GARBO

GARBAGE MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHARON SAJI
Reg No: 22PMC151

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

GARBO

GARBAGE MANAGEMENT SYSTEM

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHARON SAJI
Reg No: 22PMC151

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

A Project Report on

GARBO

GARBAGE MANAGEMENT SYSTEM

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

BY

SHARON SAJI
Reg No: 22PMC151

Under the guidance of
Sr. Italia Joseph Maria
Assistant Professor
PG Department of Computer Applications
Marian College Kuttikkanam (Autonomous)

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam (Autonomous)

Peermade, Kerala – 685 531

2023

PG DEPARTMENT OF COMPUTER APPLICATIONS

Marian College Kuttikanam (Autonomous)

[MAHATMA GANDHI UNIVERSITY, KOTTAYAM]
KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

“GARBO”

is a bonafide record of work done by

SHARON SAJI

Reg. No: -22PMC151

In partial fulfillment of the requirements for the award of Degree of

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2022 - 2023.

Sr. Italia Joseph Maria	**Mr. Win Mathew John**
Assistant Professor	**Head of the Department**
PG Department of Computer Applications Marian College Kuttikkanam (Autonomous)	**PG** Department of Computer Application Marian College Kuttikkanam (Autonomous)

External Examiner External Examiner

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I express my sincere gratitude to Prof Dr. Ajimon George, Principal, Marian College Kuttikkanam (Autonomous), Dr. Mendus Jacob, Director, PG Department of Computer Applications for the support given throughout the project work. I extend my gratitude to Mr. Win Mathew John, HoD, PG Department of Computer Applications, who is a constant source of inspiration and whose advice helped me to complete this project work successfully.

I express my deep sense of gratitude to my project guide, SR. ITALIA JOSEPH MARIA, Associate Professor/Assistant Professor, PG Department of Computer Applications, for her profound guidance for the successful completion of this project work.

With great enthusiasm, I express my gratitude to all the faculty members of the PG Department of Computer Applications for their timely help and support.

Finally, I express my deep appreciation to all my friends and family members for the moral support and encouragement they have given to complete this project work successfully.

SHARON SAJI

ABSTRACT

The Garbage Management System is a web-based application built using the Django framework that aims to streamline and enhance the management of garbage in a given locality or community. The traditional methods of waste disposal often lead to environmental pollution and health hazards. This project introduces a modern approach that utilizes technology to streamline waste collection, segregation, and disposal processes. The system provides a user-friendly interface for customers to schedule pick-ups for their household waste at a very affordable rate.

The system incorporates various features and functionalities to achieve these objectives. Users can register and log in to the system to select a package for collection, schedule pick-ups and see the status of their pick-ups. The system admin will be able to add packages, employees and also schedule employees to different pick-up locations. This project also consists of an employee interface where the employees will be able to see their assigned pick-ups and change the status of the pick-ups as and when he does any progress in it.

By implementing the garbage management system, several benefits can be achieved. It improves the cleanliness and aesthetics of the community, reduces the risk of diseases caused by improper waste disposal, and minimizes the environmental impact of waste

OBJECTIVE AND SCOPE

- The objective and scope of a Django project for a garbage management system is to effectively and efficiently handle the collection, transportation, treatment, and disposal of waste materials in order to minimize environmental pollution, health-risks, and resource depletion.
- Develop an efficient and user-friendly web application for managing garbage/waste-related operations.
- Automate and streamline key processes, such as waste collection scheduling and waste tracking.
- Implement user authentication mechanisms to ensure secure access to the application. Define different user roles (e.g., residents, waste collectors, administrators) with specific permissions and functionalities.

PROBLEM STATEMENT

The problem at hand is the inefficient management of garbage in a particular locality or city. The existing garbage management system lacks an organized and streamlined approach, leading to various issues such as improper waste disposal, increased pollution, and health hazards for the residents. Therefore, the objective is to develop a Django-based Garbage Management System that addresses these challenges and provides an efficient and sustainable solution.

TABLE OF CONTENTS

TABLE INDEX

1. INTRODUCTION

1.1 PROBLEM STATEMENTS

The problem at hand is the inefficient management of garbage in a particular locality or city. The existing garbage management system lacks an organized and streamlined approach, leading to various issues such as improper waste disposal, increased pollution, and health hazards for the residents. Therefore, the objective is to develop a Django-based Garbage Management System that addresses these challenges and provides an efficient and sustainable solution.

1.2 PROPOSED SYSTEM

The GARBO Website is an efficient user-friendly garbage management system which enables users to dispose their household waste in an environment friendly way. The system incorporates various features and functionalities to achieve these objectives. Users can register and log in to the system to select a package for collection, schedule pick-ups and see the status of their pick-ups. The system admin will be able to add packages, employees and also schedule employees to different pick-up locations. This project also consists of an employee interface where the employees will be able to see their assigned pick-ups and change the status of the pick-ups as and when he does any progress in it.

1.3 FEATURES OF THE PROPOSED SYSTEM

The features of this website are:

- Responsive website design.
- User-Friendly navigation.
- An user-friendly scheduling system.
- Facility for the user to edit his/her profile

2. FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

The functional requirements for this website include:

- ***Login and Signup***: The user should be able to register themselves using Django users. The user will be redirected to the user home page after logging in. The admin adds the employee using the Django admin interface. Both admin and employee will be redirected to their respective pages after logging in.
- ***Packages***: The user should have an option to choose from 3 different packages according to their liking.
- ***User Details***: Once the user logs in the user should be able to give their basic details like address, phone number etc. and select a subscription package for garbage collection.
- ***Date Scheduling***: The user should be able to schedule the pick-up time and date for garbage collection once they have chosen a subscription plan.
- ***Assigning employee for pick-up***: The admin should be able to assign an employee for the pick-ups scheduled by the users.
- ***Changing the delivery status***: The employee should be able to change the delivery status (scheduled, enroute, picked-up) as and when required once he gets assigned to a pick-up.
- ***History:*** The user should be able to see all their previous schedules.

3. NON-FUNCTIONAL REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for this website are:

- ***Usability***: The proposed website is simple, provides enough insight about features and packages, interactive, lets user select packages and schedule pick-ups and all this data is stored in the database.
- ***Reliability***: The system must perform without failure in 95 percent of use cases during a month.
- ***Maintainability***: The mean time to restore the system (MTTRS) following a system failure must not be greater than 10 minutes. MTTRS includes all corrective maintenance time and delay time.
- ***Availability***: Describes how likely the system is accessible to a user at a given point in time. A user-friendly system with global accessibility should be available around-the clock. In the event that the database is corrupted or the hardware fails, a replacement page will appear. Additionally, a database backup should be kept in case of hardware failure or database corruption.
- ***Security***: Database should be backed up every hour. Under failure, system should be able to come back at normal operation under an hour. All data must be stored, protected, or protectively marked.

4. FEATURES AND HIGHLIGHTS

FEATURES AND HIGHLIGHTS

The features and highlights of this project are:

- ***Login and registration***: The user will be able to register themselves using Django users. The user will be redirected to the user home page after logging in. The admin adds the employee using the Django admin interface. Both admin and employee will be redirected to their respective pages after logging in from the interface.
- ***Add user details:*** The user will be able to add his/her details after logging in and can also update their details whenever they would like to using the interface**.**
- ***Adding the packages:*** The admin will be able to add the packages using the django admin interface with an option to delete and update them as required.
- ***Selection of packages:*** The user will be able to select from the given packages once he/she has logged in.
- ***Scheduling the pick-up:*** The user will be able to schedule pick-up using a modal form. Each package will have a specified no of pick-ups. The user will only be able to make those many no of schedules for a particular month according to the package chosen.
- ***Employee Details***: The employees will be added directly by the admin and the is_staff permission will be enabled for the employees while adding them. The employee can login in using his/her mail id and password provided by the admin.
- ***Assigning the employees:*** The admin will be able to assign a particular employee to a scheduled task using the Django admin interface.
- ***Changing the status of a task:*** The employee will be able to see the task assigned by the admin on his interface and will be able to change the status (scheduled, enroute, order picked) of a particular task.

- ***Status updation on the user page:*** The user will be able to see the status of each schedule on their interface as and when the employee changes it.

- ***History:*** The user will be able to see all the previous pick-up history on his/her page to keep track of the no. of pick-ups scheduled by them.

4. THIRD PARTY LIBRARIES

THIRD-PARTY LIBRARIES

Third-party applications and libraries in Django are pre-built components or packages developed by the community or other companies that you can use to extend the functionality of your Django projects. These libraries provide pre-built solutions for common tasks, saving developers time and effort in implementing certain features from scratch. They are designed to seamlessly integrate with Django and follow its best practices.

Third-party libraries can be installed using package managers like pip, and they usually come with their own documentation and examples to guide developers in their usage. These libraries can cover a wide range of functionalities

The third-party libraries used in this project are:

- ***Django Crispy forms:*** Django Crispy Forms is a third-party library for Django that helps in rendering Django forms in a more concise and elegant way. It provides a simple and flexible way to control the layout, styling, and rendering of forms in Django projects. A library that helps you create beautiful, responsive forms using Bootstrap, Foundation, or other CSS frameworks. Crispy Forms enables you to easily customize the rendering of form controls such as fields, labels, buttons, and form errors. You can control the CSS classes, HTML attributes, and overall appearance of each form element. I have also installed a Bootstrap5 template pack for django-crispy-forms.
- ***Django jazzmin:*** Django Jazzmin is a third-party library for Django that provides an improved admin interface. It is a modern, responsive, and customizable replacement for Django's default admin interface, a drop-in app to jazz up your Django admin site, with plenty of things you can easily customize, including a built-in UI customizer.

5. DATABASE DESIGN

DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. It is a collection of interrelated data stored with minimum redundancy to serve many users quicklyand efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user. It is the most widely used relational database. It offers various features and provides users with many niceties. Computer databases can store data in different forms from simple lines of text to complex data structure that includes pictures, sounds or video images. Data management involves creating, modifying, deleting and adding data in files and using this data to generate reports. The software that allows performing this function is known as a database management system.

PACKAGES TABLE

```
class packages(models.Model):
    pack_name = models.CharField(max_length=100)
    pack_duration = models.CharField(max_length=100)
    pack_price = models.IntegerField()
    pack_details = models.TextField()
    day_pickup = models.CharField(max_length=100)
```

USER MEMBER TABLE

```
class user_member(models.Model):
    member = models.ForeignKey(User, on_delete=models.CASCADE)
    pack = models.ForeignKey(packages, on_delete=models.CASCADE)
    addr = models.CharField(max_length = 150)
    place = models.CharField(max_length = 50)
    pin_code = models.IntegerField()
    phone = models.IntegerField()
```

DELIVERY STATUS TABLE

```
class delivery_status(models.Model):
    status = models.CharField(max_length=10)
    ui_text = models.CharField(max_length=100)
```

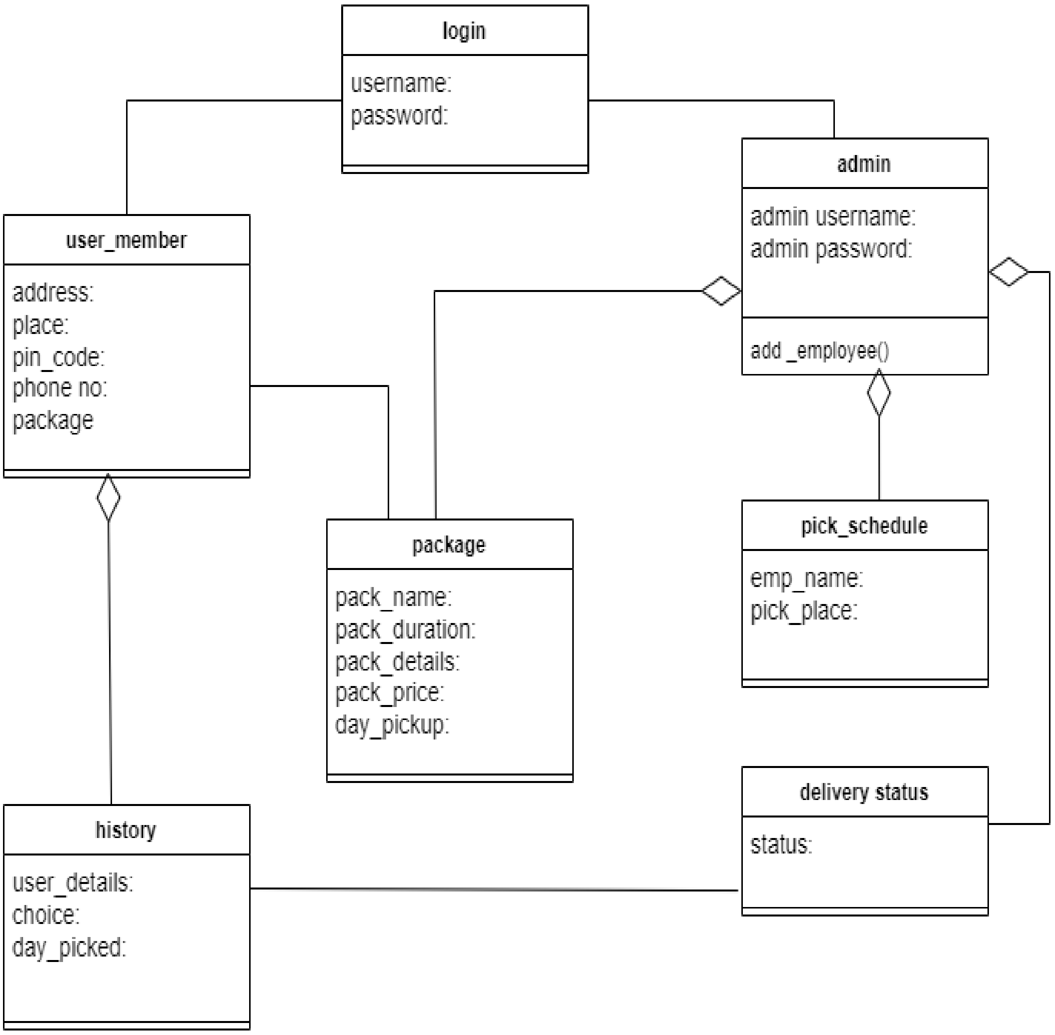
HISTORY TABLE

```
class history(models.Model):
    user_details = models.ForeignKey(user_member, on_delete=models.CASCADE)
    choice = models.ForeignKey(delivery_status, on_delete=models.CASCADE)
    day_picked = models.DateTimeField()
```

PICKER SCHEDULE TABLE

```
class picker_schedule(models.Model):
    emp_name = models.ForeignKey(User, on_delete=models.CASCADE)
    pick_place = models.ForeignKey(history, on_delete=models.CASCADE)
```

6.1.CLASS DIAGRAM



7. CHALLENGES

CHALLENGES FACED

The challenges faced during the project:

The major challenge faced by me was to plan the process flow of the project. While coming to the coding part I had difficulties in using the Django user for registering customers as it was new to me. The next major challenge which I faced was during the integration of crispy-forms in my project. Even though it made my work easy by helping me from the hectic task of creating html pages, I did require some time to learn about the working of it. I also had difficulty in making a change from function-based views to class-based views. I had to refer to django documentation and other websites for this conversion. Class-based views did help me in defining many functions together in one class rather than creating different def function for a particular page.

8.FUTURE ENHANCEMENTS

FUTURE ENHANCEMENTS

- A payment portal can be added in the future for paying for the packages. Offer various payment options, including credit/debit cards, digital wallets (e.g., PayPal, Apple Pay), or alternative payment methods like bank transfers. Allow users to select their preferred payment method and provide relevant instructions or interfaces for each option.
- Live tracking can be integrated to the website by which the users will be able to know the correct location of the garbage collector while making the collection. Implementing live tracking of employees as a future enhancement in the project can provide various benefits, such as improved productivity, enhanced security, and efficient resource management.
- Introduction of a mobile application can also be done so that this system becomes more user friendly.

9.CONCLUSION

CONCLUSION

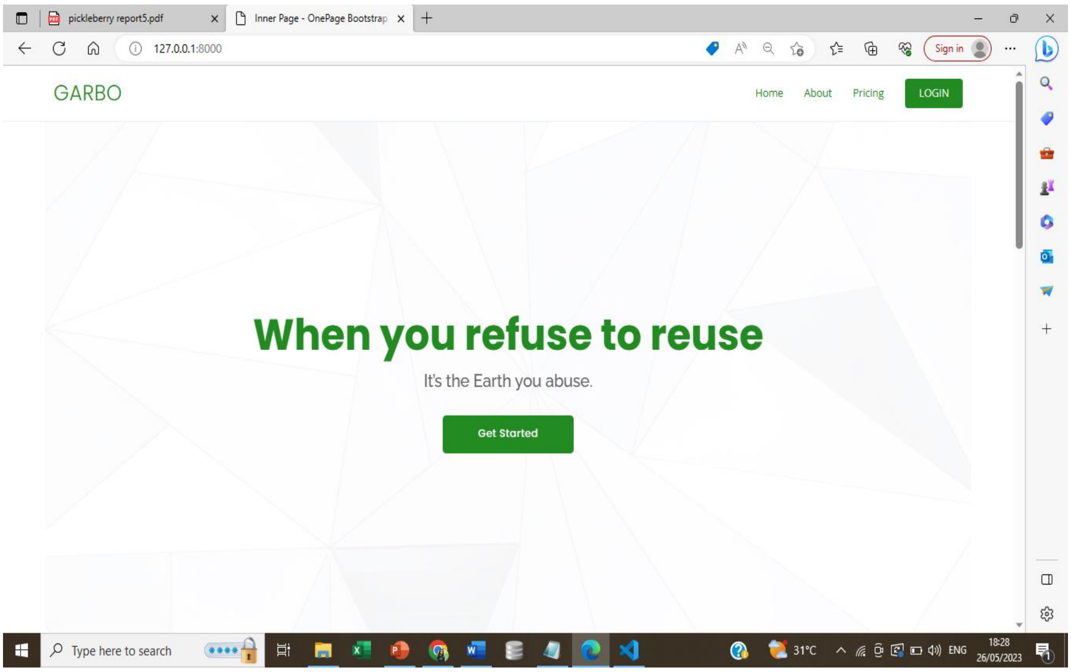
Garbo is a web-based application which helps customers to efficiently dispose their household wastes. This website consists of a user-friendly interface from where he/she will be able to choose from different package options given by the website. The website provides the user with an option to schedule pick-ups as per their convenience. Not only that the user will also be able to see the status of their scheduled pick-ups. Apart from the users the admin and employee also play a vital role in this website. Overall, a garbage management system is essential for effective waste management and environmental sustainability. With the increasing population and urbanization, proper management of garbage has become a critical issue worldwide. Implementing a garbage management system using Django can streamline the process and ensure efficient handling of waste.

10.REFERENCES

REFERENCES

- https://dev.to/thearjun/integrate-django-jazzmin-theme-to-django-admin-adminlte-dashboard-5aao
- https://docs.djangoproject.com/en/4.2/
- https://getbootstrap.com/docs/5.0/getting-started/introduction/
- https://pypi.org/project/django-crispy-forms/

11.ANNEXURE



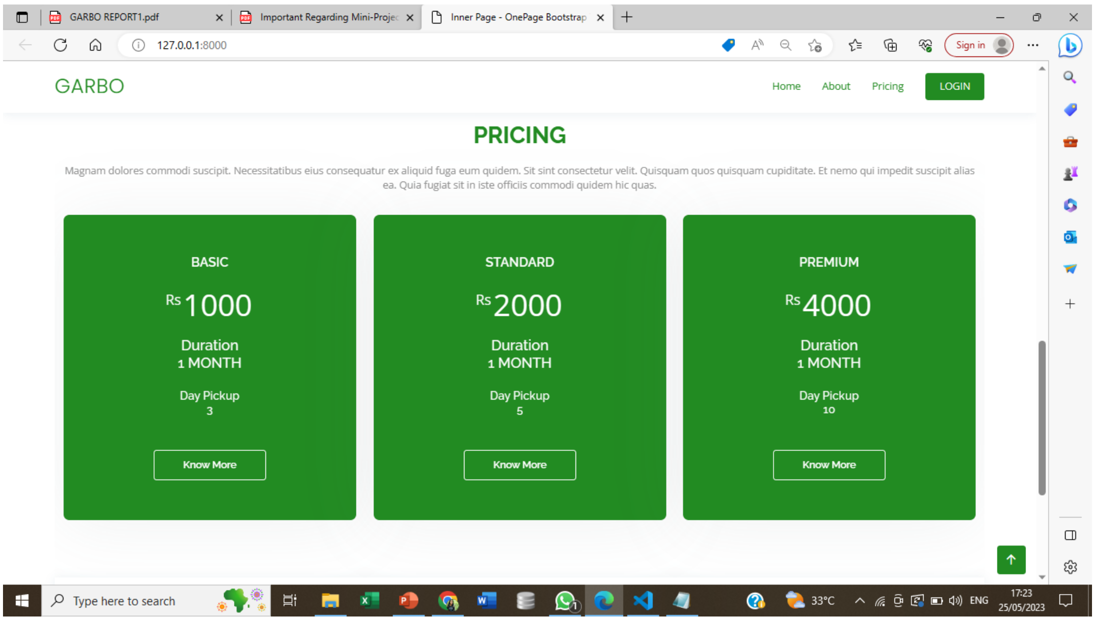
GARBO

Home About Pricing LOGIN

When you refuse to reuse

It's the Earth you abuse.

Get Started



GARBO

Home About Pricing LOGIN

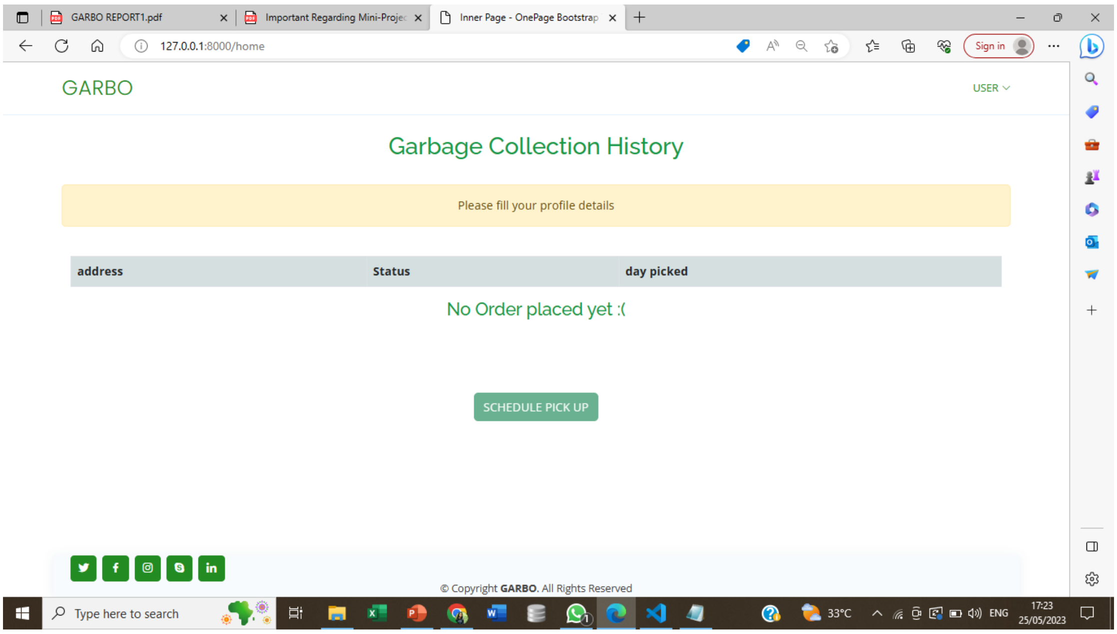
PRICING

Magnam dolores commodi suscipit. Necessitatibus eius consequatur ex aliquid fuga eum quidem. Sit sint consectetur velit. Quisquam quos quisquam cupiditate. Et nemo qui impedit suscipit alias ea. Quia fugiat sit in iste officiis commodi quidem hic quas.

BASIC
Rs 1000
Duration
1 MONTH
Day Pickup
3
Know More

STANDARD
Rs 2000
Duration
1 MONTH
Day Pickup
5
Know More

PREMIUM
Rs 4000
Duration
1 MONTH
Day Pickup
10
Know More



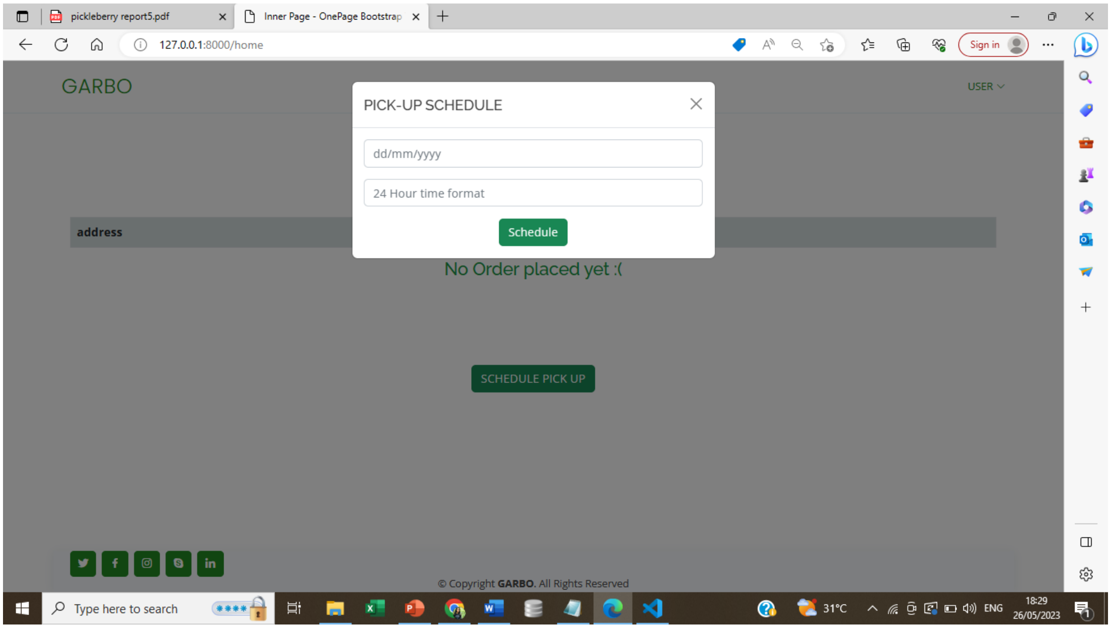
Garbage Collection History

Please fill your profile details

address	Status	day picked

No Order placed yet :(

© Copyright **GARBO**. All Rights Reserved



PICK-UP SCHEDULE

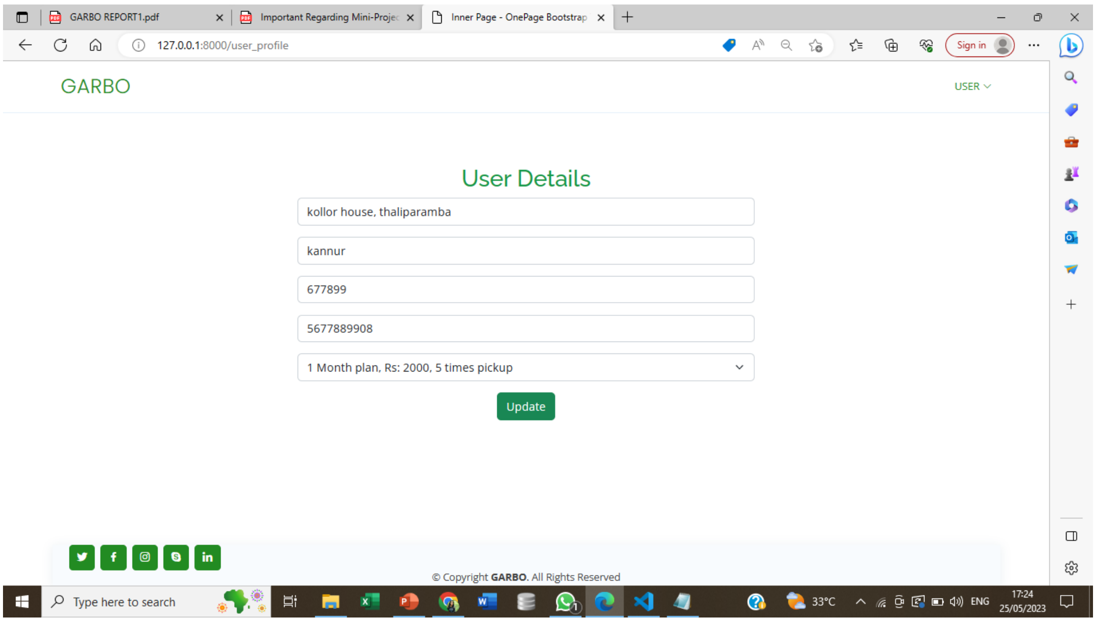
dd/mm/yyyy

24 Hour time format

address

No Order placed yet :(

© Copyright **GARBO**. All Rights Reserved


GARBO REPORT1.pdf
Important Regarding Mini-Proje
Inner Page - OnePage Bootstrap
127.0.0.1:8000/user_profile
Sign in
GARBO
USER
Update
Type here to search
33°C
ENG
17:24
25/05/2023

User Details

kollor house, thaliparamba

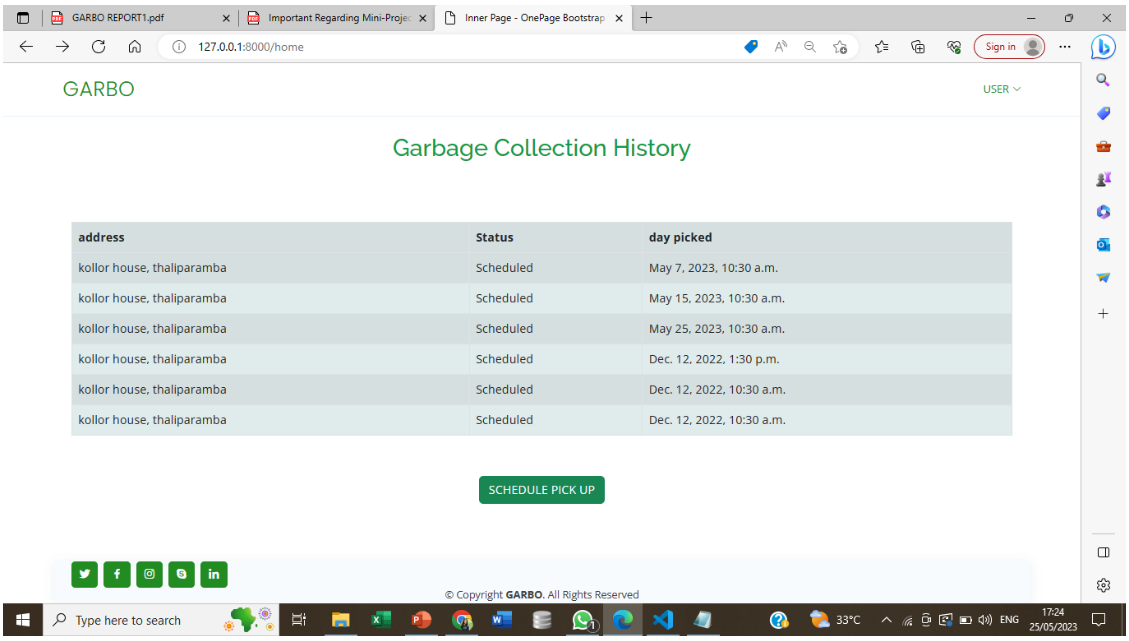
kannur

677899

5677889908

1 Month plan, Rs: 2000, 5 times pickup

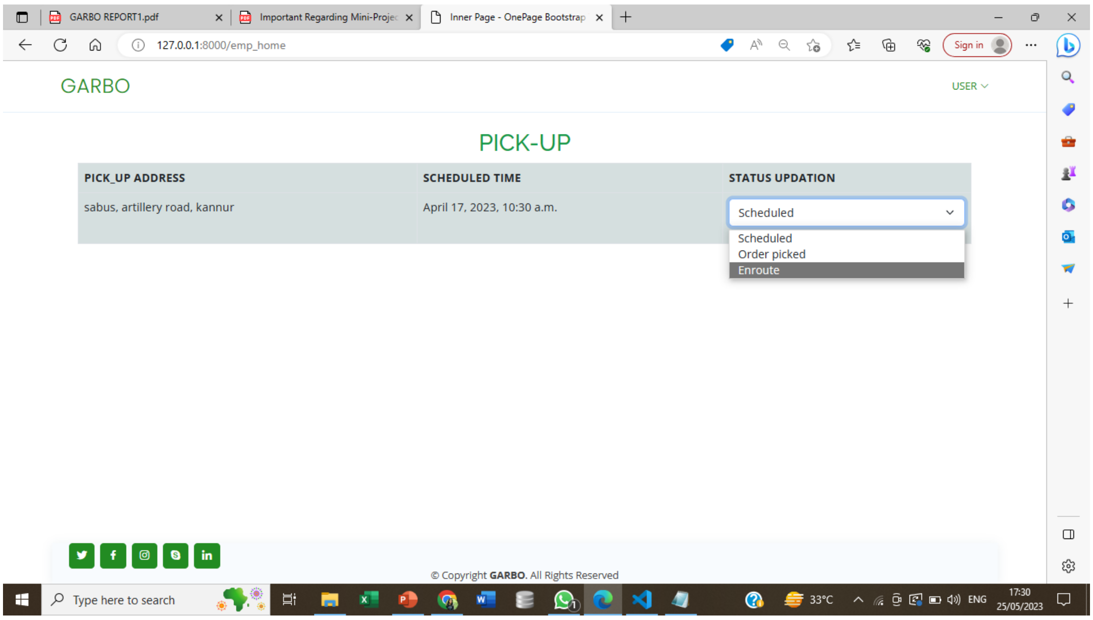
© Copyright **GARBO**. All Rights Reserved



Garbage Collection History

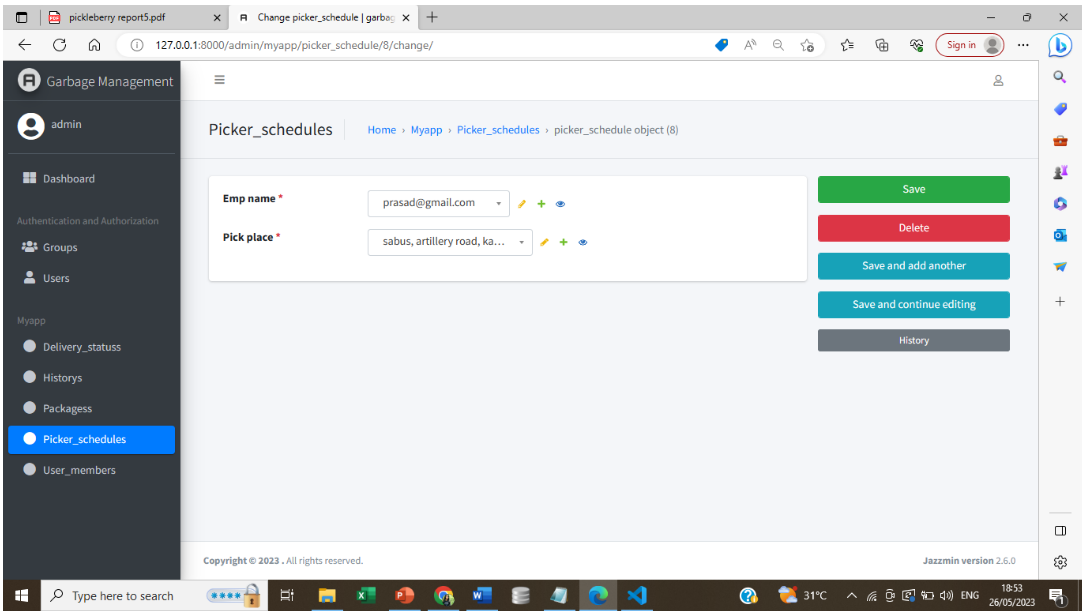
address	Status	day picked
kollor house, thaliparamba	Scheduled	May 7, 2023, 10:30 a.m.
kollor house, thaliparamba	Scheduled	May 15, 2023, 10:30 a.m.
kollor house, thaliparamba	Scheduled	May 25, 2023, 10:30 a.m.
kollor house, thaliparamba	Scheduled	Dec. 12, 2022, 1:30 p.m.
kollor house, thaliparamba	Scheduled	Dec. 12, 2022, 10:30 a.m.
kollor house, thaliparamba	Scheduled	Dec. 12, 2022, 10:30 a.m.

© Copyright **GARBO**. All Rights Reserved


GARBO REPORT1.pdf
Important Regarding Mini-Projec
Inner Page - OnePage Bootstrap
127.0.0.1:8000/emp_home
Sign in
GARBO
USER

PICK-UP

PICK_UP ADDRESS	SCHEDULED TIME	STATUS UPDATION
sabus, artillery road, kannur	April 17, 2023, 10:30 a.m.	Scheduled



Picker_schedules

Home › Myapp › Picker_schedules › picker_schedule object (8)

Emp name * prasad@gmail.com

Pick place * sabus, artillery road, ka...