MUNICIPAL CORPORATION

A REAL TIME RESEARCH PROJECT REPORT

Submitted by

K.NANDU

(22841A0592)

K.JAGAN GOUD

(22841A05C5)

B.NAMRATHA

(23845A0507)

B.MATHSYAGIRI

(23845A0508)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

AURORA'S TECHNOLOGICAL AND RESEARCH INSTITUTE

(Affiliated to JNTU, Hyderabad and approved by AICTE, NewDelhi)

Parvathapur, Uppal, Hyderabad-500039

JULY 2024

AURORA’S TECHNOLOGICAL AND RESEARCH INSTITUTE

(Affiliated to JNTU, Hyderabad and approved by AICTE, NewDelhi)

Parvathapur, Uppal, Hyderabad-500039

DECLARATION

We hereby declare that the work described in this project, entitled **“MUNICIPAL CORPORATION”** which is being submitted by us in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** to **AURORA’S TECHNOLOGICAL AND RESEARCH INSTITUTE** is the result of investigation carried by us.

The work is original and has not been submitted for any degree of this or any other university.

Place:Hyderabad

Date:

K.NANDU(22841A0592)

K.JAGAN GOUD*(22841A05C5)*

B.NAMRATHA*(23845A0507)*

B.MATHSYAGIRI*(23845A0508)*

AURORA’S TECHNOLOGICAL AND RESEARCH INSTITUTE

(Affiliated to JNTU, Hyderabad and approved by AICTE, NewDelhi)

Parvathapur, Uppal, Hyderabad-500039

BONAFIDECERTIFICATE

Certified that this project report “BREAST CANCER DIAGNOSIS USING ADAPTIVE VOTING ENSEMBLE MACHINE LEARNING ALGORITHM” is the bonafide work of “G.PRIYANKA (22841A0505), S.V.VAISHNAVI (22841A0505), G.PRIYANKA (22841A0505), S.V.VAISHNAVI (22841A0505)” who carried out the project work under our supervision.

PROJECTCOORDINATOR	**HEAD OF THE DEPARTMENT**
Mr. CH. Krishna Rao	**S.Mahesh**
Assoc.Prof	Assoc.Prof
Dept. of CSE	Dept. of CSE

PRINCIPAL

ACKNOWLEDGMENT

This work has been done during the project period and it was a very good opportunity to put theoretical knowledge into planned exercise with an aim to solve a real time problem and also to develop confidence to face various practical situations.

We would also like to express our gratitude to Mr. Srikanth Jatla, Director, Aurora's Technological and Research Institute for providing us with a congenial atmosphere and encouragement.

We express our sincere thanks to Head of the Department Mr. S.Mahesh for giving us the support and her kind attention and valuable guidance to us throughout this course.

We express our sincere thanks to Project Coordinator Mr. CH. Krishna Rao for helping us to complete our project work by giving valuable suggestions and for providing encouragement, constant support and guidance which was of great help to complete this project successfully.

MUNICIPAL CORPORATION

ABSTRACT

This project is mainly based on the domain (Web Application Development). The common people under the jurisdication of municipal corporation have to deliver their grievances about day to day problem in their ward. In India we don’t have any direct communication between the government and public in an efficient way for solving the problems. A web application is proposed to overcome their problem by delivering the grievances to the government. It will provide a common man to deliver his complaints and problems to municipal authority as well as let the municipal authorities to address the problem in a short period of time. It act as interface to register one’s complained and follow it up and also it provides a complaint module which helps clicking up a picture of any problem that people are facing and upload its image, text information and location along with the complaint. In addition to that it also includes the online discussion forums and feedback forms which will help them to communicate well with the government and then how effectively the funds are utilized for the development purpose can be known by Public.

CHAPTER-1

INTRODUCTION

The main purpose of the project is to help the public who are facing different problems in the localities by this online application. This project is having that potential to reduce the gap between people and Govt. It can control unethical work of bribe and even it can reduce the processing time. In this project identification and solution for the complaints given by the people, rectifying them is the main concept of the project. A clear report is generated by the system which shows name, complaint type, etc. All the above attributes help while viewing the report of complaints. The admin examines weather the problem is rectified or not within the grace period. The main objective of this project is to make easy the process of complaint reporting with very simplified and effective way. This project involves major problem solving modules where these acts as best solution for incoming bulk complaints. For every submission of complaint, the user gets complaint acknowledgement. All these type of acknowledgement is generated by the computer; the solution of time may differ from the type of the complaint and category. To make any complaint, it is made mandatory for the user to mention his contact details, so that it does not receive any anonymous complaint details.

CHAPTER-2

LITERATURE SURVEY

Title: Online Complaint Management System Using Image Recognition

AUTHOR: S.Balakrishnan, T.Rohith, J.Janep

ABSTRACT: Clean and safe environments are the responsibility of the government. Currently, people have to lodge their complaints with the Municipal Corporation, and then workers take time to respond to them. In this paper, we have made a platform to overcome the communication gap between the user and the municipal corporation. This research study proposes an online complaint management system in a web application, where the citizen can complain by taking a picture. After the picture is uploaded, image recognition is used to automatically detect the problem raised by the user, and Google Map API to detect the location automatically.

TITLE: Natural language mobile interface to register citizen complaints

AUTHOR: Sunil Kumar kopparapu

ABSTRACT: Municipal corporation (MC) of a city is a local governing body which takes care of the functioning of the city. Among many other things one of the chief responsibilities of an MC includes addressing the complaints that the residents of the city might have. Maintenance of a large city requires that the MC be aware of any shortcomings either through surveillance (sensors/cameras) or by allowing the citizens to report them. The second option is usually preferred because there is a sense of belonging. A mechanism to accept complaints from citizens 24 times 7 would be the expectation from both the citizens and the MC.

TITLE: Utility Point: An User Service Assistance Based Mobile Application For the Citizens of Bangladeshi City Corporation

AUTHOR: Alve Ahmad, Arup Saha

ABSTRACT: Abstract:City-corporation is one of the important institutions that can improve the city people lifestyles by providing necessary utility services along with socioeconomic development. Currently, most of the services offered by the city corporation are handled offline mode. To keep pace with the modern world, the incorporation of information and communication technology (ICT) based digital technology is necessary. At present, the existing works related to city corporation service do not consider multiple utility features, problem reporting, and fee payment features at the same time. Thus, the existing city corporation-related service delivery works may experience huge time and cost wastage. To cope up with the existing city corporation-related issues, in this paper we have presented a user service assistance mobile application for the citizens of the Bangladeshi city corporation. Our proposed utility point mobile application offers several features for the citizens like electricity, water, gas bill payment, trash service payment, cemetery plot booking service, holding tax payment, e-trade and vehicle license apply, birth and citizen certificate, problem reporting, among others. Our user survey results indicate that almost 80% of users are satisfied with the features of the proposed city corporation mobile application.

Title: Waste Management Improvement in Cities using loT

AUTHOR: Shivam Jagtap, Adithya Gandhi, Raviraj Bochare

Abstract: Garbage collection is one of the most critical problems faced by Municipal Corporation. While implementing the waste management in cities the biggest challenge is the management of waste in cost optimal way with high performance. The current process of collecting the waste, separating it and transporting the containers everyday which is a complicated process. This paper deals with the concept of waste management and the smart system for waste management with higher benefits to the society. The proposed system for waste management will use various sensors for sensing the type of waste and separate the waste in different categories and actuator to inform the management to collect the waste container. This system will save money and time compared to the already available process of waste management and also improves the society

Title: Sustainable Waste Management Model

AUTHOR: Indu Chawla, Shika Jain, Rishabh Jain

Abstract: The problems concerning waste management in the world are rapidly increasing day by day. Improper planning of waste management and lack of technical support are the main reasons of these problems. This greatly affects the health conditions of the citizens. Consequently, it is vital to propose a feasible solution for this problem. This paper represents a sustainable waste management system which can be adopted by any city. It can be served as a guiding tool to the municipal corporation. The system ensures a sequential execution of the procedures following collection, disposal, transportation and processing of waste. The idea is to divide the management system into four modules-division of the city into zones to reduce the problem size, assignment of scrape yards to each zone, optimization of waste collection route and setting an efficient budget for investing in waste processing machines, thereby designing an overall waste management model. This management model helps both ways, environmentally and economically. The proposed model gives precise results for the dataset.

CHAPTER-3

SYSTEM ANALYSIS

EXISTING SYSTEM

The existing municipal corporation system often faces challenges such as bureaucratic inefficiencies, corruption, and insufficient infrastructure development. Disadvantages may include inadequate waste management, slow decision-making processes, and disparities in service delivery across different areas within the municipality.

A corporation is a distinct legal entity. The business is governed by a board of directors. A different structure may suit you better if you want to maintain full control of your business.

DISADVANTAGES:

1. Bureaucratic Inefficiencies:

Municipal corporations often suffer from bureaucratic complexities, leading to slow decision-making processes and hindering the efficient execution of projects and services. The layers of approval and hierarchical structures can impede agility in responding to the dynamic needs of the community.

2. Corruption and Lack of Transparency:

Large budgets and significant control over resources make municipal corporations susceptible to corruption. Maintaining transparency and accountability can be challenging, and instances of bribery, embezzlement, or favoritism can erode public trust, impacting the effective functioning of the corporation.

3. Slow Response to Community Needs:

The inherent bureaucratic nature of municipal corporations may result in a delayed response to the evolving needs of the community. This lag in responsiveness can lead to dissatisfaction among citizens and a perception of a disconnect between the local government and the residents.

4. Budget Constraints:

Limited financial resources pose a continual challenge for municipal corporations. Balancing the budget while meeting the diverse needs of the community requires careful financial management.

Budget constraints can lead to inadequate service delivery, restricted infrastructure development, or unpopular outcomes such as increased taxation.

5. Difficulty in Adapting to Rapid Urban Changes:

Municipalities often struggle to adapt swiftly to the rapidly changing urban environment. Technological advancements, demographic shifts, and environmental considerations contribute to challenges in urban planning and infrastructure development, leaving municipal corporations grappling to keep pace

PROPOSED SYSTEM:

Designing a municipal corporation system involves key elements like governance structure, service provision, and community engagement. Could you specify which aspects you're interested in, such as organizational structure, civic services, or citizen participation

- Municipal corporations offer advantages such as efficient local governance, focused urban development, improved civic services, and better infrastructure management. They empower local communities to address specific needs and foster local participation in decision-making.

ADVANTAGES:

1. Bureaucratic Inefficiencies:

Municipal corporations often suffer from bureaucratic complexities, leading to slow decision-making processes and hindering the efficient execution of projects and services. The layers of approval and hierarchical structures can impede agility in responding to the dynamic needs of the community.

2. Corruption and Lack of Transparency:

Large budgets and significant control over resources make municipal corporations susceptible to corruption. Maintaining transparency and accountability can be challenging, and instances of bribery, embezzlement, or favoritism can erode public trust, impacting the effective functioning of the corporation.

3. Slow Response to Community Needs:

The inherent bureaucratic nature of municipal corporations may result in a delayed response to the evolving needs of the community.

CHAPTER-4

SYSTEM REQUIREMENTS:

HARDWARE INTERFACES

System	**:**	i5
Hard Disk	**:**	40GB.
Ram	**:**	4GB.

4.2.2 SOFTWARE INTERFACES

Technology	**:**	Java 2 Standard Edition, JDBC
Web Server	**:**	Tomcat 7.0
Client-Side Technologies	**:**	HTML, CSS, JavaScript
Server-Side Technologies	**:**	Servlets, JSP
Data Base Server	**:**	MySQL
Editor	**:**	Netbeans8.1

CHAPTER-5

SYSTEM STUDY

5.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates, During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

5.2 FEASIBILITY ANALYSIS

Three key considerations involved in the feasibility analysis are

ECONOMICAL FEASIBILITY

TECHNICAL FEASIBILITY

SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER-6

SYSTEM ARCHITECTURE

6.2 UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

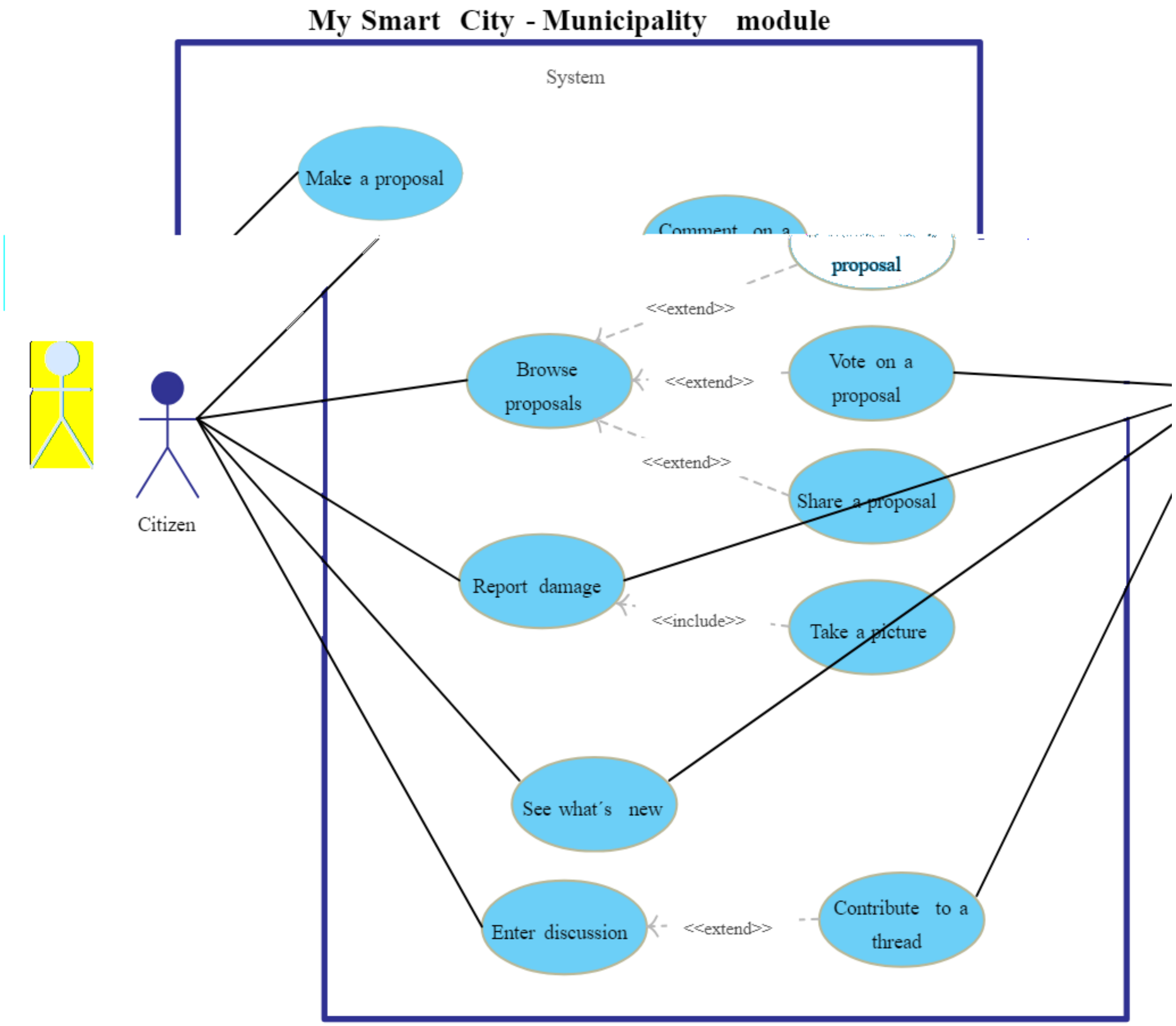
5. Encourage the growth of the 00 tools market.

6. Support higher-level development concepts such as collaborations, frameworks, patterns, and components.

7. Integrate best practices.

6.2.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



6.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. Class diagrams are a fundamental tool in object-oriented modeling and are widely used in software development.

6.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

6.2.4 COLLABARATION DIAGRAM:

A Collabaration diagram, also known as a Communication diagram, is a type of Unified Modeling Language (UML) diagram used to visually represent the interactions and communication between objects or components in a system or software application. It focuses on showing how various objects or components collaborate to achieve specific functionality or a particular use case. Collaboration diagrams are particularly useful for understanding the dynamic aspects of a system and how objects interact over time.

6.2.5 ACTIVITY DIAGRAM:

An Activity diagram is a type of Unified Modeling Language (UML) diagram that is used to model the dynamic aspects of a system, particularly the flow of activities and actions within a business process, use case, or other system behaviors. Activity diagrams provide a graphicalrepresentation of how different activities and actions are organized and interact to achieve a specific goal.Activity diagrams help stakeholders, including analysts, designers, and developers, to understand the sequential and parallel flows of activities and make them a valuable tool for system design and documentation.

CHAPTER-7

INPUT AND OUTPUT DESIGN

7.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people

keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

➤ What data should be given as input?

➤ How the data should be arranged or coded?

➤ The dialog to guide the operating personnel in providing input.

➤ Methods for preparing input validations and steps to follow when error occur.

7.1.10BJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from thecomputerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the datamanipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the userwill not be in maize of instant. Thus the objective of input design is to create.

7.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision- making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

◆Convey information about past activities, current status or projections of the ◆Future.

Signal important events, opportunities, problems, or warnings.

◆Trigger an action.

CHAPTER-8

IMPLEMENTATION

MODULES ON MUNICILPAL CORPORATION

Administrative Module: This module deals with the administrative functions of the municipal corporation, including personnel management, budgeting, financial management, and legal affairs.

Urban Planning and Development Module: This module focuses on urban planning, land use management, zoning regulations, and infrastructure development within the city.

Public Works Module: This module covers the maintenance and construction of public infrastructure such as roads, bridges, parks, and sanitation systems.

Revenue and Taxation Module: This module involves revenue generation through property taxes, licensing fees, permits, and other sources of municipal income.

Health and Sanitation Module: This module deals with public health initiatives, sanitation services, waste management, and pollution control measures.

Social Services Module: This module encompasses social welfare programs, public housing initiatives, community development projects, and initiatives to address homelessness and poverty.

Environmental Management Module: This module focuses on environmental protection, conservation efforts, and sustainable development practices within the city.

Legal and Regulatory Compliance Module: This module ensures that the municipal corporation complies with relevant laws, regulations, and policies at the local, state, and national levels.

These modules help streamline the operations of municipal corporations and ensure efficient delivery of services to city residents. They may vary depending on the size, needs, and priorities of each municipality.

CHAPTER-9

SOFTWARE ENVIRONMENT

9.1 Java Technology

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

Simple

Architecture neutral

Object oriented

Portable

Distributed

High performance

Interpreted

Multithreaded

Robust

Dynamic

Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

The Java Virtual Machine (Java VM)

The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

The essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions used by applets.

Networking: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

Internationalization: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

Security: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

Software components: Known as JavaBeansTM, can plug into existing component architectures.

Object serialization: Allows lightweight persistence and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBCTM): Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

How Will Java Technology Change My Life?

We can’t promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

Get started quickly: Although the Java programming language is a powerful object-oriented language, it’s easy to learn, especially for programmers already familiar with C or C++.

Write less code: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

Write better code: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people’s tested code and introduce fewer bugs.

Develop programs more quickly: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

Avoid platform dependencies with 100% Pure Java: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure JavaTM Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

Write once, run anywhere: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

Distribute software more easily: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn’t as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database

To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. ***SQL Level API***

   The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

2. ***SQL Conformance***

   SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. ***JDBC must be implemental on top of common database interfaces***

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. ***Provide a Java interface that is consistent with the rest of the Java system***

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. ***Keep it simple***

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. ***Use strong, static typing wherever possible***

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

7. ***Keep the common cases simple***

Because more often than not, the usual SQL calls used by the programmer are simple SELECT’s, INSERT’s, DELETE’s and UPDATE’s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

Java ha two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple

Object-oriented

Distributed

Interpreted

Robust

Architecture-neutral

Portable

High-performance

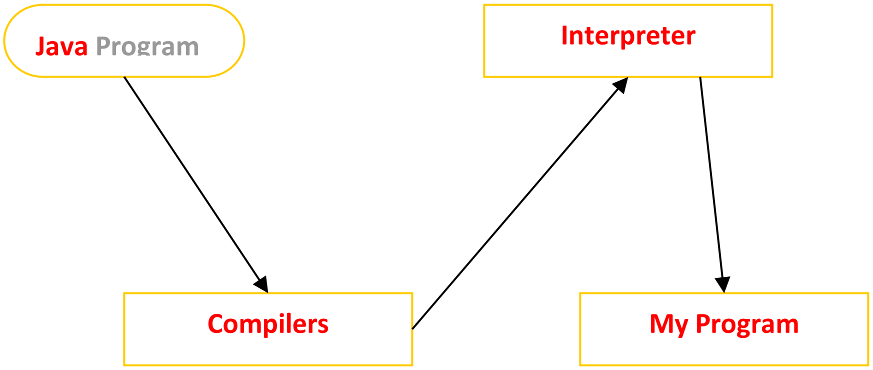
multithreaded

Dynamic

Secure

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



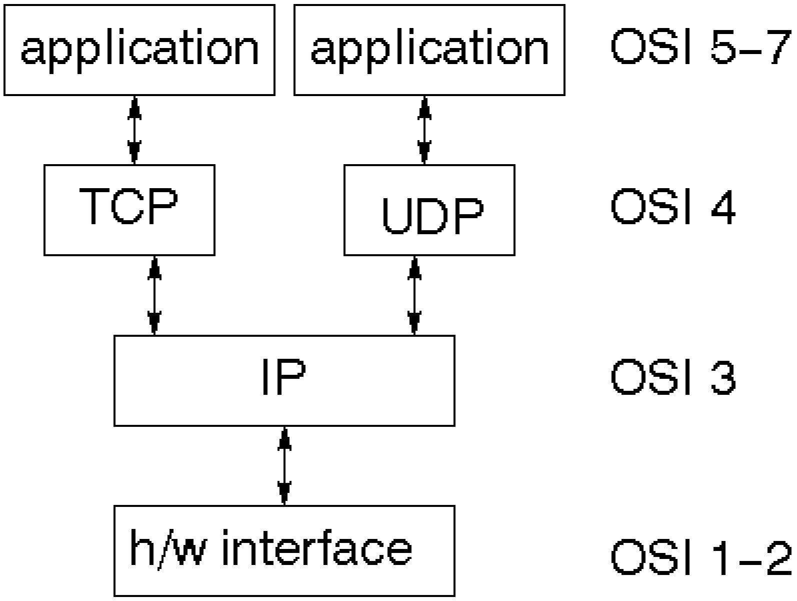
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

Networking

TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

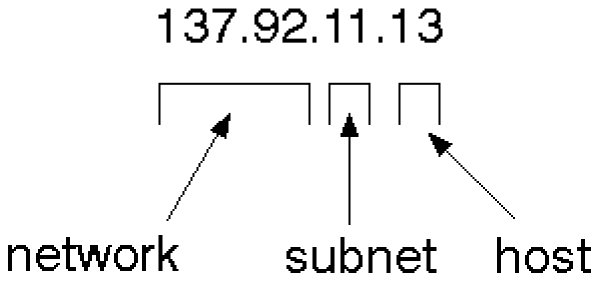
Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address



The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with `Read File` and `Write File` functions.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int family, int type, int protocol);
```

Here "family" will be `AF_INET` for IP communications, `protocol` will be zero, and `type` will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

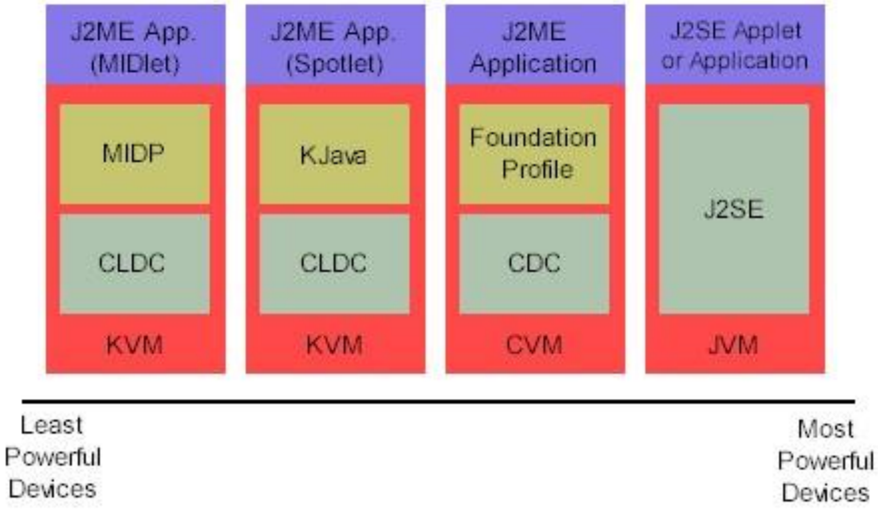
4. Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME architecture



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2.Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

3.Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

* Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.

* Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.

* Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

4.Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

* **Connected Limited Device Configuration (CLDC)** is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

* **Connected Device Configuration (CDC)** is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5.J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is com.sun.kjava. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application

development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

* java.lang

* java.io

* java.util

* javax.microedition.io

* javax.microedition.lcdui

* javax.microedition.midlet

* javax.microedition.rms

SOURCE CODE:

MUNCIPAL HOMEPAGE:

```
<html>

    <head>

        <title>MUNCIPAL CORPORATION</title>

        <style>


           .menu ul li{


                float: center;

                font-size: 20px;

                width: 100%;

                display: inline-block;


                margin: 0px 50px;

                padding-top:  20px;
```

```
}
.menu ul  {


    display: inline-block;
    height: 50px;
    width: 100%;







}


a{
    text-decoration: none;
    color: blue;
    display: inline-block;
```

```
}

a:hover{

    background-color: red;

}

#heading{

    background-image: url(d);

    background-repeat: no-repeat;

    background-size: cover;

    margin: 0px;

    height: 620px;

    width: 100%;

}

h1{

    margin-top: : 45px;



}



#menu{

    display: flex;

    height: 100px;

    width: 100%;

    background-color: lightblue;
```

```
        }


    </style>

</head>

<body>


        <div id="menu">

           <image src="t" >

            <center>


                <h1 style=" color: red;  ">

                    MUNCIPAL CORPORATION

                </h1>

                </center>




        <ul>

            <li><a href="muncipal.html">Home</a></li>

            <li><a href="admin.html">Admin</a></li>

            <li><a href="user.html">user</a></li>
            <li><a href="departement.html">Departement</a></li>
          </ul>


          </div>
          <div id="heading">


            </div>


      </body>
  </html>
```

Userpage:

```
<html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>





        #body{
            background-image: url(l);
            background-repeat: no-repeat;
            background-size: cover;
            height: 100vh;
            width: 100%;
         }
        h1{
```

```
        padding-top: 50px;
    }jj
    ::placeholder{
        padding-left: 30px;
        color: black;
    }
    option{
        width: 200px;
        height: 30px;
        margin-left: 30px;
    }
    a:hover{

        background-color: greenyellow;
        width:100% ;

    }


    </style>
</head>
<body>

    <div id="body">
        <center>
            <div id="box">
                <h1 style="color: white;">User Login</h1>
                <form action="useraction.jsp">
                <table>
                    <tr>
                        <th style="color: white; ">username:</th>
```

```
                            <td><input type="password" name="username" placeholder="username"
value="username"></td>
                        </tr>
                        <tr>
                            <th style="color: white;">password:</th>
                            <td><input type="password" name="password" placeholder="password"
value="password"></td>

                        </tr>
                        <tr>
                            <td style="color: white;"><a  ><button type="submit">sign up</button></a></td>
                        </tr>
                    </table>
                    </form>



                </div>
            </center>
        </div>
    </body>
</html>
```

USERACTION.JSP;

```
<%--
    Document   : adminlogaction
    Created on : Jun 28, 2024, 1:53:59 PM
    Author     : hp
--%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>
```

```
<%@page import="java.sql.Connection"%>

<%@page import="java.sql.DriverManager"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>


<%

    String username=request.getParameter("username");


 String password=request.getParameter("password");

try{

     Class.forName("com.mysql.jdbc.Driver");

    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/muncipal","root","root");

    Statement st=con.createStatement();

    ResultSet rs=st.executeQuery("select * from table2 where username='"+username+"' and
'"+password+"'");

             if(rs.next()){

                 session.setAttribute("username",username);

                 %>

                 <script>

                     window.alert("logged in successfully.......");

                     window.location="userhome.jsp"

                 </script>

                 <%

             }

else{

%>

<script>

                     window.alert("logged failed.......");

                     window.location="user.html"

                 </script>

                 <%
}


}
catch(Exception e){
    out.println(e);
}
%>
```

USERHOME.JSP:

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>



    <html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>
            #header{
                display: flex;
                height: 200px;background-color: wheat;
                width: 100%;
                margin: 0px;


            }
            #box{
                display: flex;
                height: 120px;
                background-color: lightgrey;
                width: 100%;
                margin: 1px;

            }
            .image{
                width: 100%;
                height: 120px;
            }
            b{
                color: red;

            }
            #box1{
                display: flex;
                height: 700px;
                background-image: url(OIP.jpeg);
                background-repeat: no-repeat;
                background-size: cover;
            }
            label{
                margin-left: 450px;
                color: blue;
            }
            p{
                color:  white;
            }
             #boy{
                            height: 300px;
                              width: 500px;
                              border:1px solid white;
                              background: lightsteelbluewhitesmoke;
                             margin-top: 300px;
                             color: white;

              }
              table{
                  color: white;
              }
               .header ul  {

                 display: inline-block;
                 height: 50px;
                 width: 100%;
               }
                .header ul li{

                 float: center;
                 font-size: 20px;
                 width: 100%;
                 display: inline-block;

                 margin: 0px 50px;
                 padding-top:  20px;




            a{
                 text-decoration: none;
                 color: blue;
```

```
            display: inline-block;





        }
                         a:hover{
            background-color: red;
        }
                        p{
                            align-items: center;
                            width: 100%;
                            padding-top: 10px;
                            padding-left: 150px;

                            size: 50px;
                        }
                        b1{
                            color: white;
                        }





    </style>
</head>
<body>

    <div id ="header" > <center>
        <label for="language-names"> select a language:</label>
        <select name="language-names" id="language-names">
            <option value="telugu">telugu</option>
        <option value="english">english</option>
        <option value="hindi">hindi</option>
      </select>
      </center>

      <ul>
        <li> <a href="userhome.jsp">home</a></li>
        <li><a href="browsproposes.jsp">brow</a></li>
        <li><a href="reportdamage.jsp">report</a></li>
        <li><a href="user.html">log out</a></li>
        <li><a href="enterdiscussion.jsp">enter</a></li>

      </ul>
      <% String username=(String)session.getAttribute("username");%>
      <h1>wele come to <%=username%></h1>






      </div>
       <div id="box">
        <image src="t" >
        <h1><b>GREATER HYDERABAD MUNICIPAL
            CORPORATION</b>Government Of Telangana</h1>
       </div>
    <div id="box1">
      <p><b1>for user id  change request,please click on Register yourself.After registration,login
into the page .then
          select userid change request under transanction tab and enter the related field and
submithe request</b1></p>
```

```
      <center>
                          <div id="boy">
                                   <h2> Citizen login</h2>
                                    <table>
                                 <tr>
                                  <th>Phone No</th>
                                   <td><input type="text"  name="phone no"></td>
                                 </tr>

                                <tr>
                                 <th>Password</th>
                               <td><input type="text" name="password"></td>
                               </tr>
                                <tr>

                                   <td><a><button type="submit">Login</button></a></td>
                               </tr>
                               <td>
                                   <a href="userhome.jsp">Reset</a>
                               </td>
                             </table>
                          </div>
                       </center>
   </div>



</body>
```

```
</html>
```

BROWS PROPOSSES;

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>


    <html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>
            #header{
                display: flex;
                height: 200px;background-color: wheat;
                width: 100%;
                margin: 0px;

            }
            #box{
                display: flex;
                height: 120px;
                background-color: lightgrey;
                width: 100%;
                margin: 1px;

            }
            .image{
```

```
    width: 100%;
    height: 120px;
}
b{
    color: red;

}
#box1{
    display: flex;
    height: 700px;
    background-image: url(OIP.jpeg);
    background-repeat: no-repeat;
    background-size: cover;
}
label{
    margin-left: 450px;
    color: blue;
}
p{
    color:  white;
}
 #boy{
                 height: 300px;
                   width: 500px;
                  border:1px solid white;
                  background: lightsteelbluewhitesmoke;
                margin-top: 300px;
                color: white;


 }
 table{
      color: white;
  }
   .header ul  {

     display: inline-block;
     height: 50px;
     width: 100%;
   }
    .header ul li{

     float: center;
     font-size: 20px;
     width: 100%;
     display: inline-block;

     margin: 0px 50px;
     padding-top:  20px;




a{
     text-decoration: none;
     color: blue;
     display: inline-block;



 }
                    a:hover{
     background-color: red;
```

```
        }
                            p{
                                align-items: center;
                                width: 100%;
                                padding-top: 10px;
                                padding-left: 150px;

                                size: 50px;
                            }
                            b1{
                                color: white;
                            }




    </style>
</head>
<body>

    <div id ="header" > <center>
        <label for="language-names"> select a language:</label>
        <select name="language-names" id="language-names">
            <option value="telugu">telugu</option>
            <option value="english">english</option>
            <option value="hindi">hindi</option>
        </select>
        </center>

        <ul>
            <li> <a href="userhome.jsp">home</a></li>
        <li><a href="browsproposes.jsp">brow</a></li>
        <li><a href="reportdamage.jsp">report</a></li>
        <li><a href="user.html">log out</a></li>
        <li><a href="enterdiscussion.jsp">enter</a></li>


      </ul>
      <% String username=(String)session.getAttribute("username");%>
      <h1>welcome to brows proseses <%=username%></h1>






      </div>
       <div id="box">
        <image src="t" >
        <h1><b>GREATER HYDERABAD MUNICIPAL
            CORPORATION</b>Government Of Telangana</h1>
       </div>
    <div id="box1">
      <p><b1>for user id  change request,please click on Register yourself.After registration,login
into the page .then
          select userid change request under transanction tab and enter the related field and
submithe request</b1></p>
      <center>
                   <div id="boy">
                         <h2> Citizen login</h2>
                          <table>
                        <tr>
                         <th>Phone No</th>
                          <td><input type="text"  name="phone no"></td>
```

```
                                               </tr>


                                              <tr>
                                               <th>Password</th>
                                             <td><input type="text" name="password"></td>
                                             </tr>
                                              <tr>


                                                  <td><a><button type="submit">Login</button></a></td>
                                             </tr>
                                             <td>
                                                 <a href="userhome.jsp">Reset</a>
                                             </td>
                                          </table>
                                      </div>
                                  </center>
        </div>







    </body>
</html>
```

REPORT .JSP;

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>


    <html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>
            #header{
                display: flex;
                height: 200px;background-color: wheat;
                width: 100%;
                margin: 0px;

            }
            #box{
                display: flex;
                height: 120px;
                background-color: lightgrey;
                width: 100%;
                margin: 1px;

            }
            .image{
                width: 100%;
                height: 120px;
            }
            b{
                color: red;

            }
```

```
#box1{
    display: flex;
    height: 700px;
    background-image: url(OIP.jpeg);
    background-repeat: no-repeat;
    background-size: cover;
}
label{
    margin-left: 450px;
    color: blue;
}
p{
    color:  white;
}
 #boy{
                 height: 300px;
                   width: 500px;
                  border:1px solid white;
                  background: lightsteelbluewhitesmoke;
                margin-top: 300px;
                color: white;


 }
 table{
     color: white;
 }
  .header ul  {


    display: inline-block;
    height: 50px;
    width: 100%;
  }
   .header ul li{

    float: center;
    font-size: 20px;
    width: 100%;
    display: inline-block;

    margin: 0px 50px;
    padding-top:  20px;




a{
    text-decoration: none;
    color: blue;
    display: inline-block;



 }
                  a:hover{
    background-color: red;
 }
                 p{
                    align-items: center;
                    width: 100%;
                    padding-top: 10px;
                    padding-left: 150px;
```

```
                         size: 50px;
                       }
                       b1{
                         color: white;
                       }





    </style>
</head>
<body>

    <div id ="header" > <center>
        <label for="language-names"> select a language:</label>
        <select name="language-names" id="language-names">
            <option value="telugu">telugu</option>
            <option value="english">english</option>
            <option value="hindi">hindi</option>
        </select>
        </center>

        <ul>
            <li> <a href="userhome.jsp">home</a></li>
            <li><a href="browsproposes.jsp">brow</a></li>
            <li><a href="reportdamage.jsp">report</a></li>
            <li><a href="user.html">log out</a></li>
            <li><a href="enterdiscussion.jsp">enter</a></li>

        </ul>
        <% String username=(String)session.getAttribute("username");%>
```

```
            <h1>welcome to report damage <%=username%></h1>







            </div>
              <div id="box">
                <image src="t" >
                <h1><b>GREATER HYDERABAD MUNICIPAL
                        CORPORATION</b>Government Of Telangana</h1>
              </div>
        <div id="box1">
            <p><b1>for user id  change request,please click on Register yourself.After registration,login
into the page .then
                    select userid change request under transanction tab and enter the related field and
submithe request</b1></p>
            <center>
                                        <div id="boy">
                                                    <h2> Citizen login</h2>
                                                      <table>
                                                  <tr>
                                                   <th>Phone No</th>
                                                     <td><input type="text"  name="phone no"></td>
                                                  </tr>

                                                 <tr>
                                                  <th>Password</th>
                                                <td><input type="text" name="password"></td>
                                                </tr>
                                                 <tr>
```

```
                                                   <td><a><button type="submit">Login</button></a></td>
                                              </tr>
                                              <td>
                                                  <a href="userhome.jsp">Reset</a>
                                              </td>
                                           </table>
                                       </div>
                                   </center>
        </div>




    </body>
</html>
```

ENTER DISCUSION.JSP;

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>


    <html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
```

```
<style>
    #header{
        display: flex;
        height: 200px;background-color: wheat;
        width: 100%;
        margin: 0px;

    }
    #box{
        display: flex;
        height: 120px;
        background-color: lightgrey;
        width: 100%;
        margin: 1px;

    }
    .image{
        width: 100%;
        height: 120px;
    }
    b{
        color: red;

    }
    #box1{
        display: flex;
        height: 700px;
        background-image: url(OIP.jpeg);
        background-repeat: no-repeat;
        background-size: cover;
    }
```

```
label{
    margin-left: 450px;
    color: blue;
}
p{
    color:  white;
}
 #boy{
                 height: 300px;
                   width: 500px;
                  border:1px solid white;
                  background: lightsteelbluewhitesmoke;
                margin-top: 300px;
                color: white;

 }
 table{
     color: white;
 }
  .header ul  {

    display: inline-block;
    height: 50px;
    width: 100%;
  }
   .header ul li{

    float: center;
    font-size: 20px;
    width: 100%;
    display: inline-block;
    margin: 0px 50px;
    padding-top:  20px;




a{
    text-decoration: none;
    color: blue;
    display: inline-block;




 }
                  a:hover{
    background-color: red;
 }
                 p{
                     align-items: center;
                     width: 100%;
                     padding-top: 10px;
                     padding-left: 150px;

                     size: 50px;
                 }
                 b1{
                     color: white;
                 }
    </style>
</head>
<body>

    <div id ="header" > <center>
        <label for="language-names"> select a language:</label>
        <select name="language-names" id="language-names">
            <option value="telugu">telugu</option>
            <option value="english">english</option>
            <option value="hindi">hindi</option>
        </select>
        </center>

        <ul>
            <li> <a href="userhome.jsp">home</a></li>
            <li><a href="browsproposes.jsp">brow</a></li>
            <li><a href="reportdamage.jsp">report</a></li>
            <li><a href="user.html">log out</a></li>
            <li><a href="enterdiscussion.jsp">enter</a></li>

        </ul>
        <% String username=(String)session.getAttribute("username");%>
        <h1>wele come to enter discussion <%=username%></h1>






        </div>
```

```
          <div id="box">
           <image src="t" >
           <h1><b>GREATER HYDERABAD MUNICIPAL
                CORPORATION</b>Government Of Telangana</h1>
          </div>
      <div id="box1">
        <p><b1>for user id  change request,please click on Register yourself.After registration,login
into the page .then
              select userid change request under transanction tab and enter the related field and
submithe request</b1></p>
        <center>
                           <div id="boy">
                                  <h2> Citizen login</h2>
                                    <table>
                                 <tr>
                                  <th>Phone No</th>
                                   <td><input type="text"  name="phone no"></td>
                                 </tr>

                                <tr>
                                 <th>Password</th>
                               <td><input type="text" name="password"></td>
                               </tr>
                                <tr>

                                   <td><a><button type="submit">Login</button></a></td>
                               </tr>
                               <td>
                                  <a href="userhome.jsp">Reset</a>
                               </td>
                             </table>
                             </div>
                          </center>
      </div>



   </body>
</html>
```

Admin log;

```
<html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>
            body{
                background-image: url(a);
                background-repeat: no-repeat;
                background-size: cover;
                height: 100px;
                width: 100%;
            }
            #p1{
                text-align: center;
                color: white;
                padding-top: 50px;
            }
            .bg{
                color: red;
                height: 100vh;
                width: 100vw;

            }
            .imgcls{
                height: 400px;
                width: 800px;
            }
            box{
                height: 250px;
                width: 350px;
                padding: 40px;
                background-color: red;
            }
            h1{
                color: white;
                padding-top: 30px;
            }

        </style>
    </head>
    <body>
        <div id="bg">
            <center>
```

```
                <div id="box">
                    <h1 id="h1">Admin Login</h1>
                    <form  action="adminaction.jsp">
                    <table>
                        <tr>
                            <td style="color: white;">name:</td>
                            <td><input type="text"  name="name" value="name"></td>
                        </tr>
                        <tr>
                            <td style="color: white;">password</td>
                            <td><input type="password"  name="password" value="password"></td>
                        </tr>
                        <tr>

                            <td style="color: white"><a ><button type="submit">login</button></a></td>
                    </tr>

                    </table>
                    </form>



                </div>
            </center>
        </div>
    </body>
```

```
</html>
```

ADMIN.JSP;

```
<%--
    Document   : adminlogaction
    Created on : Jun 28, 2024, 1:53:59 PM
    Author     : hp
--%>


<%@page import="java.sql.ResultSet"%>


<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.DriverManager"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>


<%
    String name=request.getParameter("name");


 String password=request.getParameter("password");
try{
     Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/muncipal","root","r
oot");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select * from tablename1 where name='"+name+"'
and '"+password+"'"");
             if(rs.next()){
                 %>
```

```
                <script>
                    window.alert("logged in successfully.......");
                    window.location="adminhome.jsp"
                </script>
                <%
            }
else{
%>
<script>
                    window.alert("logged failed.......");
                    window.location="admin.html"
                </script>
                <%

}

}
catch(Exception e){
    out.println(e);
}
%>
```

ADMINHOME.JSP;

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
        <style>




        #body{
            background-image: url(l);
            background-repeat: no-repeat;
            background-size: cover;
            height: 100vh;
            width: 100%;
        }
        h1{
            padding-top: 50px;
        }jj
        ::placeholder{
            padding-left: 30px;
            color: black;
        }
        option{
            width: 200px;
            height: 30px;
            margin-left: 30px;
        }
        a:hover{


            background-color: greenyellow;
            width:100% ;


        }




        </style>
    </head>
    <body>


        <div id="body">
            <center>
                <div id="box">
                    <h1 style="color: white;">User Login</h1>
                    <form action="useraction.jsp">
                    <table>
                        <tr>
                            <th style="color: white; ">username:</th>
                            <td><input type="password" name="username"
placeholder="username" value="username"></td>
                        </tr>
                        <tr>
                            <th style="color: white;">password:</th>
                            <td><input type="password" name="password"
placeholder="password"  value="password"></td>
```

```
                        </tr>
                        <tr>
                            <td style="color: white;"><a  ><button type="submit">sign up</button></a></td>
                        </tr>
                    </table>
                    </form>



                </div>
            </center>
        </div>
    </body>
</html>
```

Registration page :

```
<html>
    <head>
        <style>
            #register{
                background-image: url(f);

                width: 100%;
                height: 500px;;
                background-repeat: no-repeat;
                background-size: cover;
            }
            #boy{
                text-align: center;
                background-color: white;
                height: 400px;
                width: 500px;
                padding: 40px;
                border: solid blueviolet;
            }

            }
            h1{
                text-align: center;
                padding-bottom: 50px;
            }



        </style>
        <title>MUNCIPAL CORPORATION</title>
    </head>

        <body>
            <div id="register"> <center>
                    <h1 >Registration form</h1> </center>
        <center>
                       <div class="boy">
                           <form action="departementacton.jsp">


                           <table>
                             <tr>
                           <th style="color: white; ">Name:</th>
                           <td><input type="password" name="Name" placeholder="Name"
value="Name"></td>
                       </tr>




                              <tr>
                           <th style="color: white;">password:</th>
                           <td><input type="password" name="password"
placeholder="password"  value="password"></td>


                       </tr>
                      <tr>
                           <th style="color: white;">phone:</th>
                           <td><input type="phone" name="phone" placeholder="98"
value="phone"></td>


                       </tr>
```

```
                                    <tr>
                            <td style="color: white;"><a><button type="submit">sign up</button></a></td>
                        </tr>


                            </table>
                            </form>



                        </div>
                    </center>
            </div>



        </body>


</html>
```

DEPORTMENT.JSP;

```
<%--
    Document   : departementacton.jsp
    Created on : Jun 28, 2024, 10:46:08 PM
    Author     : hp
--%>
<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
```

```
<%@page import="java.sql.DriverManager"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    String Name=request.getParameter("Name");



                    String password=request.getParameter("password");
                     String phone=request.getParameter("phone");






try{
     Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/muncipal","root","r
oot");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select * from table3 where Name='"+Name+"' and
'"+password+"'  and  '"+phone+"'  ");
            if(rs.next()){
                %>
                <script>
                    window.alert("Register in successfully.......");
                    window.location="departementhome.jsp"
                </script>
                <%
            }
else{
```

```
%>
<script>
                    window.alert("Registration failed.......");
                    window.location="departement.html"
                </script>
                <%

}


}
catch(Exception e){
    out.println(e);
}
%>
```

DEPORTEMENT HOME.JSP;

```
<%--
    Document   : userloginhome.jsp
    Created on : Jun 28, 2024, 2:37:43 PM
    Author     : hp
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>



    <html>
    <head>
        <title>MUNCIPAL CORPORATION</title>
```

```
<style>
    #header{
        display: flex;
        height: 50px;background-color: wheat;
        width: 100%;
        margin: 0px;

    }
    #box{
        display: flex;
        height: 120px;
        background-color: lightgrey;
        width: 100%;
        margin: 1px;

    }
    .image{
        width: 100%;
        height: 120px;
    }
    b{
        color: red;

    }
    #box1{
        display: flex;
        height: 700px;
        background-image: url(bilding);
    background-repeat: no-repeat;
    background-size: cover;
}
label{
    margin-left: 450px;
    color: blue;
}
p{
    color:  white;
}
 #boy{
                 height: 300px;
                   width: 500px;
                  border:1px solid white;
                  background: lightsteelbluewhitesmoke;
                margin-top: 300px;
                color: white;

 }
 table{
     color: white;
 }

a{
                              color: white;
                              text-decoration: none;
                     background-color: red;
                     margin-left: 50px;
```

```
                                     }
                            p{
                                align-items: center;
                                width: 100%;
                                padding-top: 10px;
                                padding-left: 150px;

                                size: 50px;
                            }
                            b1{
                                color: black;
                            }



    </style>
</head>
<body>
    <div id ="header" > <center>
        <label for="language-names"> select a language:</label>
        <select name="language-names" id="language-names">
            <option value="telugu">telugu</option>
            <option value="english">english</option>
            <option value="hindi">hindi</option>
        </select>
        </center>
    </div>
      <div id="box">
        <image src="t" >
        <h1><b>GREATER HYDERABAD MUNICIPAL
            CORPORATION</b>Government Of Telangana</h1>


      </div>
    <div id="box1">
      <p><b1>for user id  change request,please click on Register yourself.After
registration,login into the page .then
          select userid change request under transanction tab and enter the related
field and submithe request</b1></p>
      <center>
                   <div id="boy">
                        <h2> Citizen login</h2>
                         <table>
                       <tr>
                        <th>Phone No</th>
                         <td><input type="text"  name="phone no"></td>
                       </tr>


                      <tr>
                       <th>Password</th>
                      <td><input type="text" name="password"></td>
                      </tr>
                      <tr>
                       <th>City Name</th>
                      <td><input type="text" name="City Name"></td>
```

```
                                    </tr>
                                     <tr>


                                        <td><a><button
type="submit">Login</button></a></td>
                                    </tr>
                                    <td>
                                       <a href="departementhome.jsp">Reset</a>
                                    </td>
                                  </table>
                               </div>
                            </center>
     </div>







     </div>
   </body>
</html>
```

CHAPTER-10

RESULTS/DISCUSSIONS

10.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.1.1 TEST CASES:

Test case1 for Login form:

FUNCTION:

LOGIN

EXPECTED RESULTS:

Should Validate the user and check his existence in database

ACTUAL RESULTS:

Validate the user and checking the user against the database

LOW PRIORITY

No

HIGH PRIORITY

Yes

Test case2:

Test case for User Registration form:

FUNCTION:

USER REGISTRATION

EXPECTED RESULTS:

Should check if all the fields are filled by the user and saving the user to database.

ACTUAL RESULTS:

Checking whether all the fields are field by user or not through validations and saving user.

LOW PRIORITY

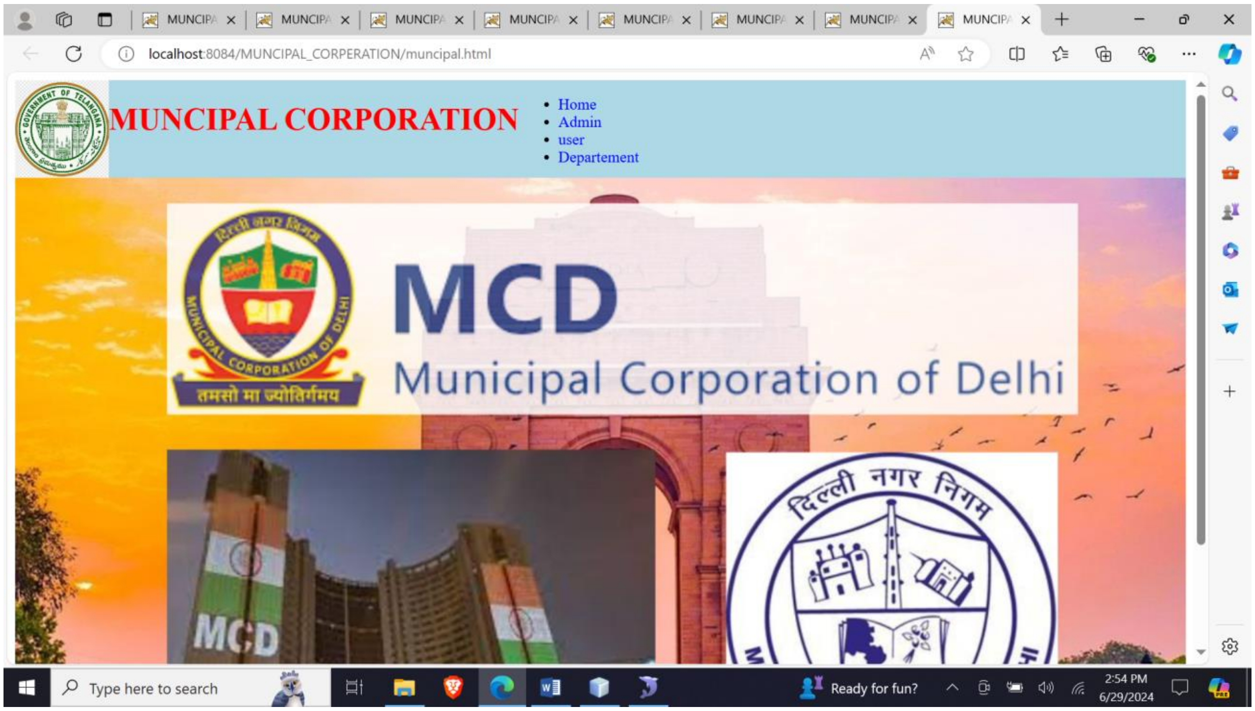
No

HIGH PRIORITY

Yes

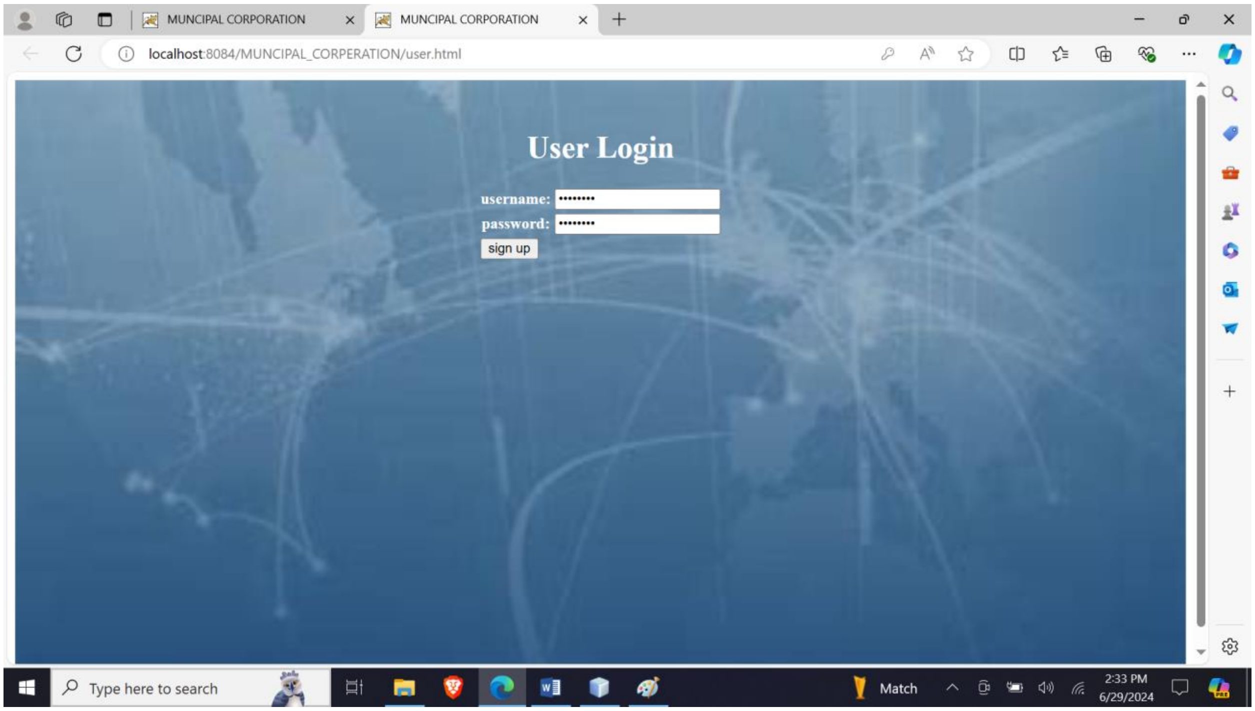
10.2 SCREENSHOTS:

HOMEPAGE:

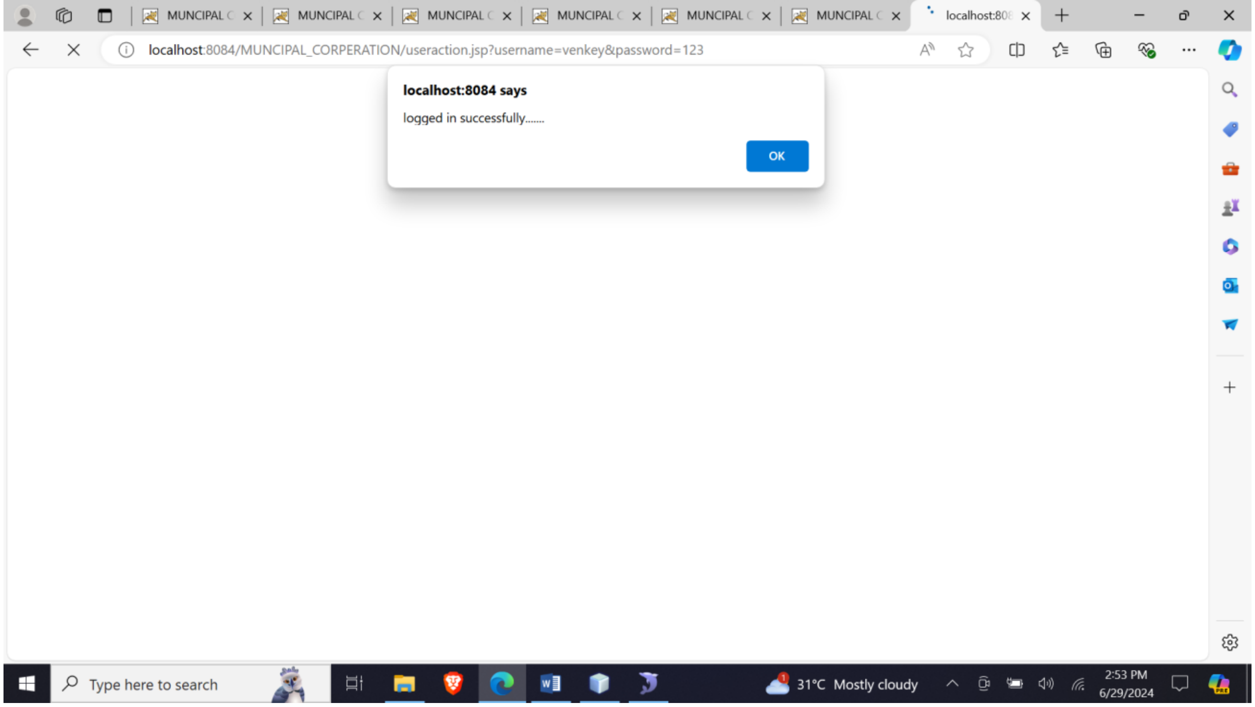


USER PAGE:

user login entails accessing a system or application by entering a unique combination of credentials, typically a username and password. Security may be reinforced through additional measures such as Multi-Factor Authentication (MFA).



USERACTION.JSP;

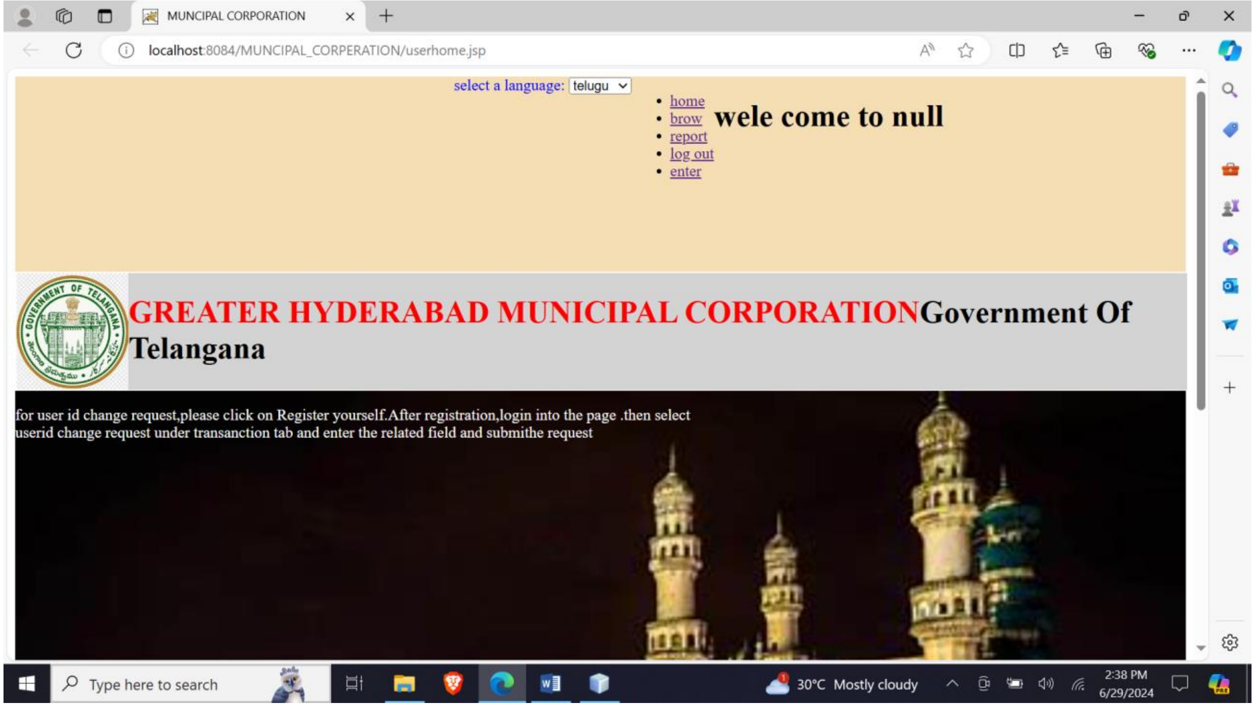


localhost:8084/MUNCIPAL_CORPERATION/useraction.jsp?username=venkey&password=123

localhost:8084 says

logged in successfully.......

USERHOME.JSP;



localhost:8084/MUNCIPAL_CORPERATION/userhome.jsp

select a language: telugu

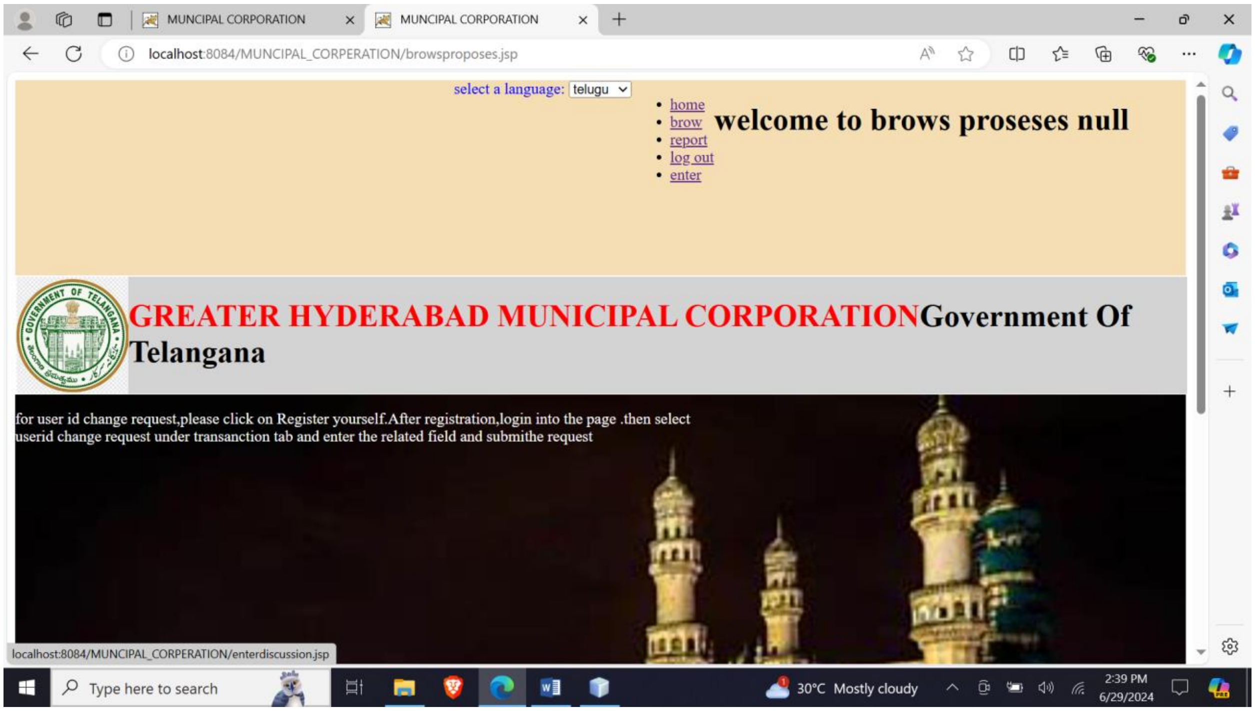
- home
- brow
- report
- log out
- enter

wele come to null

GREATER HYDERABAD MUNICIPAL CORPORATIONGovernment Of Telangana

for user id change request,please click on Register yourself.After registration,login into the page .then select userid change request under transaction tab and enter the related field and submithe request

BROWS PROSESSES.JSP;



MUNCIPAL CORPORATION
MUNCIPAL CORPORATION
localhost:8084/MUNCIPAL_CORPERATION/browsproposes.jsp
select a language: telugu
home
brow
report
log out
enter

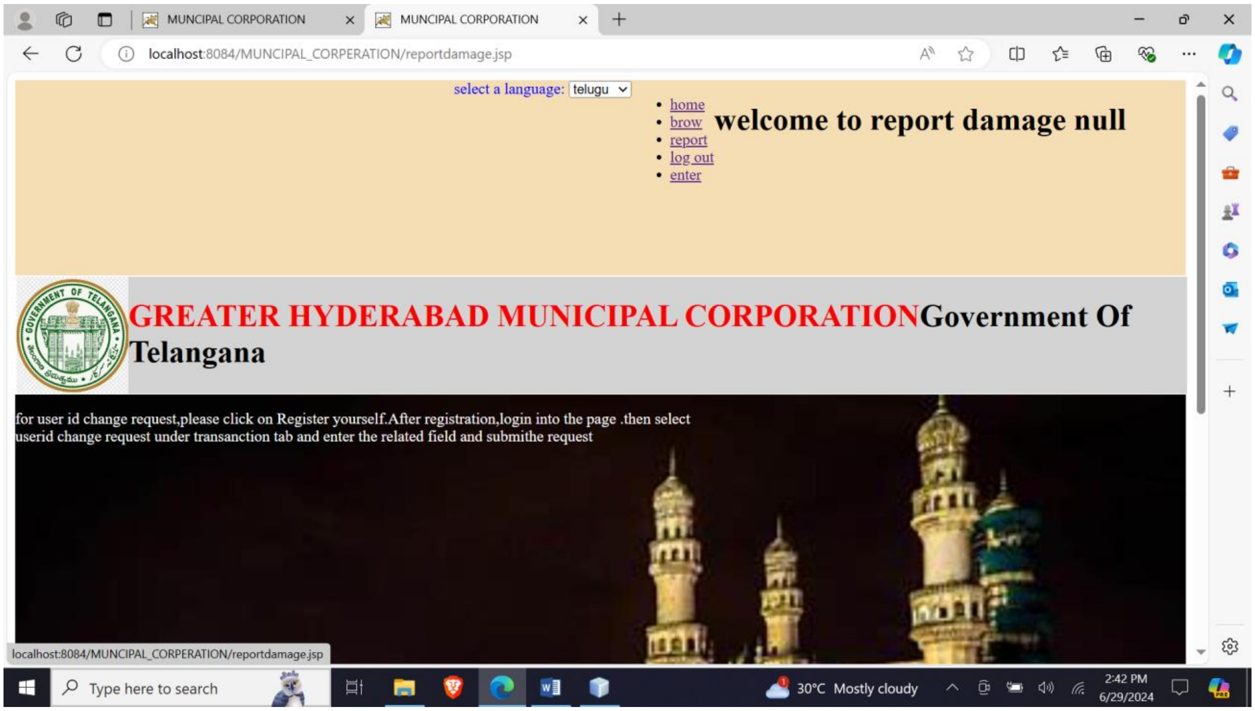
welcome to brows proseses null

GREATER HYDERABAD MUNICIPAL CORPORATIONGovernment Of Telangana

for user id change request,please click on Register yourself.After registration,login into the page .then select userid change request under transaction tab and enter the related field and submithe request

localhost:8084/MUNCIPAL_CORPERATION/enterdiscussion.jsp
Type here to search
30°C Mostly cloudy
2:39 PM
6/29/2024

REPORT DAMAGE.JSP;



MUNCIPAL CORPORATION
MUNCIPAL CORPORATION
localhost:8084/MUNCIPAL_CORPERATION/reportdamage.jsp
select a language: telugu
home
brow
report
log out
enter

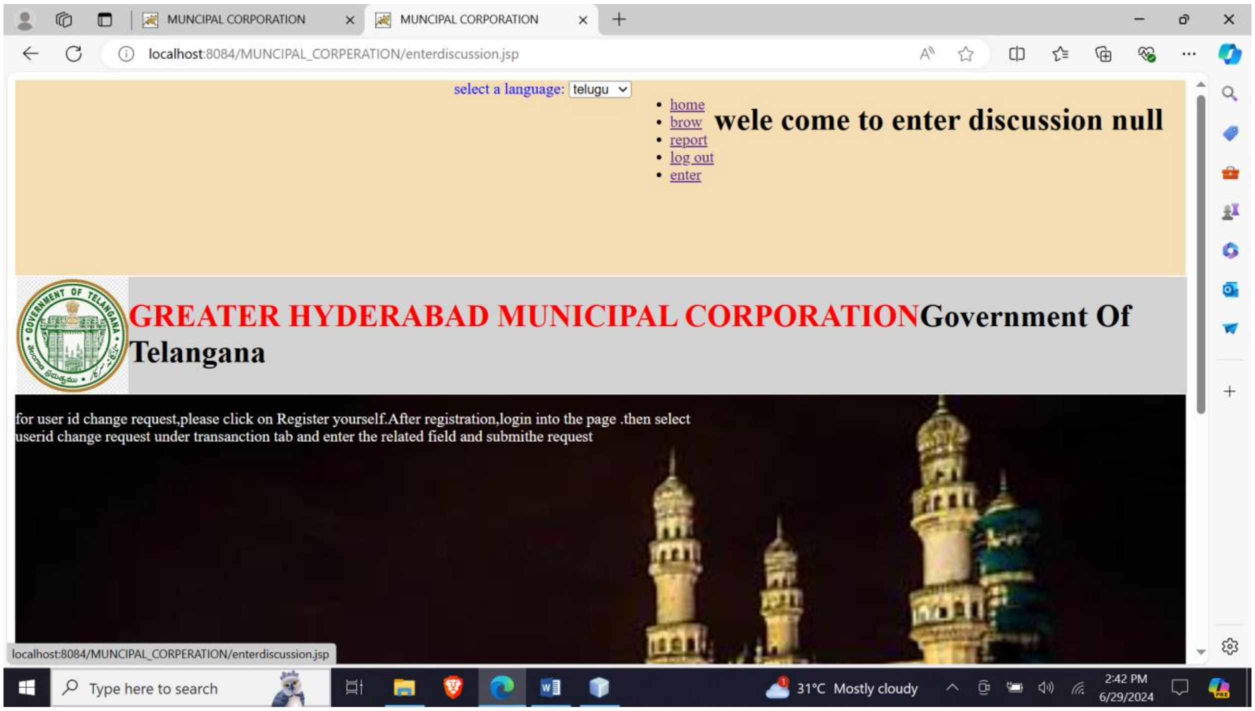
welcome to report damage null

GREATER HYDERABAD MUNICIPAL CORPORATIONGovernment Of Telangana

for user id change request,please click on Register yourself.After registration,login into the page .then select userid change request under transaction tab and enter the related field and submithe request

localhost:8084/MUNCIPAL_CORPERATION/reportdamage.jsp
Type here to search
30°C Mostly cloudy
2:42 PM
6/29/2024

ENTER DISCUSSION.JSP;



select a language: telugu

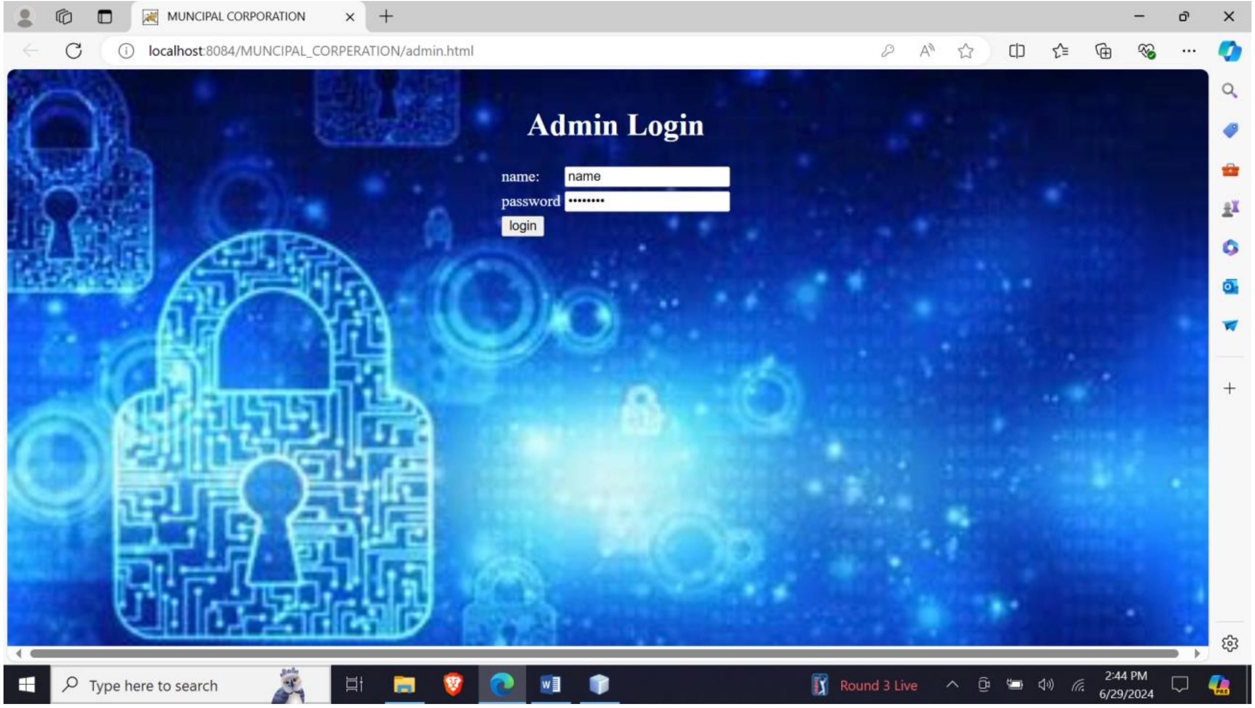
wele come to enter discussion null

GREATER HYDERABAD MUNICIPAL CORPORATIONGovernment Of Telangana

for user id change request,please click on Register yourself.After registration,login into the page .then select userid change request under transanction tab and enter the related field and submithe request

ADMIN:

Users enter their credentials, usually a combination of a username or email address and a password. Many providers enhance security through Multi-Factor Authentication (MFA), requiring a secondary verification step. Once logged in, users gain access to a dashboard where they can manage services and resources.



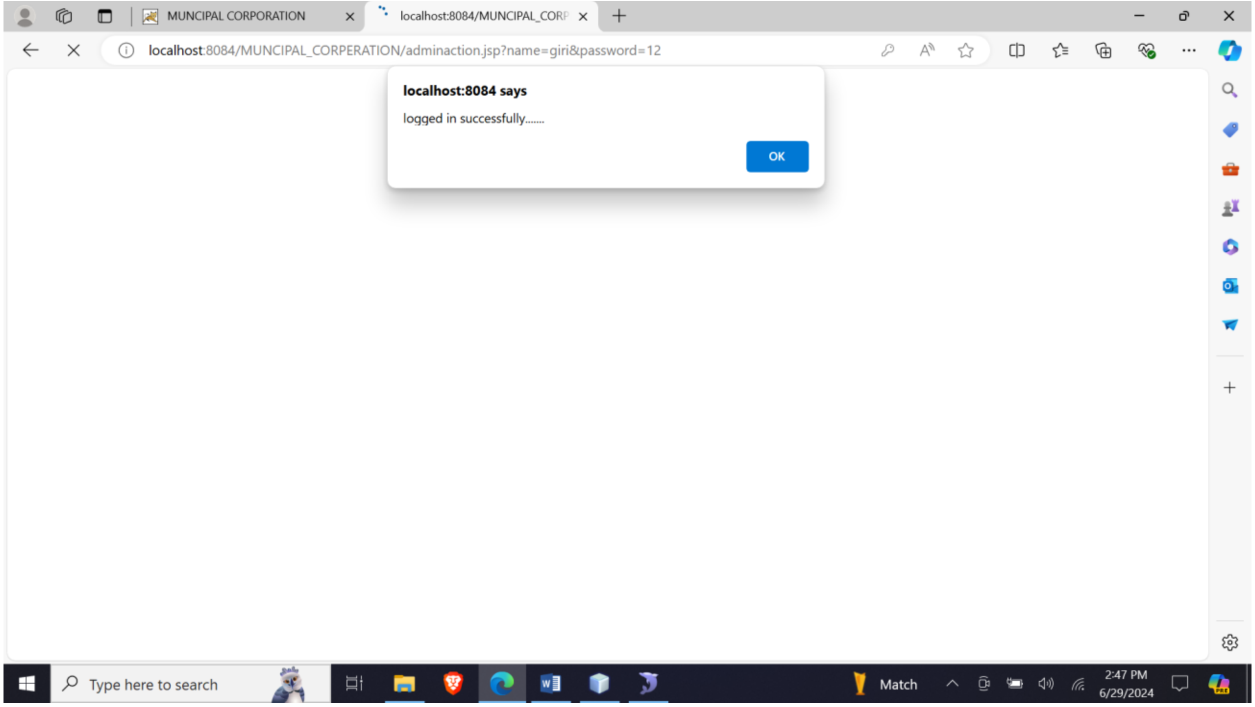
localhost:8084/MUNCIPAL_CORPERATION/admin.html

Admin Login

name: name

password ••••••••

ADMIN.JSP;

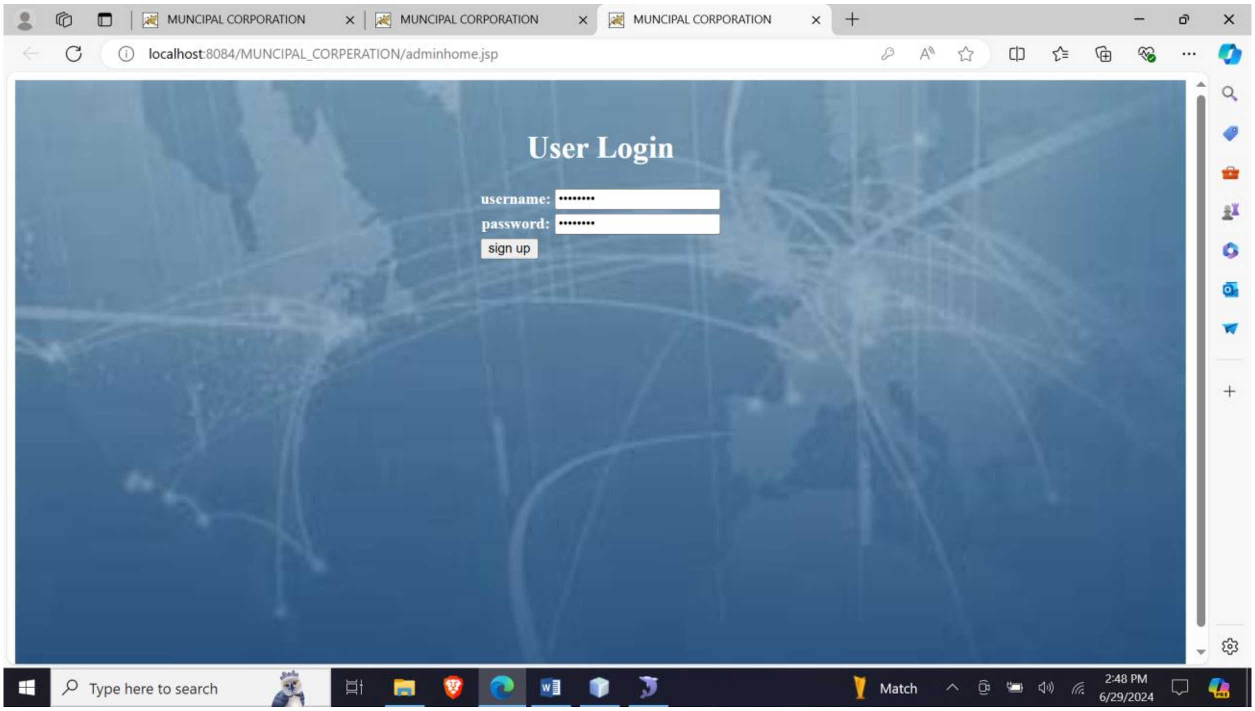


localhost:8084/MUNCIPAL_CORPERATION/adminaction.jsp?name=giri&password=12

localhost:8084 says

logged in successfully.......

ADMINHOME.JSP;



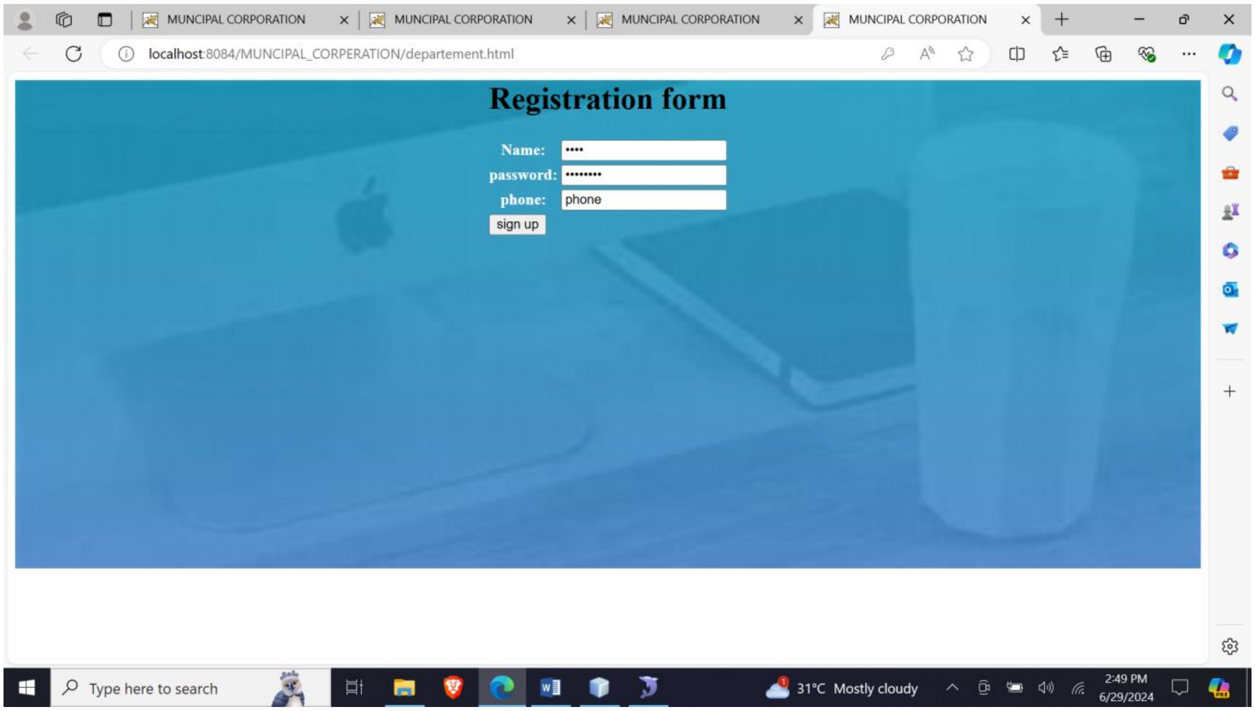
User Login

username: ••••••••

password: ••••••••

sign up

REGISTRATION:



Registration form

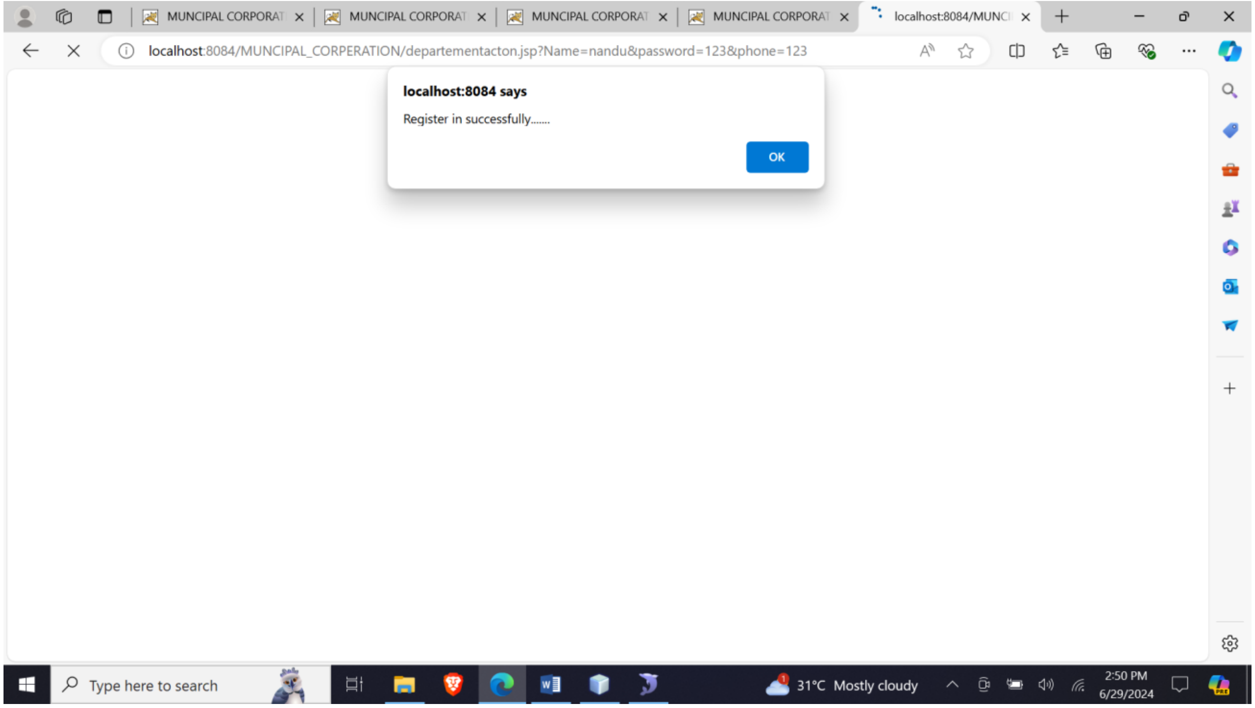
Name: ••••

password: ••••••••

phone: phone

sign up

DEPARTMENT.JSP;

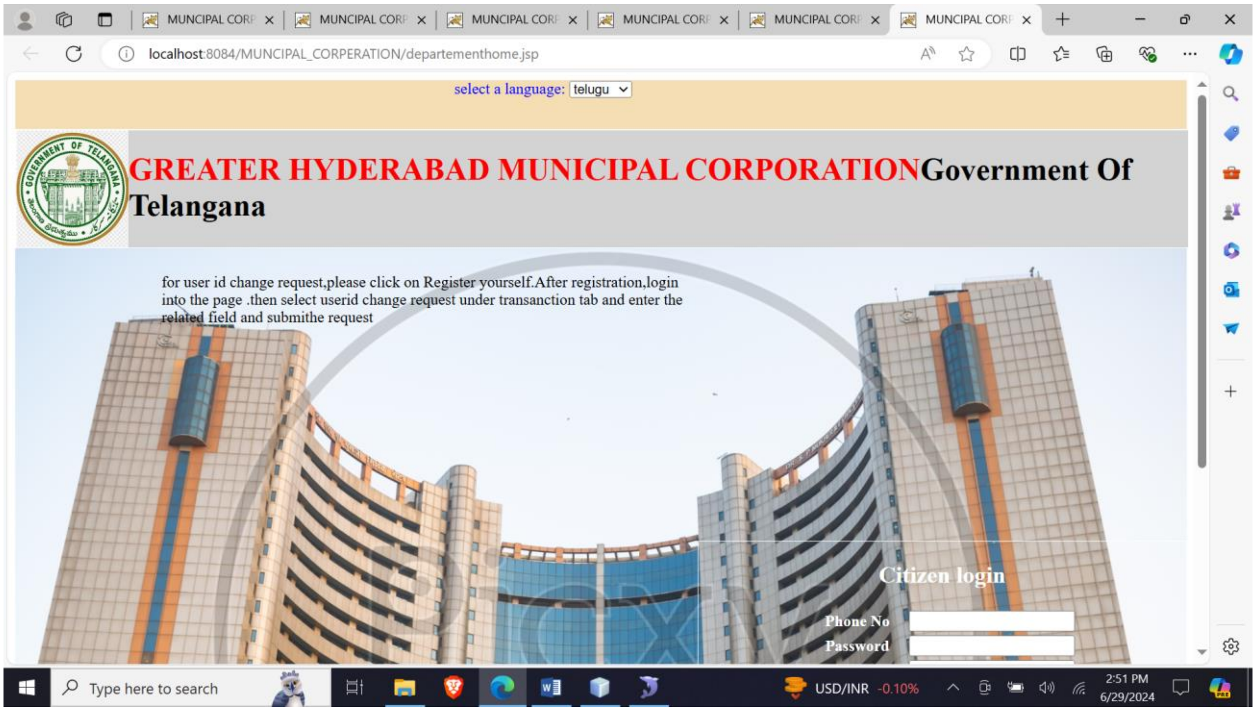


localhost:8084/MUNCIPAL_CORPERATION/departementacton.jsp?Name=nandu&password=123&phone=123

localhost:8084 says

Register in successfully.......

DEPARTMENTHOME.JSP;



localhost:8084/MUNCIPAL_CORPERATION/departementhome.jsp

select a language: telugu

GREATER HYDERABAD MUNICIPAL CORPORATIONGovernment Of Telangana

for user id change request,please click on Register yourself.After registration,login into the page .then select userid change request under transanction tab and enter the related field and submithe request

Citizen login

Phone No

Password

CHAPTER-11

CONCLUSION

Municipal corporations are established in large cities with a population of more than one million. They have significant powers and responsibilities. These responsibilities are providing essential services, maintaining public infrastructure, ensuring public health and sanitation, promoting education and culture, and supporting tourism and recreation. Municipalities and Nagar Panchayats are established in smaller urban areas. They also play an important role in providing basic civic amenities and promoting local development.

Over the years, the role of municipalities in India has evolved, and they have become increasingly important. They address the challenges faced by urban areas, including rapid urbanization, population growth, and the need for sustainable development. Through effective governance, planning, and management, municipalities in India can help create livable and vibrant urban areas that meet the needs of their citizens.

11.2 FUTURE SCOPE

The future scope for municipal corporations is vast and pivotal in shaping the urban landscape. As our cities continue to grow, municipal corporations will play a crucial role in addressing emerging challenges and harnessing opportunities for sustainable development.One key aspect is the integration of smart technologies. Municipalities are increasingly adopting smart solutions to enhance efficiency in service delivery, infrastructure management, and citizen engagement. From smart waste management systems to automated traffic control, these technologies can streamline operations, reduce costs, and improve the overall quality of urban life.Environmental sustainability will be a focal point for municipal corporations in the future.

CHAPTER-12

REFERENCES

Daniel Edmund O’Leary, "Google’s Duplex: Pretending to be human", *Intelligent Systems in Accounting Finance & Management* 1, pp. 46-53, January 2019.

2. Mareeswari and V. Gopalakrishnan, "Complaint go: an online complaint registration system using web services and android. IOP Conference Series: Materials Science and Engineering", vol. 263, pp. 042073, 2017.

3.S. Balakrishnan, J.P. Ananth, L. Ramanathan, R. Sachinkanithkar and S. Rajkumar, "A High Security Framework Through Human Brain Using Algo Mixture Model Deep Learning Algorithm", *Deep Learning in Data Analytics. Studies in Big Data*, vol. 91, 2022

S Balakrishnan, K. Aravind and A. Jebaraj Ratnakumar, "A Novel Approach for Tumor Image Set Classification Based On Multi-Manifold Deep Metric Learning", *International Journal of Pure and Applied Mathematics*, vol. 119, no. 10c, pp. 553-562, 2018.

Sunil Kopparapu, Akhilesh Srivastava, and PVS Rao. A natural language interface for a railway website. In Second National Conference on Innovations in Information and Communication Technology 2006, 7-8 July, PSG College of Technology - Coimbatore., 20

Bobillier Chaumon and Sandoz Guermond. Study of conditions of use of e-services accessible to visually disabled persons. In Proceedings DEGAS 2007.