CELLFIX ONLINE

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

LIBIN JACOB

Reg. No: 22PMC135

MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2023

CELLFIX ONLINE

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

LIBIN JACOB

Reg. No: 22PMC135

MARIAN COLLEGE
KUTTIKKANAM
(AUTONOMOUS)

MAKING COMPLETE

Marian College Kuttikanam Autonomous

Peermade, Kerala – 685 531

2023

A Project Report on

CELLFIX ONLINE

SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT

FOR THE AWARD OF THE DEGREE

MASTER OF COMPUTER APPLICATIONS (MCA)

OF

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

LIBIN JACOB

Reg. No: 22PMC135

Under the guidance of

Ms. Reny Jose
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 Kuttikkanam Autonomous

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

KUTTIKKANAM – 685 531, KERALA.

CERTIFICATE

This is to certify that the project work entitled

"CELLFIX ONLINE"

is a bonafide record of work done by

LIBIN JACOB

Reg. No: 22PMC135

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

MASTER OF COMPUTER APPLICATIONS [MCA]

During the academic year 2022-2023

Ms. Reny Jose
Assistant Professor
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

Mr Win Mathew John
Head of the Department
PG Department of Computer Applications
Marian College Kuttikkanam Autonomous

Examiner

Examiner

ACKNWOLEDGEMENT

I would like to take this opportunity to express my heartfelt gratitude to all those who have contributed to the successful completion of my project. Their unwavering support, guidance, and encouragement have been instrumental in shaping my journey.

First and foremost, I extend my deepest thanks to the "God Almighty" for His boundless grace and blessings, which have been a constant source of strength throughout this endeavor.

I am immensely grateful to Prof. Dr. Ajimon George, the Principal of Marian College Kuttikkanam (Autonomous), and Dr. Mendus Jacob, the Director of the PG Department of Computer Applications, for their unwavering support and guidance during the entire course of this project.

I would like to express my sincere appreciation to Mr. Win Mathew John, the Head of the PG Department of Computer Applications, whose constant motivation and valuable advice have played a crucial role in the successful completion of this project.

A special word of thanks goes to my project guide, Ms. Reny Jose, Associate Professor/Assistant Professor of the PG Department of Computer Applications, for her profound guidance and invaluable insights throughout this project. Her expertise and encouragement have been truly inspiring.

I would also like to extend my gratitude to all the faculty members of the PG Department of Computer Applications for their timely assistance and support, which have been instrumental in overcoming various challenges during this project.

Last but not least, I would like to express my deep appreciation to my friends and family members for their unwavering moral support and constant encouragement. Their presence and belief in me have motivated me to persevere and successfully complete this project.

Once again, I would like to express my heartfelt gratitude to all those mentioned above, as well as anyone else who has contributed to the realization of this project. Your support has been invaluable, and I am truly grateful.

Sincerely,

Libin Jacob

ABSTRACT OF CELLFIX ONLINE

This project documentation outlines the development of a smartphone service center management system named CELLFIX ONLINE using the Django framework. The system caters to three types of users: Admin, Customer, and Mechanic. The Admin has a comprehensive dashboard displaying key statistics and information, including the total number of customers, mechanics, enquiries, feedback, and recent customer enquiries. The Admin can manage customers by viewing, adding, and tracking customer enquiries and invoices. Mechanic management includes viewing, adding, and approving new mechanics, as well as tracking mechanic salaries and attendance. The Admin can assign mechanics to customer requests, set service costs, and change request statuses. Additionally, the Admin can personally make requests, update problem details, and assign mechanics. The system allows for modifying service costs before completion. Once a mechanic completes their work, they update the status and send a report to the Admin for review. Feedback from both customers and mechanics can be viewed and managed by the Admin. The system includes authentication for the Admin using a username and password.

Customers can create accounts by providing their personal details, and they can log in using their chosen username and password. The Customer's homepage features a dashboard displaying new requests, mobile repairs in progress, mobiles repaired, and total bills. Customers can access various features, such as viewing pending requests, making new requests, and viewing approved requests and bills. The system allows customers to view and delete pending requests, create new requests by specifying mobile details and problem descriptions, and view approved requests and bills. Feedback can be submitted to the Admin, and customers have the ability to edit and update their profiles.

Mechanics can apply for jobs by providing their personal details, skills, and credentials. After approval from the Admin, mechanics can log in to the system using their username and password. The Mechanic's homepage mirrors the Customer and Admin interfaces, displaying new works assigned, works in progress, works completed, and current salary. Mechanics can access and update the status of assigned works, view their current salary, and review their attendance records. Feedback can be submitted to the Admin, and mechanics can edit and update their profiles.

Overall, this Django-based system provides a comprehensive platform for managing smartphone service center operations, enabling efficient communication and workflow management among Admin, Customers, and Mechanics.

TABLE OF CONTENTS

CONTENT	PAGE NO
1. INTRODUCTION	1
1.1 PROBLEM STATEMENTS	2
1.2 PROPOSED SYSTEM	4
1.3 FEATURES OF THE PROPOSED SYSTEM	5
2. FEATURES AND HIGHLIGHTS	6
3. FUNCTIONAL REQUIREMENTS	9
4. NON- FUNCTIONAL REQUIREMENTS	12
5. TECHNICAL ASPECTS	14
5.1 ARCHITECTURE OF THE PROJECT	12
5.2 THIRD PARTY LIBRARIES USED	17
6. SYSYTEM DESIGN	18
6.1 INPUT DESIGN	19
6.2 OUTPUT DESIGN	22
6.3 DATABASE DESIGN	24
6.3.1 CLASS DIAGRAM	24
7. CHALLANGES FACED	26
8. FUTURE ENHANCEMENT	27
9. CONCLUSION	30
10. REFERENCES	32
11. APPENDIX	34
11.1 SCREEN SHOTS	35

1. INTRODUCTION

1.1 PROBLEM STATEMENTS

The smartphone service center currently faces several challenges due to the lack of a centralized system. Without a centralized framework, the management of customer requests, mechanics, and administrative tasks becomes disorganized and inefficient. Additionally, the absence of user authentication compromises the security and confidentiality of sensitive information. To enhance security, it is crucial to implement secure login credentials for admin, customers, and mechanics.

The admin's home page suffers from an inefficient dashboard that lacks essential metrics. This absence of a clear overview of total customers, mechanics, inquiries, and feedback hampers decision-making and monitoring of the center's performance.

Moreover, customer management within the current system is tedious for the admin. Tasks such as viewing, adding, and checking customer inquiries, as well as handling customer invoices, are cumbersome and time-consuming.

Similarly, the system's mechanisms for managing mechanics are ineffective. Admin lacks efficient features for viewing, adding, and approving new mechanics, as well as monitoring mechanic salary and attendance records. Furthermore, the process of assigning mechanics to customer requests is manual and lacks transparency, resulting in delays and improper assignments.

The existing system also fails to provide customers with the ability to track the progress of their mobile repairs or view their bills. This lack of customer request tracking leads to a lack of transparency and potential frustration. Moreover, the system does not allow customers and mechanics to provide feedback on their experiences, hindering the improvement of service quality and identification of areas for enhancement.

Mechanics face challenges accessing relevant information about their assigned tasks, such as work status and current salary. They are also unable to view their attendance records or provide feedback. Additionally, the current system lacks a standardized approval process for mechanics, resulting in confusion and potential security risks.

The system also lacks efficient profile management for both customers and mechanics. Without the ability to efficiently manage and update their profiles, the accuracy and completeness of their personal information suffer.

These problems collectively contribute to low customer satisfaction, high operational costs, poor communication, and a lack of transparency and accountability. To address these issues, a proposed Django framework project aims to develop a comprehensive system. This system will cater to the needs of admin, customers, and mechanics by providing streamlined interfaces, user authentication, efficient management functionalities, transparent request assignment, tracking capabilities, feedback mechanisms, and improved profile management. Through these enhancements, the project aims to improve the service quality and efficiency of the smartphone service center.

1.2 PROPOSED SYSTEM

The proposed system aims to address the existing challenges faced by the smartphone service center by developing a centralized framework using the Django framework. The system will provide enhanced security through user authentication for admin,customers, and mechanics. It will also offer streamlined interfaces and efficient management functionalities for improved customer and mechanic management. Additionally, the system will incorporate features such as transparent request assignment, request tracking, feedback mechanisms, and improved profile management. These enhancements will contribute to increased customer satisfaction, reduced operational costs, improved communication, and enhanced transparency and accountability.

The lack of a centralized system in the existing smartphone service center poses several challenges. Without a centralized framework, the management of customer requests,mechanics, and administrative tasks becomes disorganized and inefficient. The absence of user authentication compromises the security and confidentiality of sensitive information.

Furthermore, the admin's home page suffers from an inefficient dashboard lacking essential metrics, hindering decision-making and monitoring of the center's performance. Customer and mechanic management within the current system is tedious and time-consuming for the admin, lacking efficient features for viewing, adding, approving, and monitoring relevant information. The assignment of mechanics to customer requests is manual and lacks transparency, resulting in delays and improper assignments. Moreover, customers lack the ability to track the progress of their repairs or view their bills, leading to a lack of transparency and potential frustration. Feedback collection from customers and mechanics is also absent, hindering service quality improvement and identification of areas for enhancement. Mechanicrelated challenges include limited access to relevant information, absence of attendance records and feedback provision, and a lack of standardized approval processes. The existing system also lacks efficient profile management for both customers and mechanics, impacting the accuracy and completeness of personal information.

1.3 FEATURES OF PROPOSED SYSTEM

A web application built using Django framework that allows three types of users: admin, customer and mechanic to manage the services of a smartphone service centre.

A user authentication system that requires username and password for each user type and provides different access levels and functionalities for each user type.

A database system that stores the information of customers, mechanics, requests, invoices, feedbacks, attendances and salaries.

A user interface that consists of a homepage with a dashboard of cards displaying relevant information for each user type and a sidebar with links to access different features of the system.

A request management system that allows customers to make new requests by providing details of their mobile problems and allows admin to view, assign, approve or reject requests and set service costs. It also allows mechanics to view and update the status of their assigned requests and send reports to admin.

An invoice management system that allows admin to view and generate invoices for completed requests and allows customers to view their approved request bills.

A feedback management system that allows customers and mechanics to send feedbacks to admin and allows admin to view and remove feedbacks.

A profile management system that allows users to view and edit their personal details and profile pictures.

A mechanic management system that allows admin to view, add, approve or remove mechanics, take their attendances and view their salaries. It also allows mechanics to view their attendances and current salaries.

2. FEATURES AND HIGHLIGHTS

FEATURES AND HIGHLIGHTS

1. *User Types*

- Admin: Responsible for managing the entire system.
- Customer: Can make service requests, view invoices, and provide feedback.
- Mechanic: Assigned to repair and complete service requests.

2. *Admin Features*

- Home Page Dashboard: Displays key information such as the total number of customers, mechanics, inquiries, feedback, and recent customer inquiries.
- Customer Management: Admin can view, add, and check inquiries made by customers, as well as view customer invoices.
- Mechanic Management: Admin can view, add, and approve new mechanics, view mechanic salaries, and take mechanic attendance.
- Request Management: Admin can view all customer requests, assign mechanics, set service costs, and change request status (e.g., pending to approved).
- Self-Request: Admin can create a request on behalf of a customer, collect problem details, and assign an available mechanic and service cost.
- Service Cost Modification: Admin can change the service cost before the completion of the work.
- Work Reports: Admin can view work status reports submitted by mechanics.
- Feedback Management: Admin can view feedback from customers and mechanics, and has the ability to remove them.

3. *Customer Features*

- Home Page Dashboard: Displays the count of new requests, repairs in progress, completed repairs, and total bill.
- View Pending Requests: Customers can view and delete their pending service requests.
- Make a Request: Customers can create new service requests by providing mobile details and problem descriptions.

- Approved Requests: Customers can view their approved service requests.
- Approved Request Bill: Customers can view the bill for approved service requests.
- Feedback Submission: Customers can provide feedback to the admin.
- Profile Management: Customers can edit and update their profiles.

4. *Mechanic Features*

- Home Page Dashboard: Displays the count of new assigned works, works in progress, completed works, and current salary.
- Assigned Works: Mechanics can access and update the status of assigned works (e.g., approved, repairing, repairing done).
- Salary Information: Mechanics can view their current salary.
- Attendance View: Mechanics can view their attendances as recorded by the admin.
- Feedback Submission: Mechanics can send feedback to the admin.
- Profile Management: Mechanics can edit and update their profiles.
- These features and highlights encompass the core functionalities of the smartphone service centre management system, providing efficient management and communication between the admin, customers, and mechanics.

3. FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

1. *User Management*

The system should support three types of users: Admin, Customer, and Mechanic.

Users should be able to create accounts by providing relevant information such as first name, last name, address, mobile number, username, password, and profile picture.

Users should be able to log in using their username and password.

2. *Admin Functionality*

- Dashboard displaying the count of total customers, mechanics, enquiries, and feedback.
- Manage customers by viewing, adding, and checking enquiries made by customers.
- Manage mechanics by viewing, adding, approving new mechanics, viewing mechanic salary, and taking mechanic attendance.
- View mechanic attendance and mark them as present or absent for a specific date.
- View all requests from customers, assign a mechanic and service cost, and change the status of the requests.
- Make a request on behalf of a customer, update the problem details, and assign an available mechanic and service cost.
- Change the service cost before the completion of the work in progress by the mechanic.
- After the mechanic completes the work, the Admin should be able to view the work status report.
- View feedback from customers and mechanics and remove them if necessary.
- Username and password authentication for secure access.

3. *Customer Functionality*

- Dashboard displaying the count of new requests, mobile repairs in progress, mobiles repaired, and total bill.
- View pending requests, approved requests, and their invoices.
- Make new requests by providing mobile details, problem description, and images.
- View and delete their pending requests.
- View approved requests and their bills.
- Send feedback to the Admin.
- Edit and update their profile information.

4. Mechanic Functionality

- Apply for a job by providing relevant details.
- Log in using their username and password after approval from the Admin.
- Dashboard displaying the count of new assigned works, works in progress, works completed, and current salary.
- View and update the status of assigned works.
- View their current salary.
- View their attendance records.
- Send feedback to the Admin.
- Edit and update their profile information.

4. NON FUNCTIONAL REQUIREMENTS

NON FUNCTIONAL REQUIREMENTS

1. Security

The system should protect the data and resources from unauthorized access and malicious attacks. The system should use encryption, authentication, authorization, and auditing mechanisms to ensure security.

2. Performance

The system should respond quickly and efficiently to user requests and handle high volumes of traffic and data. The system should use caching, load balancing, database optimization, and performance testing tools to improve performance.

3. Usability

The system should provide a user-friendly and intuitive interface that meets the needs and expectations of the users. The system should use Django templates, bootstrap, CSS, and JavaScript to enhance usability.

4. Scalability

The system should be able to handle increased demand and workload without compromising performance or functionality.

5. Portability

The system should be able to run on different platforms and environments without requiring major changes or adaptations. The system should use cross-platform libraries, virtual environments, and requirements.txt files to ensure portability.

5. TECHNICAL ASPECTS

5.1 ARCHITECTURE OF PROJECT

Programming language: Python

Backend : Python django framework version 4.2.1

Frontend: HTML, CSS, JavaScipt, Bootstrap

Database Management : sqlite 3(supported by django)

The architecture of a Django project follows the Model-View-Controller (MVC) architectural pattern, with some modifications specific to Django.

1. Models

The models define the structure and behavior of data in the application. They represent database tables and are defined in Python classes using Django's model fields, such as CharField, IntegerField, etc.

Models are typically defined in the models.py file within each Django app.

Django's Object-Relational Mapping (ORM) handles the communication with the database, allowing you to query and manipulate data using Python code.

2. Views

Views handle the logic of the application and control what is displayed to the user. They receive requests from the user's browser and return responses.

Views are Python functions or classes that process the incoming requests and generate appropriate responses.

In Django, views are typically defined in the views.py file within each app.

Views can interact with models to fetch or update data and pass it to templates for rendering.

3. Templates

Templates are responsible for generating the user interface and rendering dynamic content.

Django's template engine uses HTML templates with embedded Python code and variables to generate the final HTML that is sent to the user's browser.

Templates can access data passed from views and display it to the user.

Template files are typically stored in the templates directory within each app.

4. URLs

URLs define the mapping between the user's requested URL and the appropriate view function to handle it.

In Django, URL patterns are defined in the urls.py file within each app, as well as in the project-level urls.py file.

URLs can include variables or regular expressions to capture dynamic parts of the URL, which can be passed as parameters to the associated view function.

5. Settings

The settings module contains various configurations for the Django project, such as database settings, middleware settings, installed apps, static file paths, etc.

The main settings.py file at the project level defines global settings, while individual apps may have their own settings.py file for app-specific configurations.

6. Static Files

Static files include CSS stylesheets, JavaScript files, images, etc., that are served directly to the user's browser without any processing.

Django allows you to define static file directories in the settings and use them in templates or views.

7. Middleware

Middleware is a mechanism in Django that allows you to process requests and responses globally.

Middleware can perform tasks such as authentication, logging, or modifying requests/responses before they reach the view or after they leave the view.

Middleware classes are defined in the middleware.py file within each app or in the project-level middleware settings.

5.2 THIRD PARTY LIBRARIES USED

1. Pillow

Pillow is a Python imaging library that can be used with Django to handle image processing and manipulation. To use Pillow with Django, you need to install it using pip or another package manager. You also need to add 'PIL' to your INSTALLED_APPS setting in your Django project. Then you can use the ImageField or FileField models to store images in your database and the Image module to manipulate them.

Syntax : pip install pillow

2. Jazzmin

Jazzmin is a drop-in app that can jazz up your Django admin site with a modern and responsive theme based on AdminLTE 3 and Bootstrap 4. To use Jazzmin with Django, you need to install it using pip or another package manager. You also need to add 'jazzmin' to your INSTALLED_APPS before 'django.contrib.admin'. Then you can customize various aspects of your admin site, such as the side menu, the top menu, the user menu, the change form templates, the UI colors and more.

Syntax: pip install –U django-jazzmin

3. django-extensions

Django Extensions is a collection of custom extensions for the Django Framework. These include management commands, additional database fields, admin extensions and much more. To use Django Extensions with Django, you need to install it using pip or another package manager. You also need to add 'django_extensions' to your INSTALLED_APPS setting in your Django project. Then you can use various commands and features that enhance your development experience, such as shell_plus, graph_models, runserver_plus, print_settings, validate_templates and more.

Syntax: pip install django-extensions

4. django-wigget-tewaks

Django Widget Tweaks is a package that allows you to customize the form field rendering in templates, not in python-level form definitions. You can alter CSS classes and HTML attributes of form fields using template tags and filters.You also need to add 'widget_tweaks' to your INSTALLED_APPS setting in your Django project. Then you can use the render_field tag or the attr filter to tweak the form fields in your templates.

Syntax: pip install django-widget-tweaks

6. SYSTEM DESIGN

6.1 INPUT DESIGN

1. *User Registration Form*

- Fields: First name, Last name, Username, Password, Address, Mobile number, Profile picture
- Functionality: Allow customers and mechanics to create accounts with the provided details.

2. *User Login Form*

- Fields: Username, Password
- Functionality: Allow users to log in to the system using their credentials.

3. *Admin Dashboard*

- Cards: Total customers, Total mechanics, Total enquiries, Total feedback
- Recent enquiry form: Display the details of the most recent customer enquiry.

4. *Customer Dashboard*

- Cards: New requests, Mobile repairs in progress, Mobiles repaired, Total bill

5. *View Pending Requests*

- Functionality: Allow customers to view a list of their pending requests.
- Option to delete pending requests.

6. *Make a Request Form*

- Fields: Mobile category, IMEI number, Mobile name, Mobile brand, Mobile model, Problem description, Problem description image
- Functionality: Enable customers to create new service requests by providing relevant details.

7. *Approved Requests*

- Functionality: Display a list of approved requests for the customer.
- Option to view the approved request bill from the admin.

8. *Invoice*

- Functionality: Display approved requests along with their costs, as provided by the admin.

9. *Send Feedback Form*

- Fields: Feedback message
- Functionality: Allow customers to send feedback to the admin.

10. *Edit Profile Form*

- Fields: First name, Last name, Address, Mobile number, Profile picture
- Functionality: Enable customers to edit and update their profile information.

11. *Mechanic Job Application Form*

- Fields: First name, Last name, Address, Mobile number, Skills, Username, Password, Profile picture
- Functionality: Allow mechanics to apply for a job by providing relevant details.

12. *Mechanic Dashboard*

- Cards: New works assigned, Works in progress, Works completed, Current salary

13. *Update Work Status*

- Functionality: Allow mechanics to update the status of assigned works to "Approved," "Repairing," or "Repairing done."

14. *View Current Salary*

- Functionality: Display the mechanic's current salary.

15. *View Attendance*

- Functionality: Allow mechanics to view their attendance records taken by the admin.

16. *Send Feedback Form*

- Fields: Feedback message
- Functionality: Enable mechanics to send feedback to the admin.

17. Edit Profile Form

- Fields: First name, Last name, Address, Mobile number, Skills, Profile picture
- Functionality: Allow mechanics to edit and update their profile information.

These input design features provide a basis for capturing user input and facilitating interactions within the smartphone service center management system.

6.2 OUTPUT DESIGN

1. *Admin Dashboard*

- Display the count of total customers, total mechanics, total enquiries, and total feedback in separate cards.
- Show a form displaying the details of the most recent customer enquiry.

2. *Customer Dashboard*

- Display a dashboard with four cards showing the count of new requests made, mobile repairs in progress, mobiles repaired, and the total bill.

3. *View Pending Requests*

- Display a list of pending requests for the customer to view.
- Provide an option to delete pending requests.

4. *Approved Requests*

- Show a list of approved requests for the customer to view.

5. *Approved Request Bill*

- Display the approved requests along with their costs, as assigned by the admin.

6. *Invoice*

- Show the approved requests with their respective costs from the admin.

7. *Feedback Submission Form*

- Allow customers to submit feedback to the admin.

8. *Profile Update Confirmation*

- Display a confirmation message to customers upon successful profile update.

9. *Mechanic Dashboard*

- Show a dashboard with four cards displaying the count of new works assigned, works in progress, works completed, and current salary.

10. *Update Work Status*

- Display the assigned works for the mechanic and provide options to update their status as "Approved," "Repairing," or "Repairing done."

11. *View Current Salary*

- Show the mechanic's current salary.

12. *View Attendance*

- Display the attendance records taken by the admin for the mechanic.

13. *Feedback Submission Form*

- Allow mechanics to submit feedback to the admin.

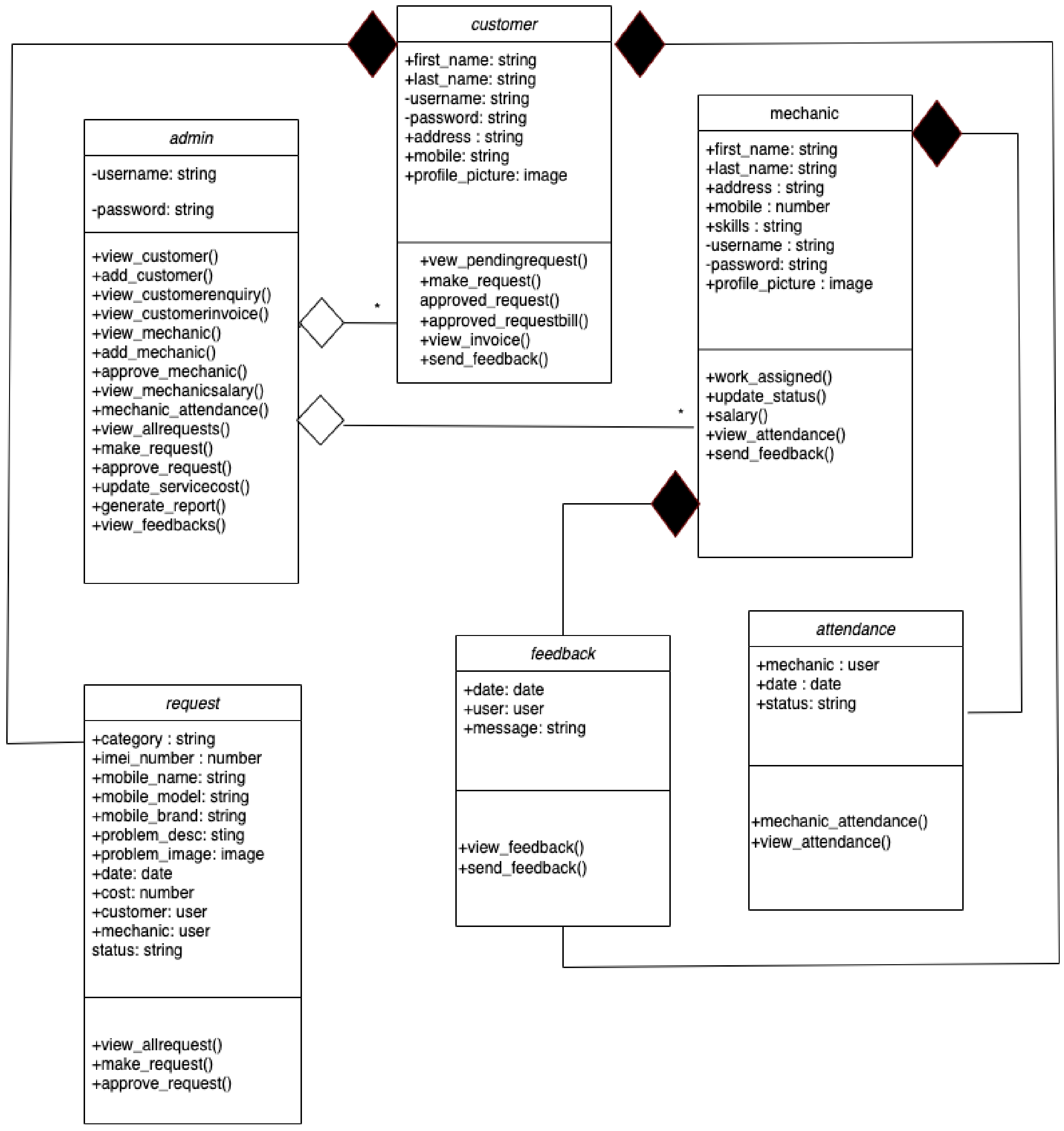
14. *Profile Update Confirmation*

- Display a confirmation message to mechanics upon successful profile update.

These output design features provide a visual representation of the information and status updates for each user type in the smartphone service center management system. They aim to present relevant data and allow users to interact with the system effectively.

6.3 DATABASE DESIGN

6.3.1 CLASS DIAGRAM



7. CHALLENGES FACED

CHALLENGES FACED

1. User Authentication and Security

Implementing a robust user authentication system with appropriate security measures can be a challenging task. You would need to handle user registration, login, password hashing, and protect sensitive user information to ensure the system is secure against potential threats like unauthorized access or data breaches.

2. Designing User Interface

Creating an intuitive and user-friendly interface is crucial for a successful smartphone sevice booking systemsystem. It involves designing visually appealing web pages, organizing information effectively, and providing a seamless user experience. Consider factors like responsive design, easy navigation, and clear call-to-action elements to enhance usability.

3. Validation Mechanisms

Implementing validation mechanisms helps ensure that user input is accurate and consistent. You need to validate user data at various stages, such as during registration, requesting, or updating information. Validations can include checking for required fields, validating email addresses, mobile number, IMEI number and enforcing password complexity.

4. Database Design and Management

Designing an efficient database schema and managing the database operations can be complex. You need to carefully plan the structure of your database, define relationships between entities (e.g., request, feedback, attendance, mechanic, and customer) handle data integrity, and optimize queries for performance.

8. FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

1. *Notification System*

- Implement a notification system to alert customers about the progress of their requests, such as when a mechanic is assigned or when the repair is completed.
- Notify mechanics about new work assignments or any updates to their assigned tasks.

2. *Service History*

- Develop a service history feature that allows customers to view their past repair requests, including details like repair date, cost, and problem description.
- Enable customers to track the repair history of their devices over time.

3. *Service Tracking*

- Introduce a service tracking mechanism that provides real-time updates to customers about the status of their repair, such as "Repair in progress" or "Device ready for pickup."
- Allow customers to track the location and progress of their device throughout the repair process.

4. *Online Payment Integration*

- Integrate an online payment system to facilitate secure and convenient payment for customers.
- Enable customers to view and pay their bills online, reducing the need for manual invoicing and payment collection.

5. *Advanced Reporting and Analytics*

- Develop comprehensive reporting features for administrators to analyze service center performance, such as customer satisfaction metrics, revenue analysis, and mechanic productivity.
- Generate graphical reports and visualizations to provide insights into key performance indicators.

6. ***Knowledge Base and FAQs***
    - Implement a knowledge base or FAQ section to provide customers with self-help resources and common troubleshooting solutions.
    - Allow customers to search for answers to frequently asked questions related to device repairs and services.
7. ***Integration with Spare Parts Inventory***
    - Integrate the system with an inventory management module to track the availability of spare parts and automate the process of ordering and restocking.
    - Enable administrators to easily manage and monitor spare parts inventory, reducing delays in repairs due to unavailable parts.
8. ***Mobile App Development***
    - Develop a mobile application for customers to access the service center functionalities on their smartphones.
    - Provide a user-friendly interface for customers to submit repair requests, track progress, and receive notifications.
    - Social Media Integration:
    - Incorporate social media integration to allow customers to share their service center experiences and provide feedback directly from the system.
    - Enable customers to log in using their social media accounts for a seamless registration and login process.
9. ***Customer Satisfaction Surveys***
    - Implement customer satisfaction surveys to gather feedback on the quality of service provided by the mechanics and the overall service center experience.
    - Analyze survey results to identify areas for improvement and address customer concerns effectively.

9. CONCLUSION

CONCLUSION

The aim of this project was to develop a Django framework project for managing the services of a smartphone service centre. The system has three types of users: admin, customer, and mechanic. Each user has different features and functionalities to access and manage the service requests, invoices, feedbacks, profiles, etc. The system also has a database to store and retrieve the data related to the users and the service requests.
The project followed the software development life cycle (SDLC) phases of planning, analysis, design, implementation, testing, and deployment. The project used various tools and technologies such as Django, Python, HTML, CSS, Bootstrap, SQLite, Git, etc. The project also followed the Agile methodology to ensure iterative and incremental development.
The project achieved its objectives of providing a user-friendly and efficient system for managing the smartphone service centre. The system allows the admin to monitor and control the service requests, mechanics, customers, feedbacks, etc. The system also allows the customers to make and track their service requests, view their invoices and feedbacks, and edit their profiles. The system also allows the mechanics to apply for jobs, view and update their work status, view their salary and attendance, and send feedbacks to the admin.
The project faced some challenges such as integrating with third-party services, ensuring data security and privacy, testing the system for different scenarios and devices, etc. The project also encountered some limitations such as lack of real-time communication between users, lack of advanced analytics and reporting features, lack of customer satisfaction surveys, online payment etc.

10. REFERENCES

https://www.w3schools.com/django/

https://www.djangoproject.com/start/

https://learndjango.com/tutorials/essential-django-3rd-party-packages

https://djangopackages.org/

https://www.stxnext.com/blog/top-django-packages-libraries/

https://www.fullstackpython.com/django-extensions-plug-ins-related-libraries.html

https://realpython.com/customize-django-admin-python/

Doorstep Mobile Repair Service at Best Price | Yaantra

Mobile Repair Online | Mobile Repairing Center In India - ShatterFix

Mobile Phone Repair: Doorstep Mobile Screen, Battery, Mic, Speaker Repair - Cashify Repair

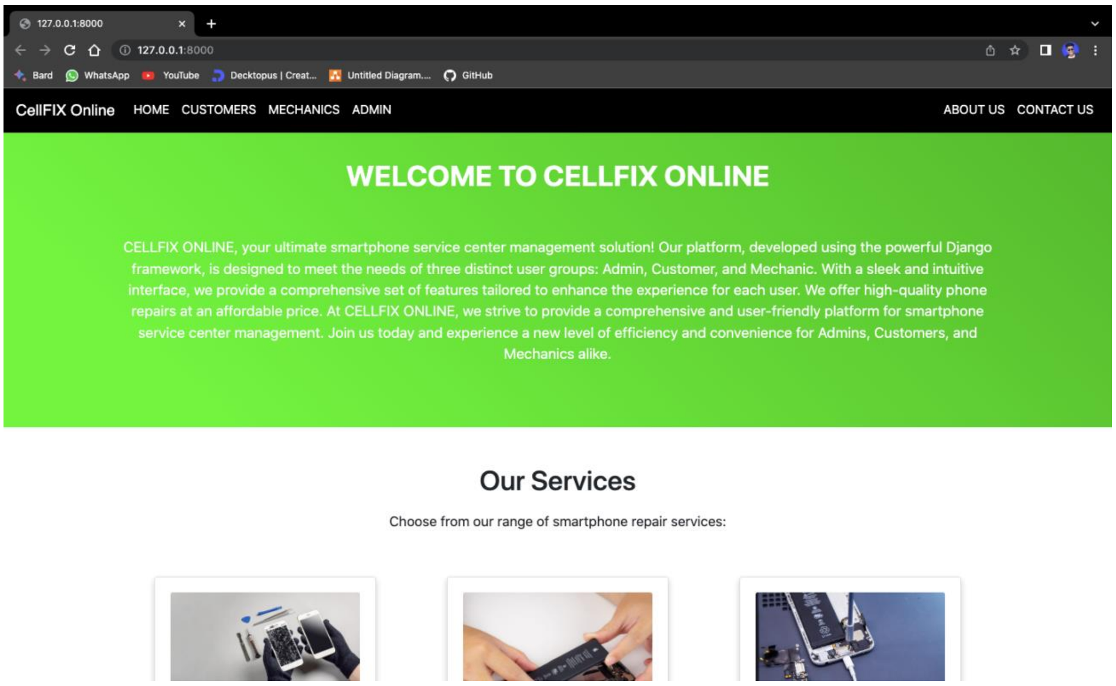
Best Free Admin Dashboard Templates | BootstrapDash

Admin Templates - Dashboard Templates | ThemeForest

11. APPENDIX

11.1 SCREEN SHOTS

Index page



CellFIX Online HOME CUSTOMERS MECHANICS ADMIN ABOUT US CONTACT US

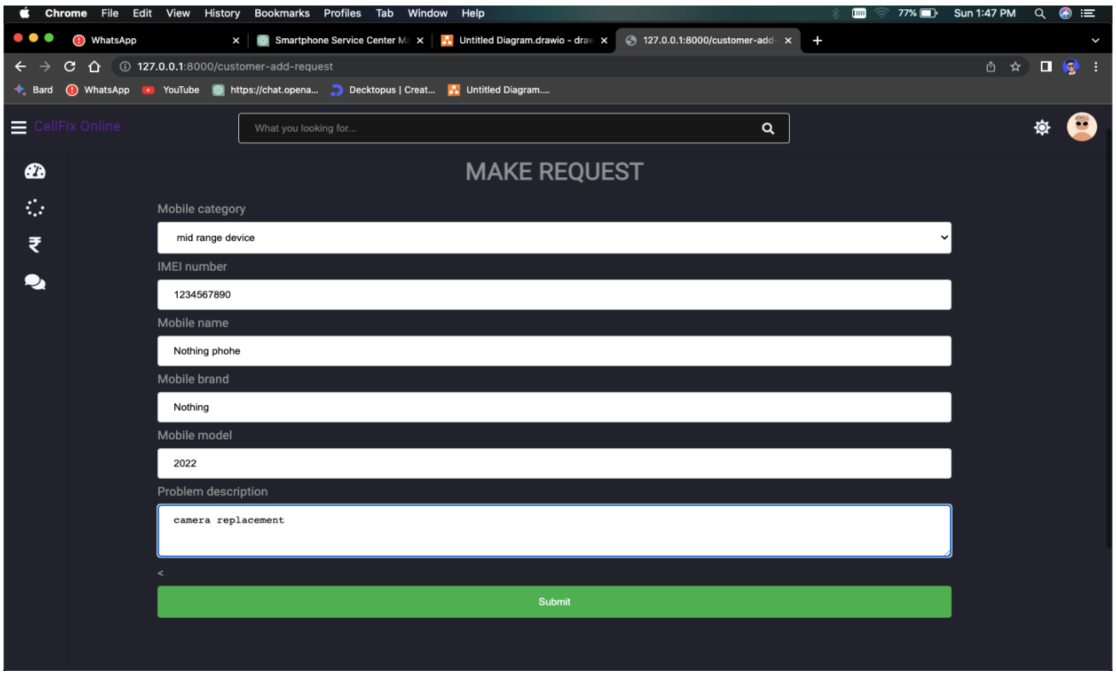
WELCOME TO CELLFIX ONLINE

CELLFIX ONLINE, your ultimate smartphone service center management solution! Our platform, developed using the powerful Django framework, is designed to meet the needs of three distinct user groups: Admin, Customer, and Mechanic. With a sleek and intuitive interface, we provide a comprehensive set of features tailored to enhance the experience for each user. We offer high-quality phone repairs at an affordable price. At CELLFIX ONLINE, we strive to provide a comprehensive and user-friendly platform for smartphone service center management. Join us today and experience a new level of efficiency and convenience for Admins, Customers, and Mechanics alike.

Our Services

Choose from our range of smartphone repair services:

Customer make request



127.0.0.1:8000/customer-add-request

MAKE REQUEST

Mobile category: mid range device

IMEI number: 1234567890

Mobile name: Nothing phohe

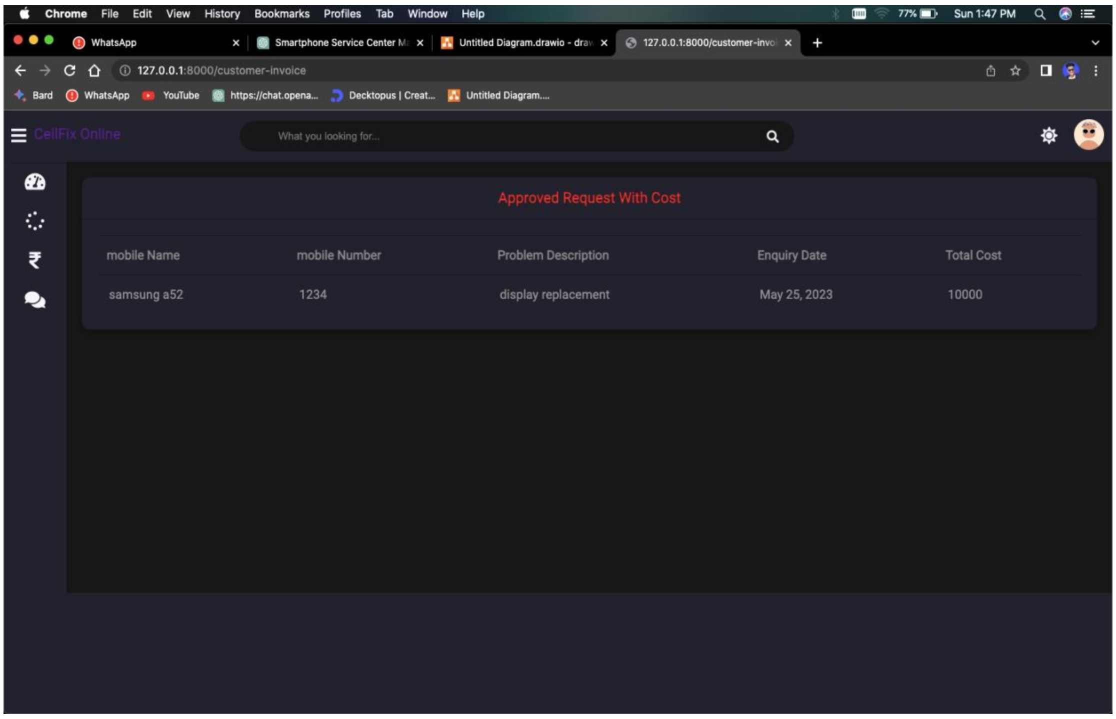
Mobile brand: Nothing

Mobile model: 2022

Problem description: camera replacement

Submit

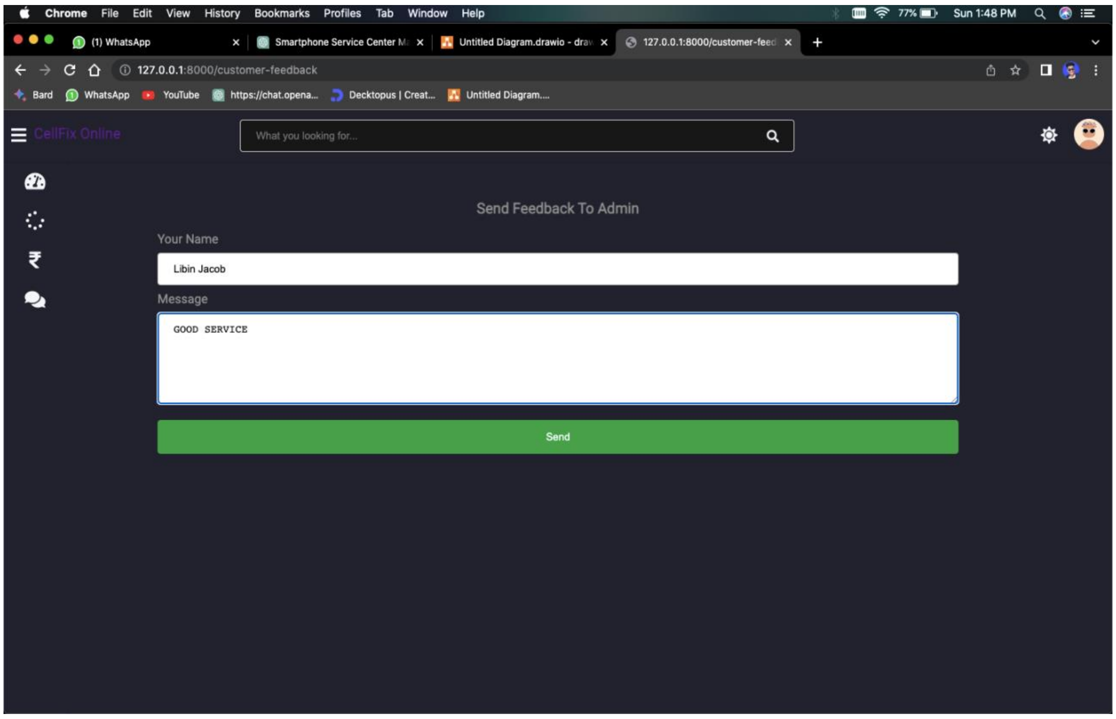
Customer approved requests with cost



Approved Request With Cost

mobile Name	mobile Number	Problem Description	Enquiry Date	Total Cost
samsung a52	1234	display replacement	May 25, 2023	10000

Customer sends feedback

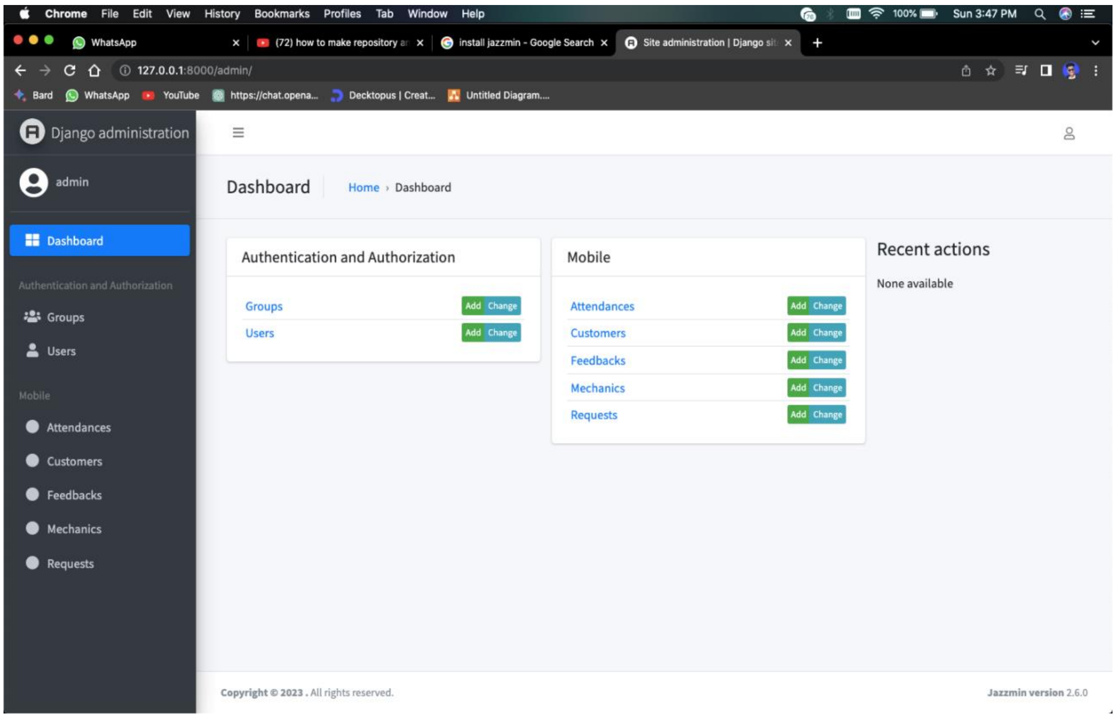


Send Feedback To Admin

Your Name: Libin Jacob

Message: GOOD SERVICE

Admin dashboard


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 3:47 PM
WhatsApp
(72) how to make repository
install jazzmin - Google Search
Site administration | Django site
127.0.0.1:8000/admin/
Bard WhatsApp YouTube https://chat.opena... Decktopus | Creat... Untitled Diagram....
Django administration
admin
Dashboard
Authentication and Authorization
Groups
Users
Mobile
Attendances
Customers
Feedbacks
Mechanics
Requests

Dashboard Home › Dashboard

Authentication and Authorization		
Groups	Add	Change
Users	Add	Change

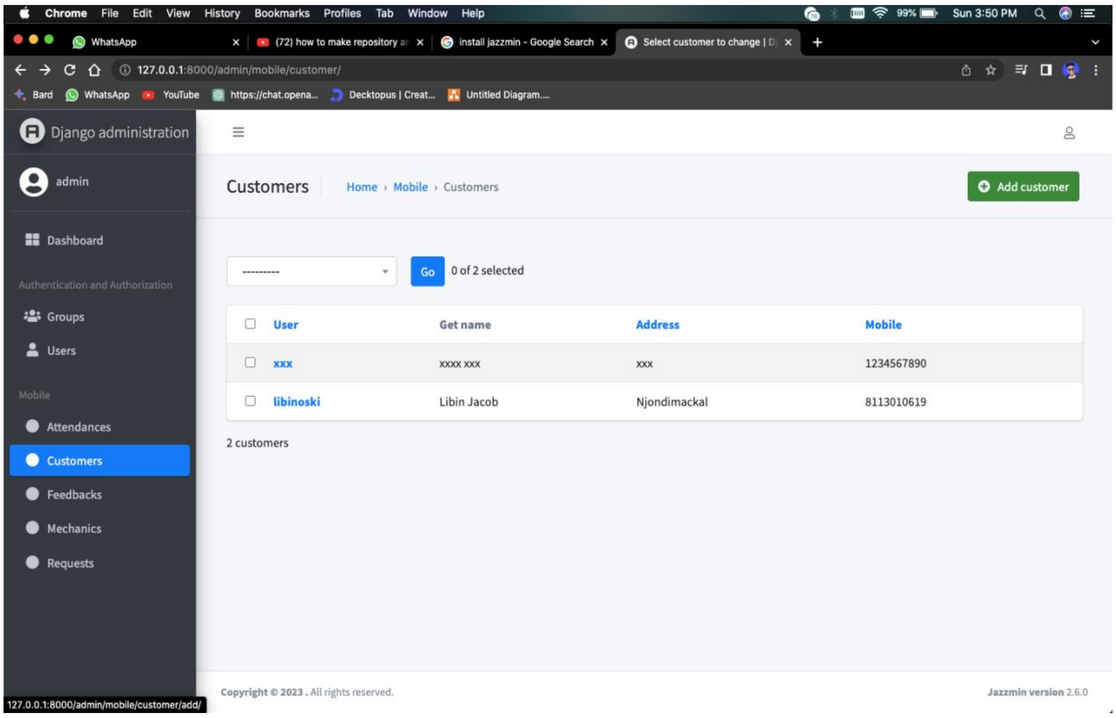
Mobile		
Attendances	Add	Change
Customers	Add	Change
Feedbacks	Add	Change
Mechanics	Add	Change
Requests	Add	Change

Recent actions

None available

Copyright © 2023 . All rights reserved. Jazzmin version 2.6.0

Admin manages customer


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 3:50 PM
WhatsApp
(72) how to make repository
install jazzmin - Google Search
Select customer to change | D
127.0.0.1:8000/admin/mobile/customer/
Bard WhatsApp YouTube https://chat.opena... Decktopus | Creat... Untitled Diagram....
Django administration
admin
Dashboard
Authentication and Authorization
Groups
Users
Mobile
Attendances
Customers
Feedbacks
Mechanics
Requests
Add customer
Go

Customers Home › Mobile › Customers

--------- Go 0 of 2 selected

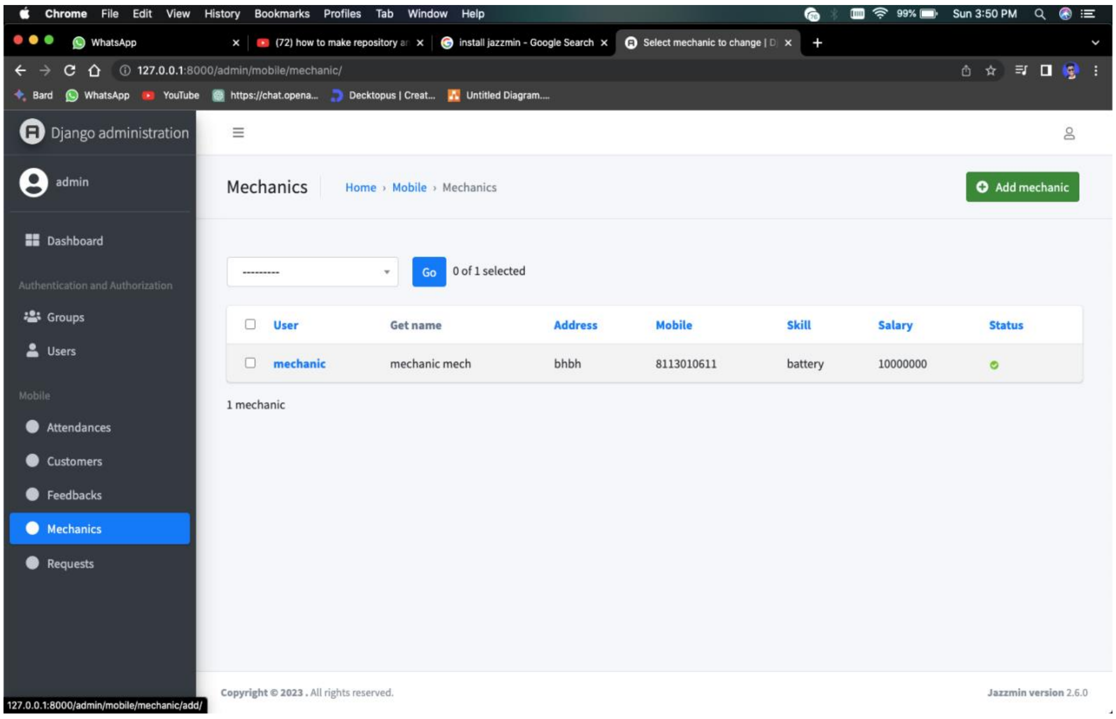
User	Get name	Address	Mobile
xxx	xxxx xxx	xxx	1234567890
libinoski	Libin Jacob	Njondimackal	8113010619

2 customers

Copyright © 2023 . All rights reserved. Jazzmin version 2.6.0

127.0.0.1:8000/admin/mobile/customer/add/

Admin manages mechanic



Django administration

admin

Dashboard

Authentication and Authorization

Groups

Users

Mobile

Attendances

Customers

Feedbacks

Mechanics

Requests

Mechanics   Home › Mobile › Mechanics   Add mechanic

--------- Go 0 of 1 selected

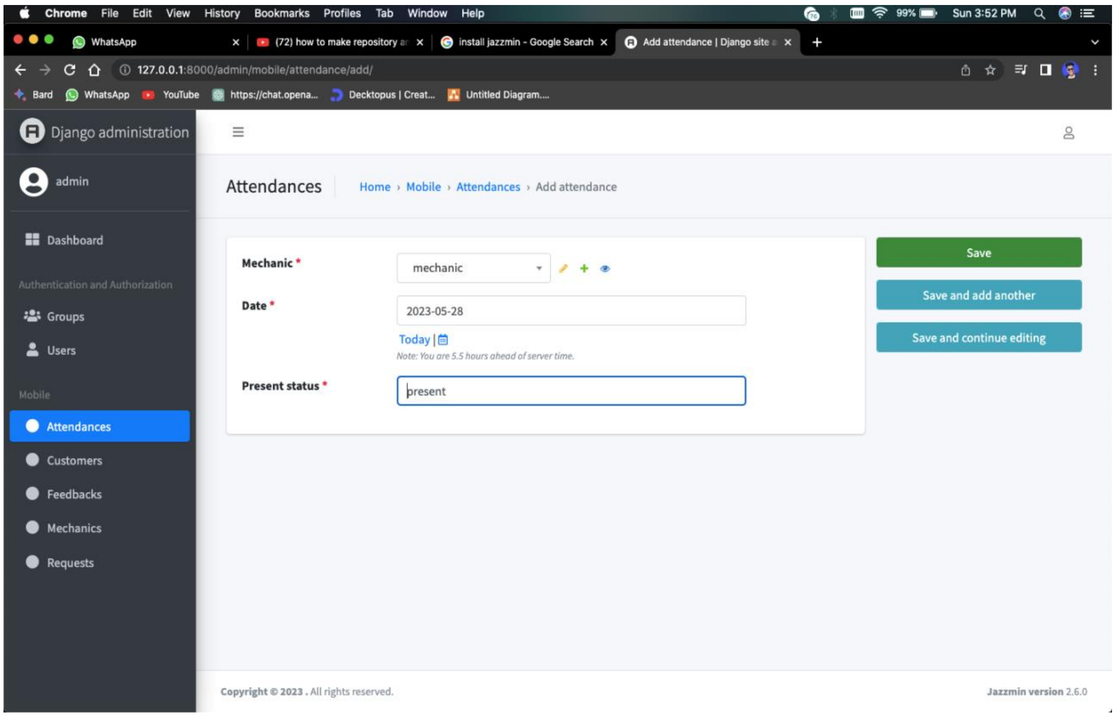
	User	Get name	Address	Mobile	Skill	Salary	Status
☐	mechanic	mechanic mech	bhbh	8113010611	battery	10000000	✓

1 mechanic

Copyright © 2023 . All rights reserved.   Jazzmin version 2.6.0

127.0.0.1:8000/admin/mobile/mechanic/add/

Admin take attendance



Django administration

admin

Dashboard

Authentication and Authorization

Groups

Users

Mobile

Attendances

Customers

Feedbacks

Mechanics

Requests

Attendances   Home › Mobile › Attendances › Add attendance

Mechanic *	mechanic
Date *	2023-05-28
	Today |
	Note: You are 5.5 hours ahead of server time.
Present status *	present

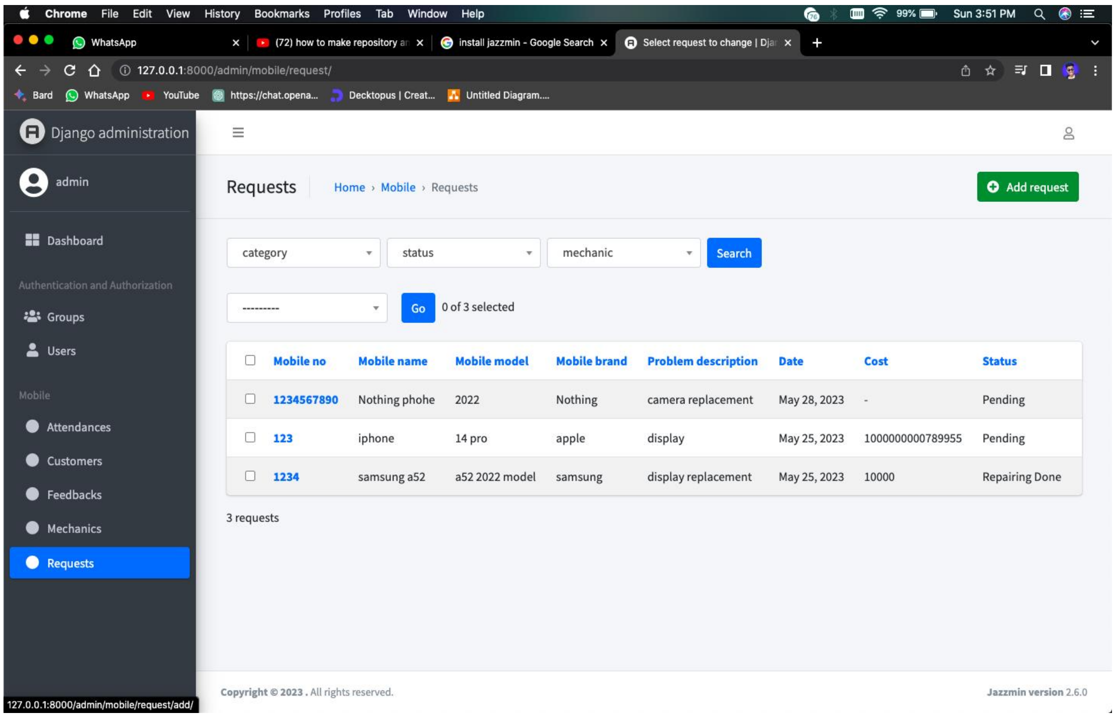
Save

Save and add another

Save and continue editing

Copyright © 2023 . All rights reserved.   Jazzmin version 2.6.0

Admin manages requests



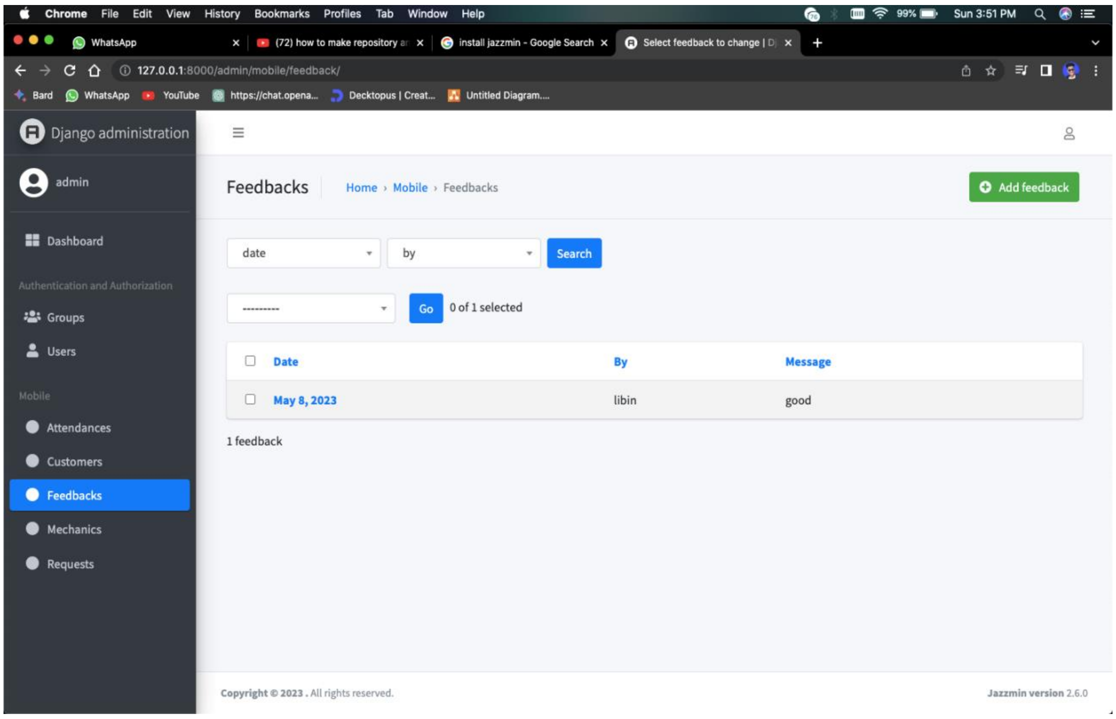
Requests

Home › Mobile › Requests

	Mobile no	Mobile name	Mobile model	Mobile brand	Problem description	Date	Cost	Status
☐	1234567890	Nothing phohe	2022	Nothing	camera replacement	May 28, 2023	-	Pending
☐	123	iphone	14 pro	apple	display	May 25, 2023	1000000000789955	Pending
☐	1234	samsung a52	a52 2022 model	samsung	display replacement	May 25, 2023	10000	Repairing Done

3 requests

Admin manages feedbacks



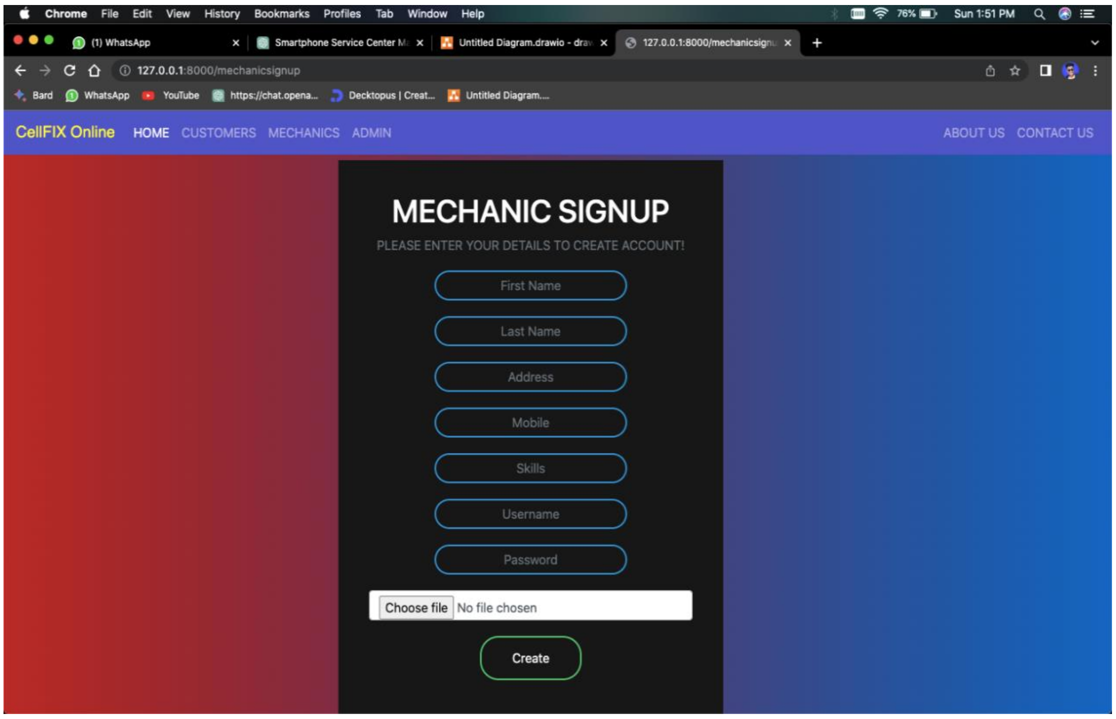
Feedbacks

Home › Mobile › Feedbacks

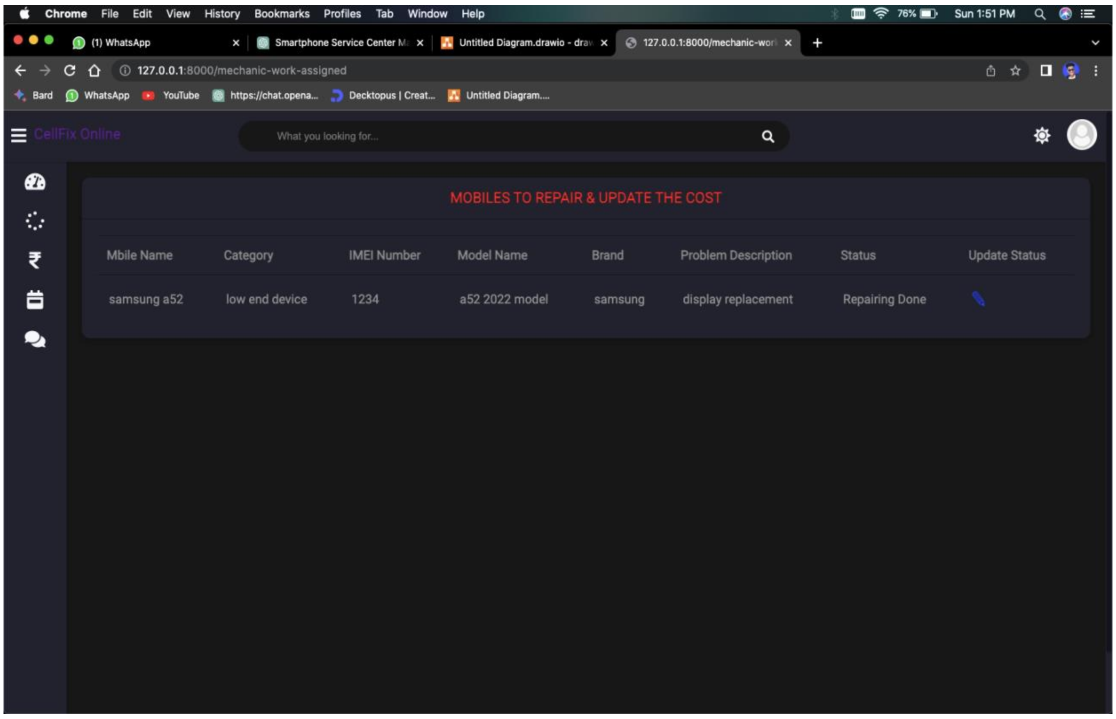
	Date	By	Message
☐	May 8, 2023	libin	good

1 feedback

Mechanic apply for job



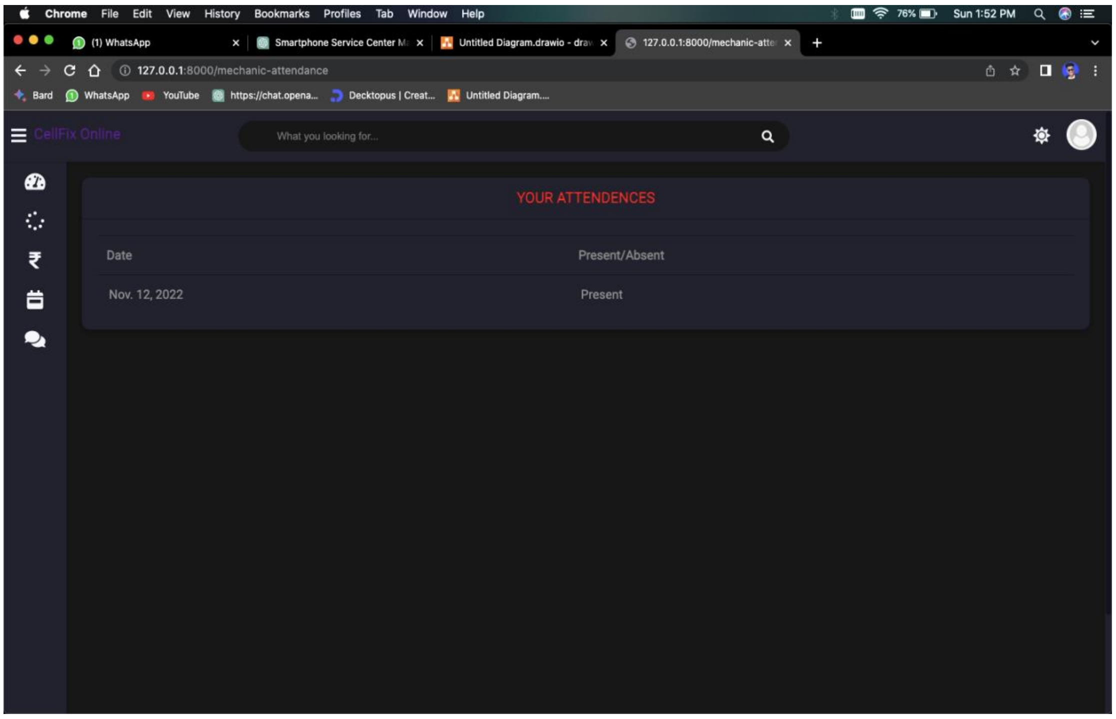
Mechanic manages works assigned



MOBILES TO REPAIR & UPDATE THE COST

Mbile Name	Category	IMEI Number	Model Name	Brand	Problem Description	Status	Update Status
samsung a52	low end device	1234	a52 2022 model	samsung	display replacement	Repairing Done	

Mechanic views attendance



YOUR ATTENDENCES

Date	Present/Absent
Nov. 12, 2022	Present