and unavailable along with successful transaction of the order placed.

The to-be Hosted payment model will allow to collect payment details from the user through an interaction hosted with listed banks and transaction modes such as debit and credit card transactions, online banking etc.

Once the user has made their choice for the type of payment mode, the page will be redirected to the payment method so they can authenticate and authorize the transaction. Once completed, they will be redirected back to the application along with the transaction success response.

5.3 REFERENCES

Since this is an inhouse project, there are no reference. Although we got ideas for the backend through geeksforgeeks website and official Django website.

1. <https://www.geeksforgeeks.org/django-project-mvt-structure/>
2. <https://docs.djangoproject.com/en/3.2/>
3. <https://getbootstrap.com/docs/4.6/>
4. <https://www.w3schools.com/>