|  |
| --- |
| **Web Application For Breast Cancer Prediction** |

**OBJECTIVE**

The main aim of this project is to implementing a web application for breast cancer prediction and providing existing predicted reports to users.

**ABSTRACT**

**Abstract:** Breast cancer is one of the most common and major causes of cancer in women. Currently, it has become a common health problem and recently its incidence has increased. Early detection is the best way to manage breast cancer results. Computer-aided detection or diagnosis (CAD) systems play a major role in early detection of breast cancer and can be used to reduce mortality in women. The main purpose of this paper is to make use of recent advances in the development of CAD systems and related technologies. The main theme of the project is to assess whether a person is suffering from breast cancer. Machine learning is nothing more than training machines to learn and operate on their own without any explicit program or instruction. So here, whether a person is suffering from breast cancer or not can be assessed with the help of trained data.

**INTRODUCTION**

Breast cancer is a disease in which cells in the breast grow out of control. There are different types of breast cancers. The type of breast cancer depends on which cells in the breast become cancerous.

Breast cancer begins in different parts of the breast. The breast is made up of three main parts: the lobules, the vessels, and the connective tissue. Lobules are glands that produce milk. The vessels are the tubes that carry milk to the nipple. It surrounds the connective tissue (which is composed of fiber and adipose tissue) and holds it all together. Most breast cancers begin in the vessels or lobules.

**The Most Common Types of Breast Cancer**

Invasive ductal carcinoma. Cancer cells begin in the vessels and grow into other parts of the breast tissue outside the vessels. Invasive cancer cells can spread to other parts of the body or even metastasize.

Invasive lobular carcinoma. Cancer cells begin in the lobules and then spread from the lobules to nearby breast tissues. These invasive cancer cells can spread to other parts of the body.

**Symptoms of Breast Cancer**

Different people have different symptoms of breast cancer. Some people do not have any signs or symptoms at all.

Some warning signs of breast cancer are—

* New lump in the breast or underarm (armpit).
* Thickening or swelling of part of the breast.
* Irritation or dimpling of breast skin.
* Redness or flaky skin in the nipple area or the breast.
* Pulling in of the nipple or pain in the nipple area.
* Nipple discharge other than breast milk, including blood.
* Any change in the size or the shape of the breast.
* Pain in any area of the breast.
* Keep in mind that these symptoms can happen with other conditions that are not cancer.

**What age does breast cancer develop?**

Your risk for breast cancer increases as you age. About 80% of women diagnosed with breast cancer each year are ages 45 or older, and about 43% are ages 65 or above. Consider this: In women ages 40 to 50, there is a one in 69 risk of developing breast cancer. From ages 50 to 60, that risk increases to one in 43.

**Life after breast cancer treatment**

You may expect your life to go back to the way it was before, but instead of returning to your old normal, you may have to adjust to a new one. With today's advanced treatment and early detection, breast cancer survivors can live a long and full life after breast cancer treatment.

**LITERATURE REVIEW**

Literature [survey](http://www.blurtit.com/q876299.html) is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the [programmers](http://www.blurtit.com/q876299.html) start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from [book](http://www.blurtit.com/q876299.html) or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

**[1]** **Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis**

Each year number of deaths is increasing extremely because of breast cancer. It is the most frequent type of all cancers and the major cause of death in women worldwide. Any development for prediction and diagnosis of cancer disease is capital important for a healthy life. Consequently, high accuracy in cancer prediction is important to update the treatment aspect and the survivability standard of patients. Machine learning techniques can bring a large contribute on the process of prediction and early diagnosis of breast cancer, became a research hotspot and has been proved as a strong technique. In this study, we applied five machine learning algorithms: Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision tree (C4.5) and K-Nearest Neighbours (KNN) on the Breast Cancer Wisconsin Diagnostic dataset, after obtaining the results, a performance evaluation and comparison is carried out between these different classifiers. The main objective of this research paper is to predict and diagnosis breast cancer, using machine-learning algorithms, and find out the most effective whit respect to confusion matrix, accuracy and precision. It is observed that Support vector Machine outperformed all other classifiers and achieved the highest accuracy (97.2%).All the work is done in the Anaconda environment based on python programming language and Scikit-learn library.

**[2] Automated Breast Cancer Detection and Classification Techniques – A survey**

Breast cancer (BC) is the world's second leading cause of death for women. Because the cause of the disease is unknown, early detection and diagnosis are critical for BC control, as they can improve treatment success, save lives, and lower costs. Gene mutation, changes in size, and breast skin texture are indicators of BC. Symptoms must be carefully investigated to offer appropriate care to patients and an automatic prediction system that can classify tumors as benign or malignant is required. The majority of data generated in today's internet world is collected on social media or healthcare websites. Using data mining (DM) techniques, symptoms can be extracted from this massive amount of data, which will be helpful in the identification and classification of BC. The main contributions in this study are to investigate major problems that faces BC diagnosis and classification methods, as well as to present an overview of recent research studies that used to identify and classify BC.

**[3] Breast Cancer Prediction: A Comparative Study Using Machine Learning Techniques**

Early detection of disease has become a crucial problem due to rapid population growth in medical research in recent times. With the rapid population growth, the risk of death incurred by breast cancer is rising exponentially. Breast cancer is the second most severe cancer among all of the cancers already unveiled. An automatic disease detection system aids medical staffs in disease diagnosis and offers reliable, effective, and rapid response as well as decreases the risk of death. In this paper, we compare five supervised machine learning techniques named support vector machine (SVM), K-nearest neighbors, random forests, artificial neural networks (ANNs) and logistic regression. The Wisconsin Breast Cancer dataset is obtained from a prominent machine learning database named UCI machine learning database. The performance of the study is measured with respect to accuracy, sensitivity, specificity, precision, negative predictive value, false-negative rate, false-positive rate, F1 score, and Matthews Correlation Coefficient. Additionally, these techniques were appraised on precision–recall area under curve and receiver operating characteristic curve. The results reveal that the ANNs obtained the highest accuracy, precision, and F1 score of 98.57%, 97.82%, and 0.9890, respectively, whereas 97.14%, 95.65%, and 0.9777 accuracy, precision, and F1 score are obtained by SVM, respectively.

**[4]** **Breast cancer detection by leveraging Machine Learning**

India has witnessed 30% of the cases of breast cancer during the last few years and it is likely to increase. Breast cancer in India accounts that one woman is diagnosed every two minutes and every nine minutes, one woman dies. Early detection and diagnosis can save the lives of cancer patients. This paper presents a novel method to detect breast cancer by employing techniques of Machine Learning. The authors carried out an experimental analysis on a dataset to evaluate the performance. The proposed method has produced highly accurate and efficient results when compared to the existing methods.

**SYSTEM ANALYSIS & FEASIBILITY STUDY**

**EXISTING METHOD:**

In an existing system, it is important for physicians to diagnose various conditions related to Brest cancer Detection at an early stage because early detection can prevent or reverse kidney damage. Early detection can provide better treatment and proper care for patients. Many regional hospitals / clinics have a shortage of nephrologists or general practitioners to diagnose symptoms. Over time, companies will be able to implement the proposed machine learning framework in regional clinics with less medical professional retention, providing early diagnosis to patients in regional areas. This process can take a long time and is even more costly.

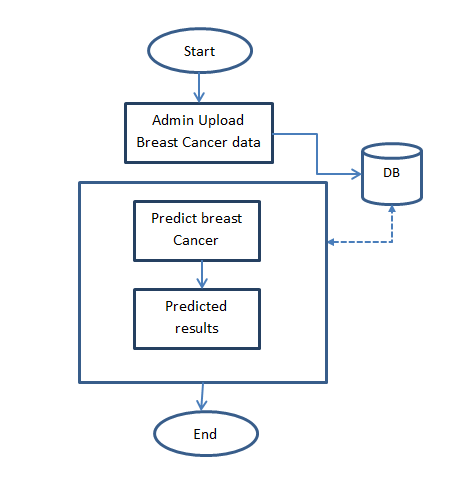
**Disadvantages:**

* Time efficiency is high
* Have manual work is high
* It is costly

**PROPOSED METHOD:**

In proposed system, we are predicting the breast cancer using Existing records. In this we are initially collect the existing records then store into our system. Then we predict the results for Brest cancer based on their parameter values. This procedure can be have taken less time and cost sufficient.

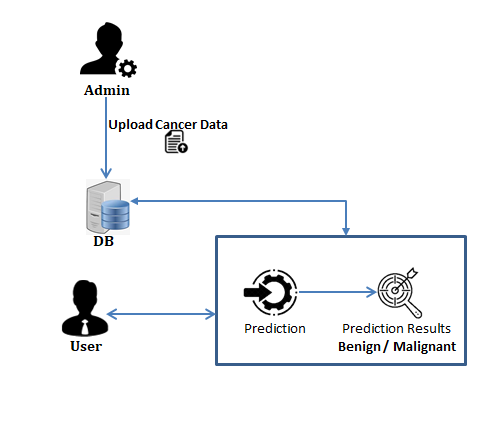
**Block Diagram**



**ADVANTAGES**

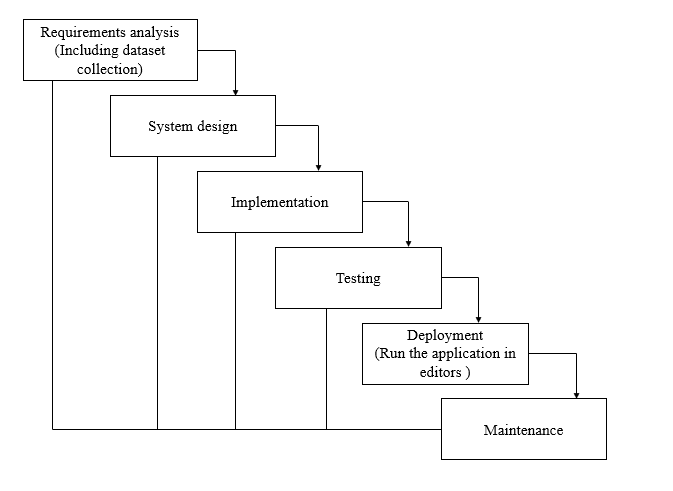
* Less time efficiency
* Reducing manual work
* Less cost

**ARCHITECTURE**



**SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC**

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.



Waterfall Model

* **Requirement Gathering and analysis** − all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**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.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**Economic 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 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.

**SYSTEM REQUIREMENTS SPECIFICATION**

**Non Functional Requirement**

Hence few things to be noted before Mobile Application Development are

* Unlike laptops/Desktop the resources available on Mobile devices are very less like processor, speed, screen dimension etc.
* Scalability of device screen is different for different mobiles.
* Network condition may vary.
* Multitasking capability and Memory available.
* Different version of OS and backward compatibility etc.

Moving on let me List out few Key types of NFR thats needs to be taken care of

* Performance
* Scalability
* Responsiveness
* Use-ability
* reliability
* Security
* Documentation
* Availability

**Functional Requirements**

Such requirements describe system behavior under specific conditions and include the product features and functions which web & app developers must add to the solution. Such requirements should be precise both for the development team and stakeholders.

The list of examples of functional requirements includes:

* Business Rules
* Transaction corrections, adjustments, and cancellations
* Administrative functions
* Authentication
* Authorization levels
* Audit Tracking
* External Interfaces
* Certification Requirements
* Reporting Requirements
* Historical Data

**HARDWARE & SOFTWARE REQUIREMENTS**

# H/W System Configuration:-

# Processor - I3/Intel Processor

* RAM - 4GB (min)
* Hard Disk - 500GB

**S/W System Configuration:-**

* Operating System : Windows 7/8/10
* Application Server : Tomcat 7.0
* Front End : HTML, JSP
* Scripts : JavaScript.
* Server side Script : Java Server Pages.
* Database : My SQL 6.0
* Database Connectivity : JDBC

**SYSTEM DESIGN**

## Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### Objectives for Input Design:

The objectives of input design are

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design:

The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**MODULES:**

**Admin**: admin login with valid credentials. Upload Existing recorders. View users. Add new record.

**Users**: Users initially register with their details then login with registered details, predict breast cancer and view their existing predicted results.

**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 OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**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.

****

**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.

****

**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.



**Collaboration Diagram:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**COMPONENT DIAGRAM**

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation.

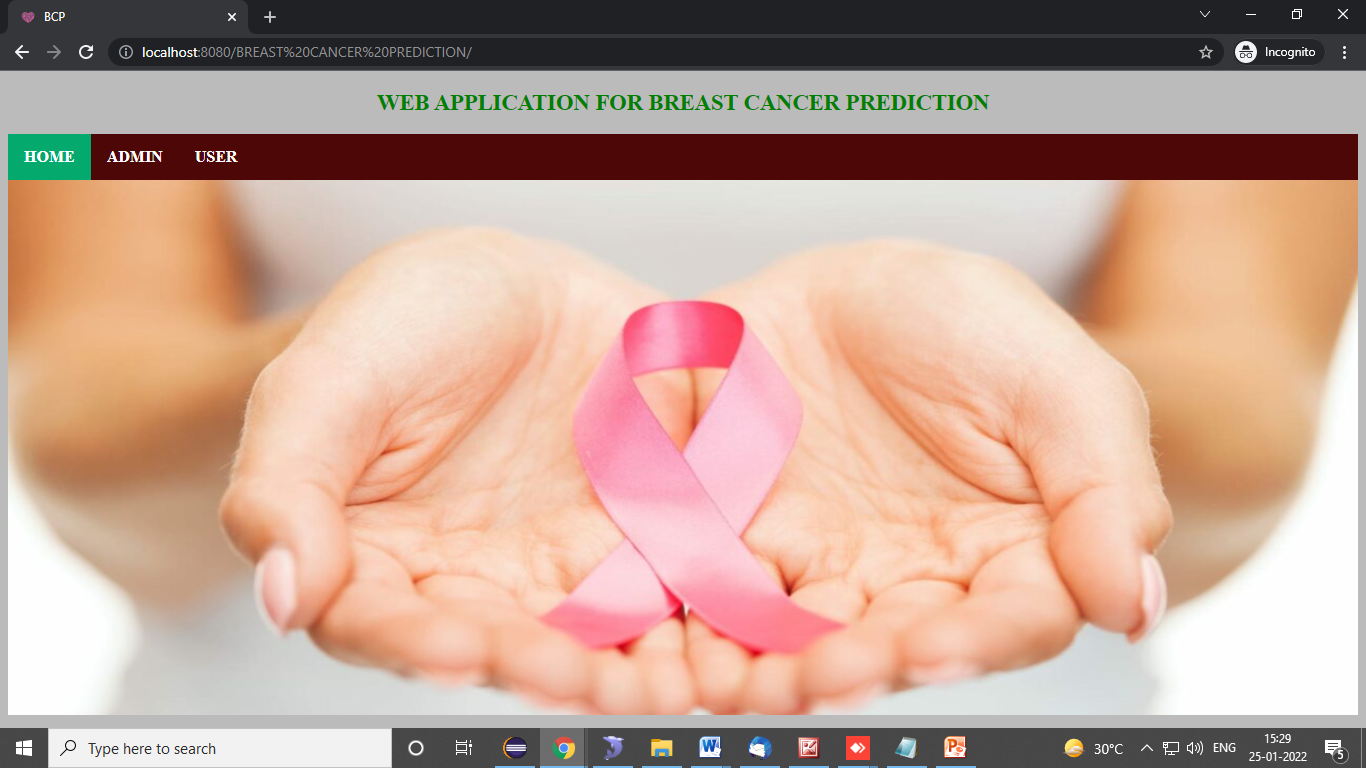


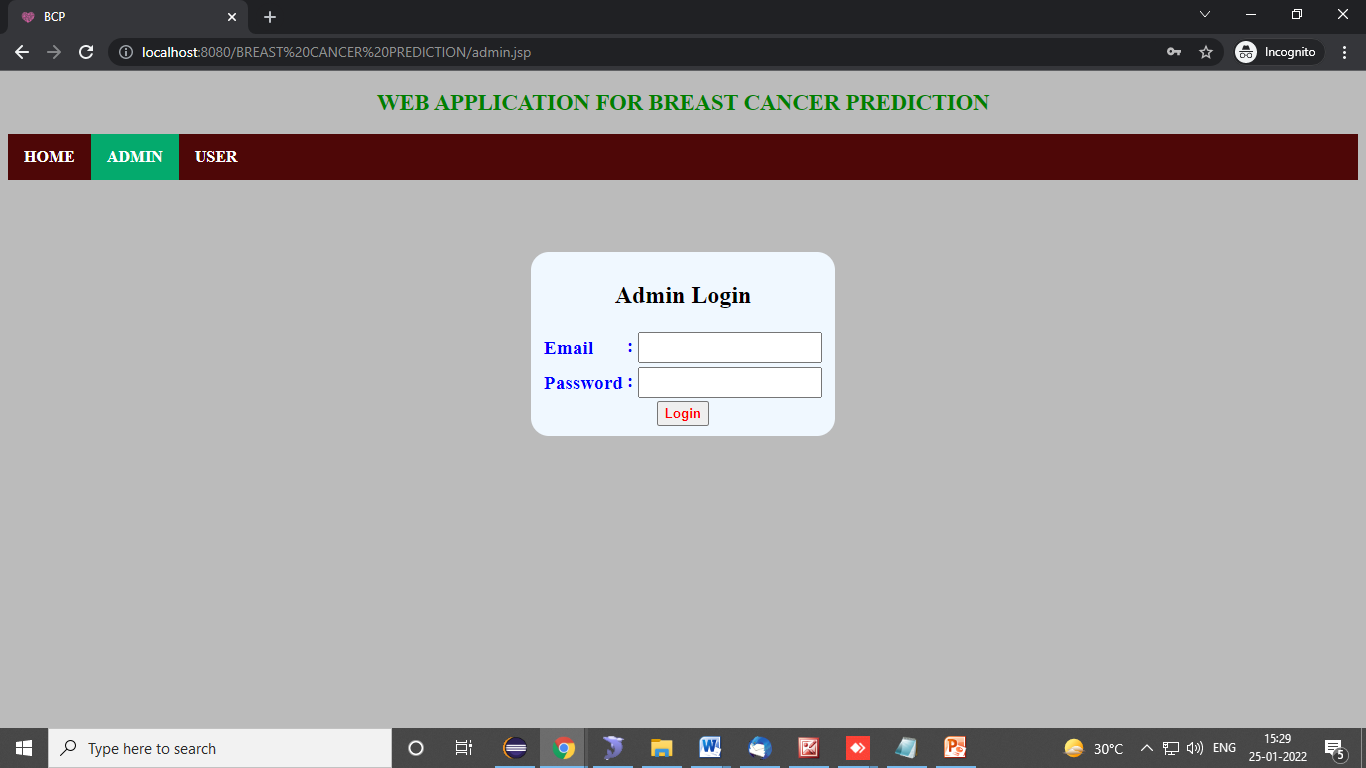
**DEPLOYMENT DIAGRAM**

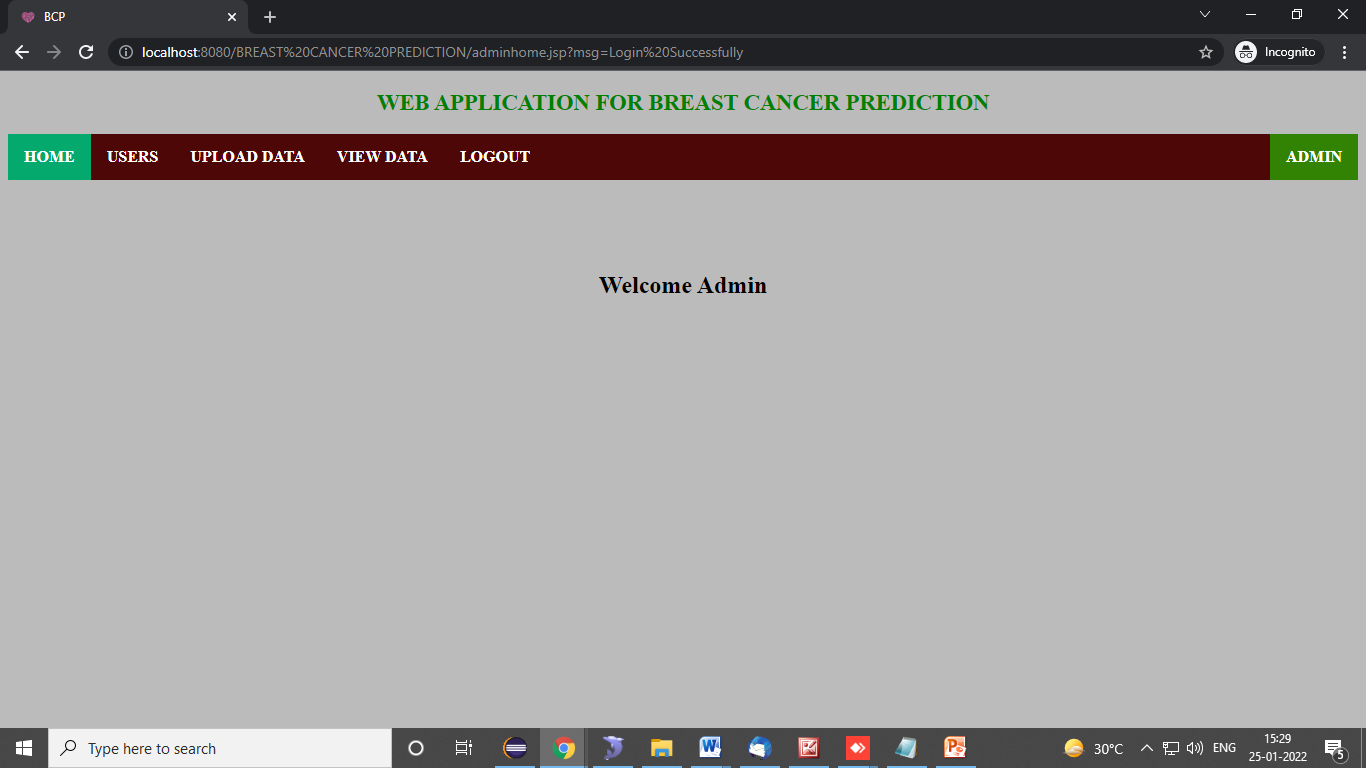
Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hard ware’s used to deploy the application.



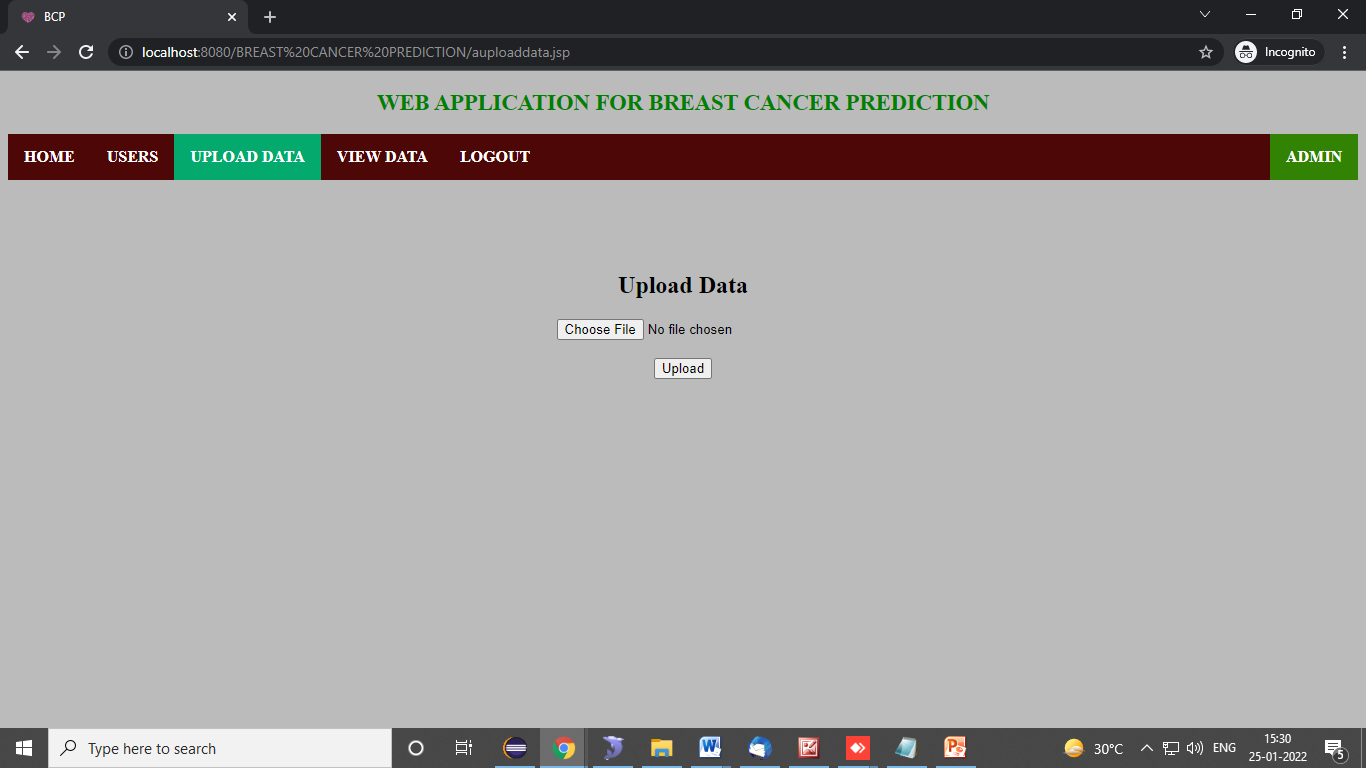
**OUTPUT SCREEN SHOTS WITH DESCRIPTION.**

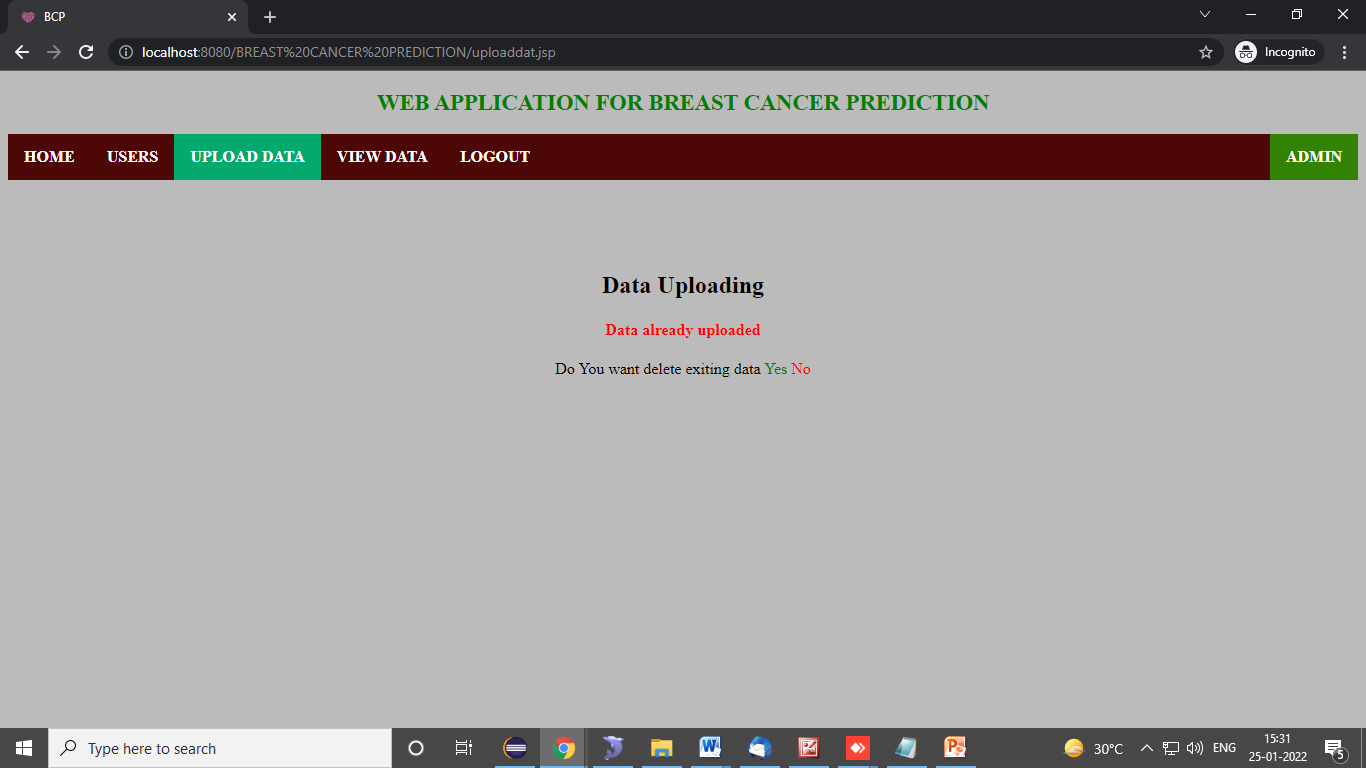
Home : this is the intial page of our project

Admin login : this admin login page

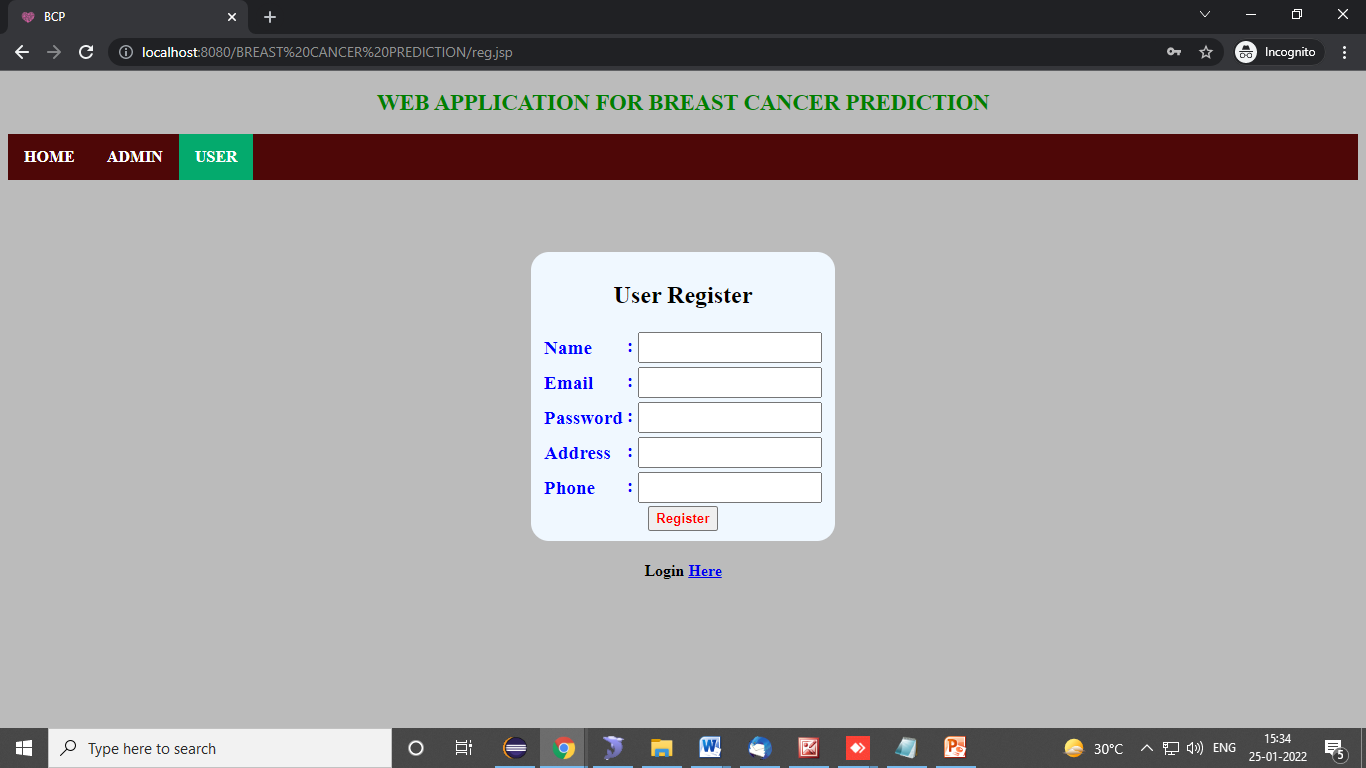
Admin home : this is admin home page. In this we can view all available admin features. 

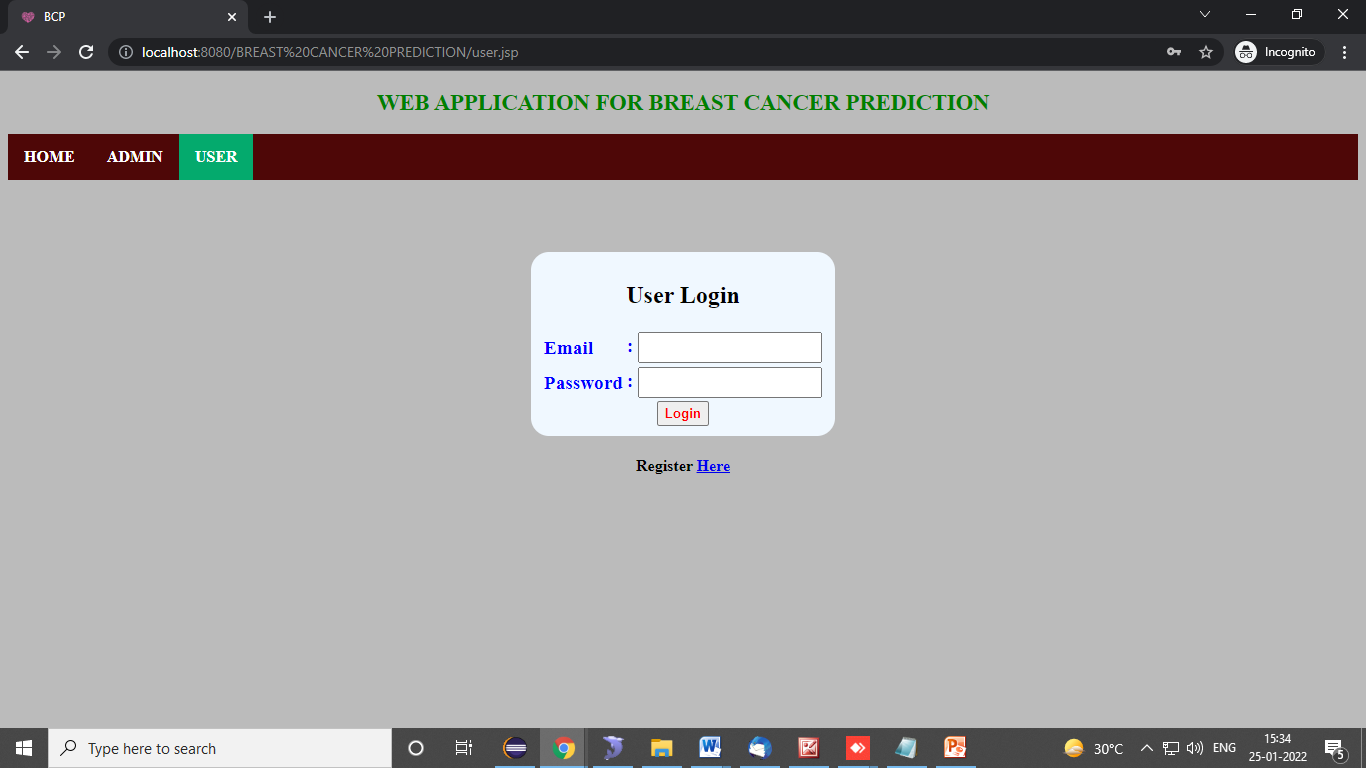
Users: admin view user details

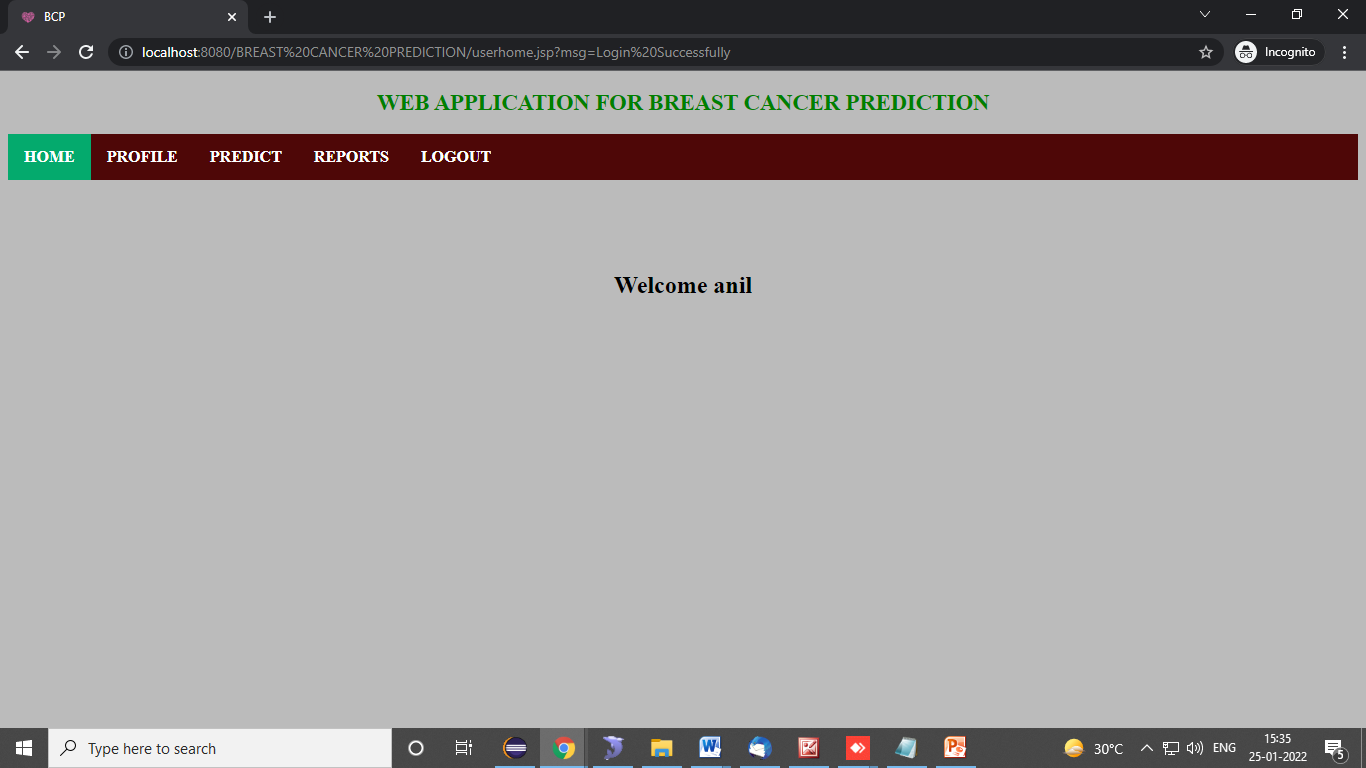
Upload Data: admin here upload breast cancer dataset

Upload Data: If we have already data uploaded this will ask to confirm to delete old data and upload new data

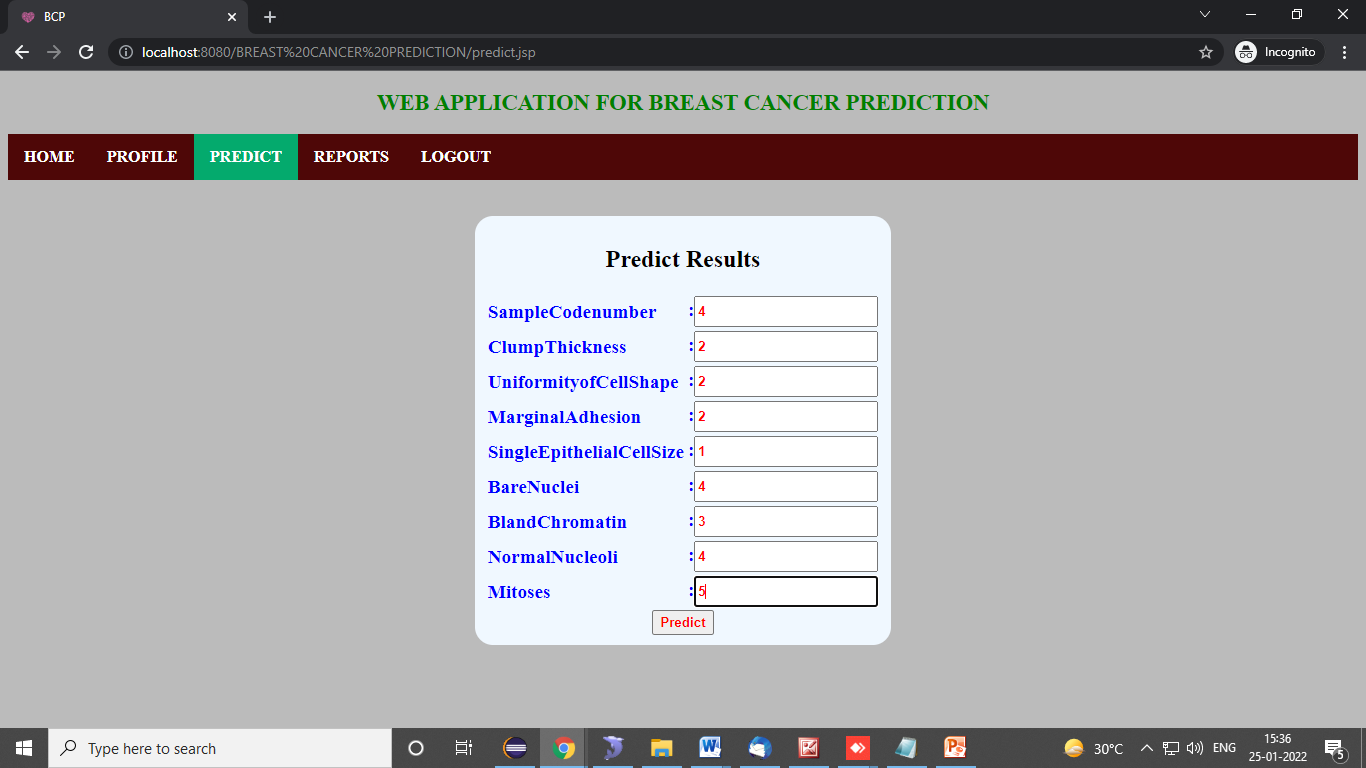
View data: here admin view uploaded data details

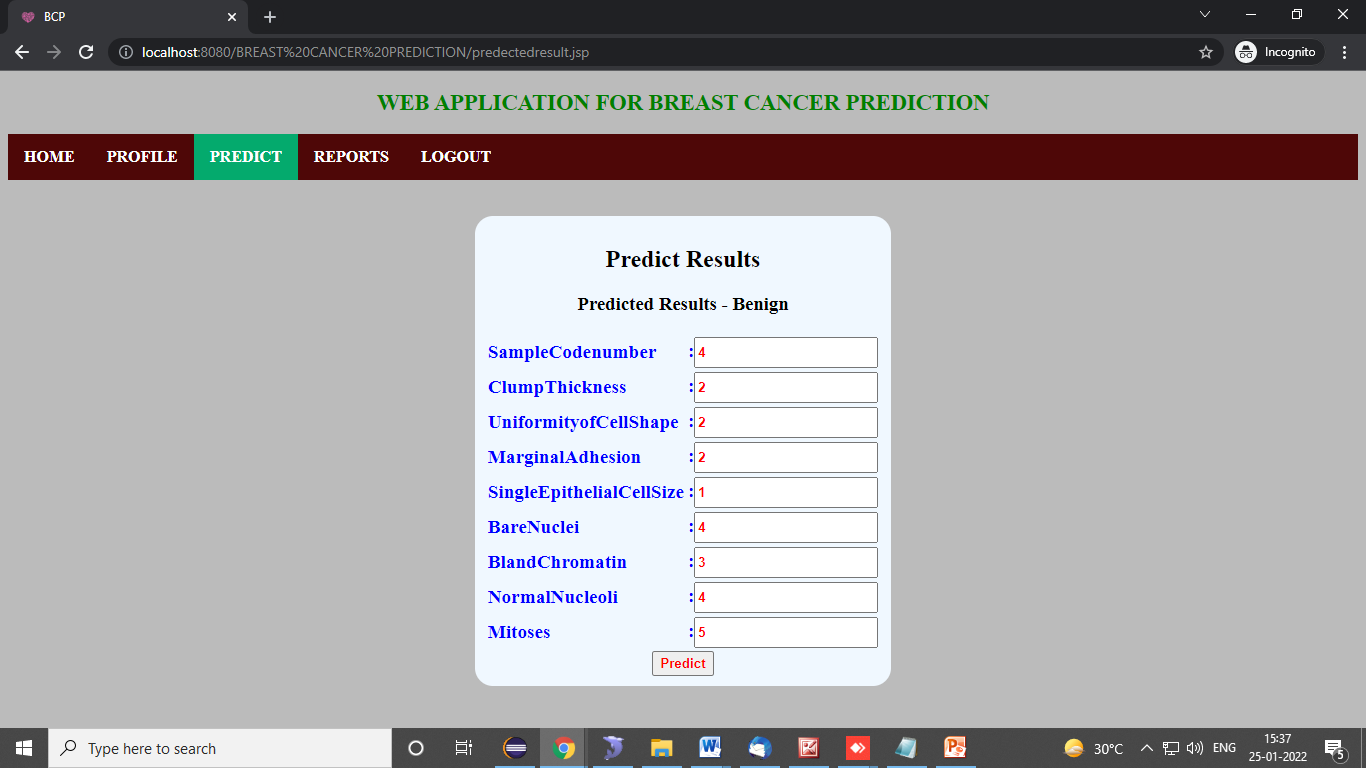
User register: user register with their details

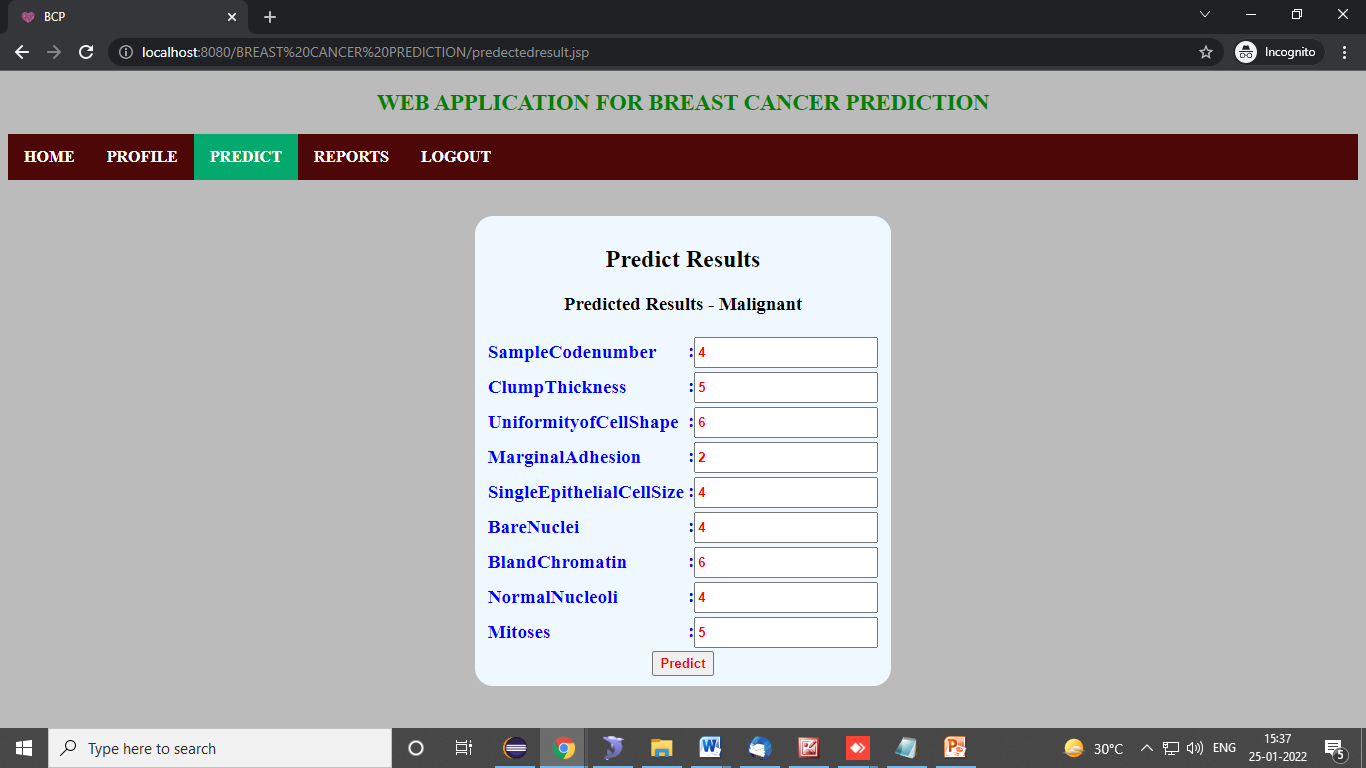
User login : user login with email and password

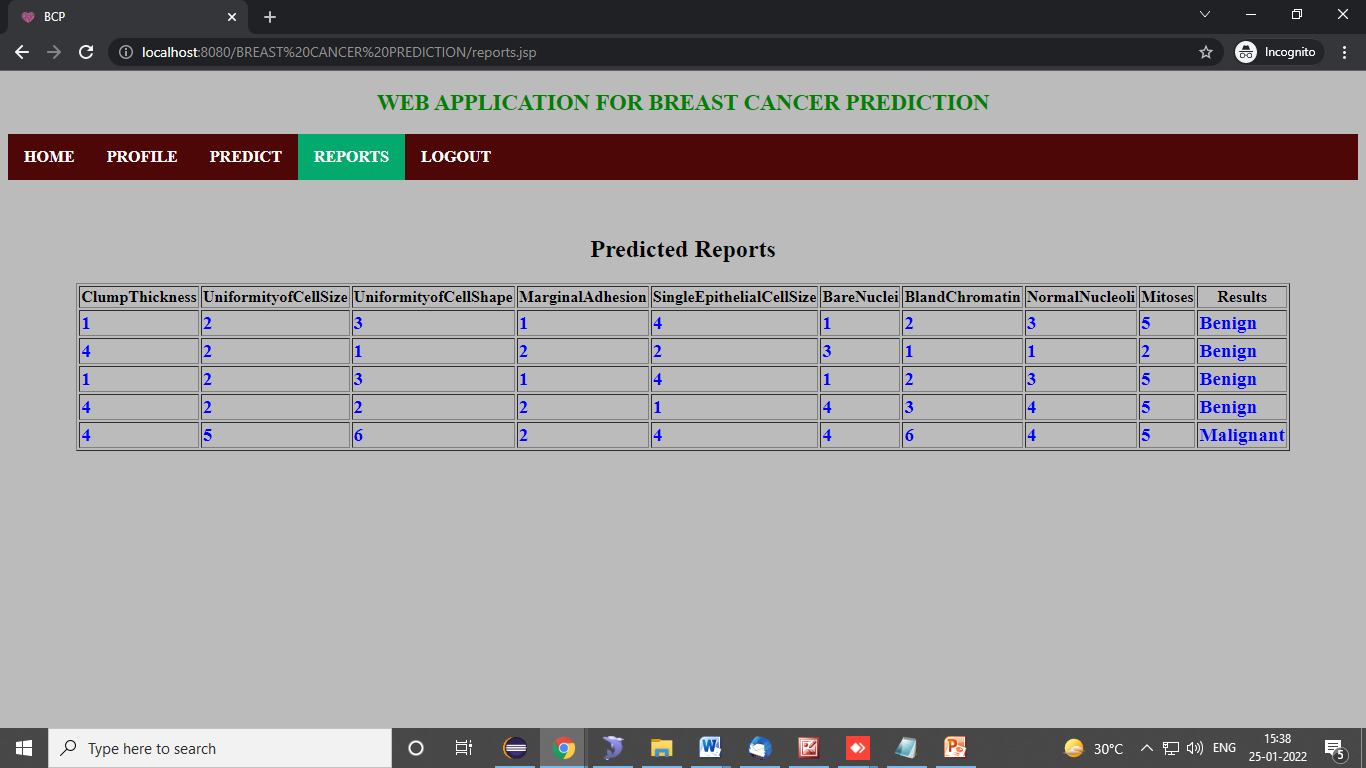
User home : this is user home page. In this we can view all available user features. 

profile : this is user profile

predict: user predict the breast caner

Prediction results1

Prediction results2

**Report: user predicted results**

**TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | Admin login | Valid/invalid credentials | Login success/ login failed | Login success/ login failed | P |
| 2 | User register | User details | If email not exist registration done else email already exists | If email not exist registration done else email already exists | P |
| 3 | User login | Valid/invalid credentials | Login success/ login failed | Login success/ login failed | P |
| 4 | Upload File | File (CSV file) | Upload success/data already exists do you want delete existing data – if yes- data will delete and upload else view data | Data upload into database | P |
| 5 | Prediction | Prediction input parameters | Prediction results | Prediction results | P |

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

**CONCLUSION**

In our application we are predicting the breast cancer. By using the existing records i.e., database results. In this user can predict results, and check their existing predicted results. And admin can add new records into database. This can be helpful to next predicting results.

**FUTURE SCOPE**

In future we can implement to add precautions and suggestions to users.

**REFERENCES**

1. Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis from Elselver & 2021
2. Automated Breast Cancer Detection and Classification Techniques – A survey from IEEE & 2021
3. Breast Cancer Prediction: A Comparative Study Using Machine Learning Techniques from Springer & 2020
4. Breast cancer detection by leveraging Machine Learning from Science direct & 2020

**BIBLIOGRAPHY:**

**SOFTWARE ENVIRONMENT AND INSTALLATION**

**Software Environment**

## 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. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.   
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 access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

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:

**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.

1. **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.

1. **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.
2. **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.

1. **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.

1. **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.

1. **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
* Architecture-neutral
* Object-oriented
* Portable
* Distributed
* High-performance
* Interpreted
* Multithreaded
* Robust
* 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.

**Java Program**

**Compilers**

**Interpreter**

**My Program**

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](http://www.gnu.org/philosophy/free-sw.html). It is distributed under the terms of the [GNU Lesser General Public Licence](http://www.gnu.org/licenses/lgpl.html) (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.

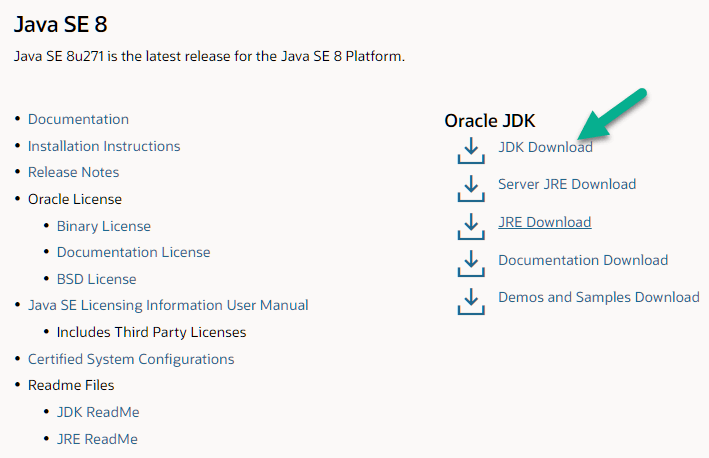
**SOFTWARE INSTALLATION FOR JAVA PROJECTS**

This Java Development Kit(JDK) allows you to code and run Java programs. It's possible that you install multiple JDK versions on the same PC. But Its recommended that you install only latest version.

## How to install Java for Windows

Following are the steps for JDK 8 free download for 32 bit or JDK 8 download 64 bit and installation

**Step 1)** Go to [link](https://www.oracle.com/java/technologies/javase-downloads.html). Click on JDK Download for Java



**Step 2)** Next,

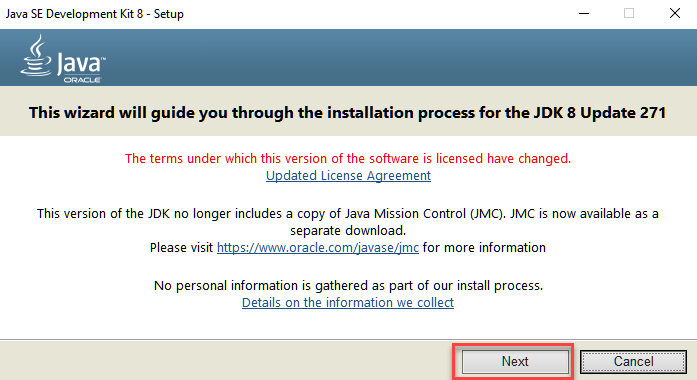
1. Accept License Agreement
2. Download Java 8 JDK for your version 32 bit or JDK 8 download for windows 10 64 bit.



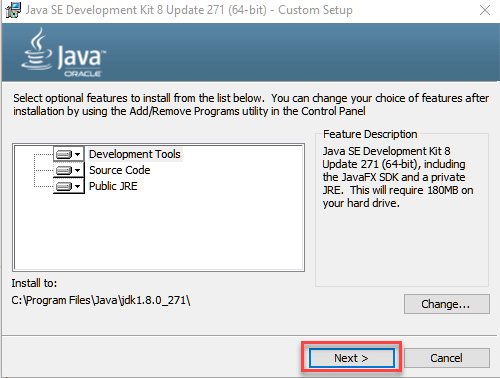
**Step 3)** when you click on the Installation link the popup will be open. Click on I reviewed and accept the Oracle Technology Network License Agreement for Oracle Java SE and you will be redirected to the login page. If you don't have an oracle account you can easily sign up by adding basics details of yours.



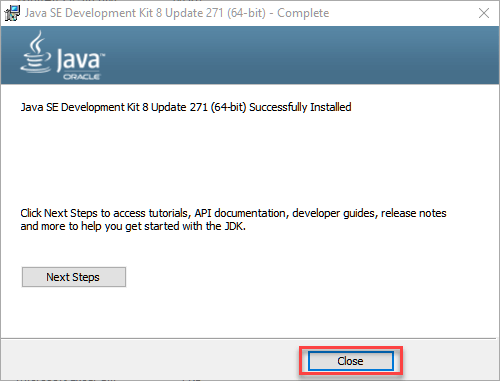
**Step 4)** Once the Java JDK 8 download is complete, run the exe for install JDK. Click Next



**Step 5)** Select the PATH to install Java in Windows and click next.



**Step 6)** Once you install Java in windows, click Close



## How to set Environment Variables in Java: Path and Class path

The PATH variable gives the location of executable like javac, java etc. It is possible to run a program without specifying the PATH but you will need to give full path of executable like **C:\Program Files\Java\jdk-13.0.1\bin\javac A.java** instead of simple **javac A.java**

The CLASSPATH variable gives location of the Library Files.

Let's look into the steps to set the PATH and CLASSPATH

**Step 1)** Right Click on the My Computer and Select the properties

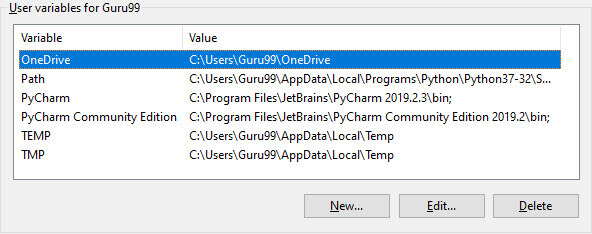


**Step 2)** Click on advanced system settings

**Step 3)** Click on Environment Variables



**Step 4)** Click on new Button of User variables



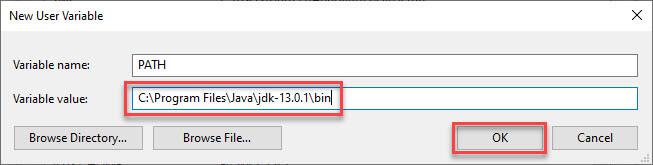
**Step 5)** Type PATH in the Variable name.



**Step 6)** Copy the path of bin folder which is installed in JDK folder.



**Step 7)** Paste Path of bin folder in Variable value and click on OK Button.



**Note:** In case you already have a PATH variable created in your PC, edit the PATH variable to

PATH = <JDK installation directory>\bin;%PATH%;

Here, %PATH% appends the existing path variable to our new value

**Step 8)**You can follow a similar process to set CLASSPATH.



**Note:** In case you java installation does not work after installation, change classpath to

CLASSPATH = <JDK installation directory>\lib\tools.jar;

**Step 9)** Click on OK button



**Step 10)** Go to command prompt and type javac commands.

If you see a screen like below, Java is installed.



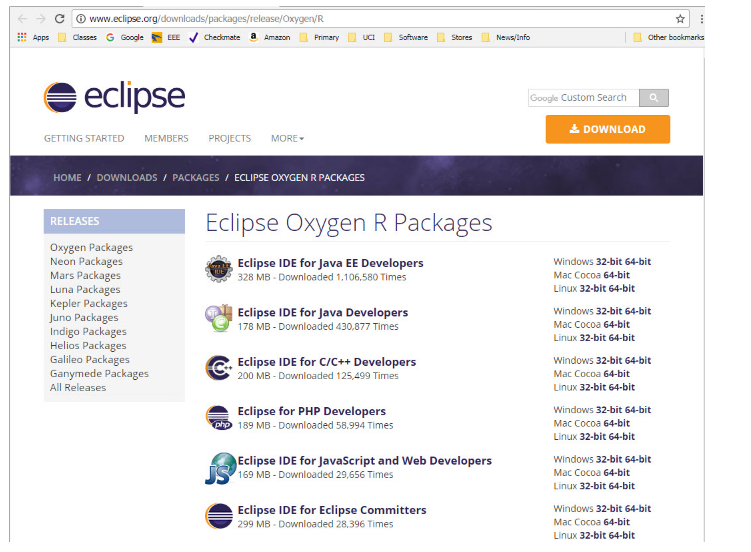
## Eclipse: (Oxygen)

The Eclipse download requires about 300 MB of disk space; keep it on your machine, in case you need to re-install Eclipse. When installed, Eclipse requires an additional 330 MB of disk space.

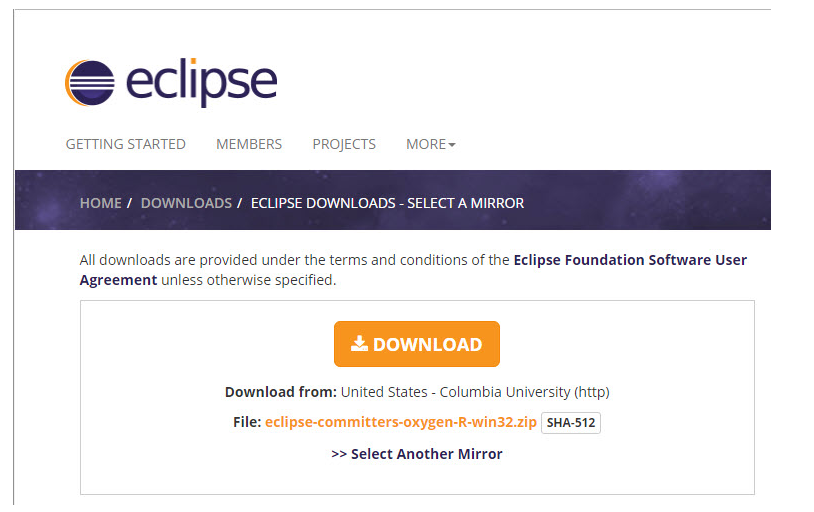
### Downloading

1.Click [Eclipse](http://www.eclipse.org/downloads/packages)

The following page will appear in your browser. In this handout we will download **Eclipse IDE for Eclipse Committers** for Windows 32 Bit; if your computer uses Windows, continue below; otherwise choose either **Mac Cocoa** or **Linux** instead.



1. It is critical that Java, Python, and Eclipse are either all 32 Bit or are all 64 Bit (and only if your Machine/OS supports 64 Bit): I think it easiest to use 32 Bit for everything.
2. Click the **32-Bit** (after Windows) to the right of the **Eclipse IDE for Eclipse Committers**.
3. You will see the following page (don't worry about the name of the institution underneath the orange **DOWNLOAD** button).



Click the orange **DOWNLOAD** button. The site named here, in orange to the right of the button: **United States - Columbia University (http)** is the random one chosen by the download page this time; yours may differ.

This file should start downloading in your standard download folder, while showing a splash screen about donating to Eclipse. This file is about 300 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 5 minutes over a cable modem). Don't worry about the exact time as long as the download continues to make steady progress. In Chrome progress is shown on the bottom-left of the window, via the icon

https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/images/eclipseoxygen/downloadprogress.jpg

The file should appear as

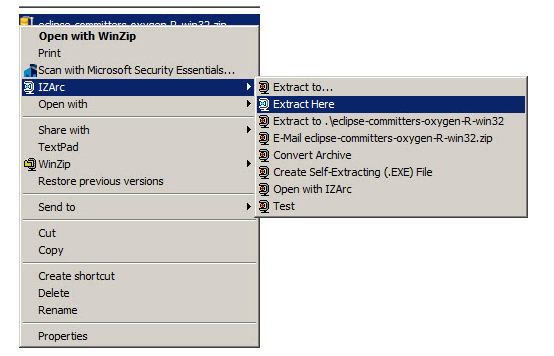
https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/images/eclipseoxygen/downloadfile.jpg

Terminate the tab browsing this webpage.

1. Move this file to a more permanent location, so that you can install Eclipse (and reinstall it later, if necessary).
2. Start the **Installing** instructions directly below.

Unzip **eclipse-committers-oxygen-R-win32.zip**, the file that you just downloaded and moved.  
On my machine (running Windows 7), I can

* Right-click the file.
* Hover over the **IZArc** command from the menu of options.
* Click **Extract Here**



1. If you do not have IZArc or an equivalent unzipping program, here is the web site to download a free copy of [IZarc](http://izarc.org/downloads" \t "_blank).

Unzipping this file creates a folder named **eclipse**; unzipping 250 MB can take a few minutes. You can leave this folder here or move it elsewhere on your hard disk. I recommend putting the downloaded file and resulting folder in the **C:\Program Files\** directory.

1. Create a shortcut on your desktop to the **eclipse.exe** file in this **eclipse** folder:   
   On most Windows machines, you can
   * Right-press the file **eclipse.exe**
   * Drag it to the desktop.
   * Release the right button.
   * Click **Create shortcut here**

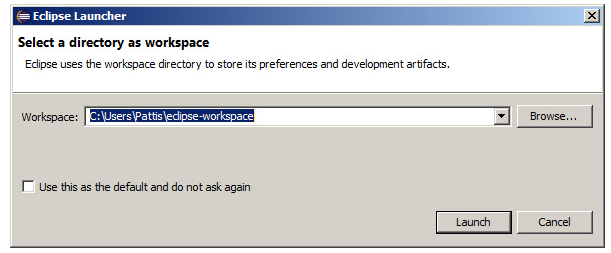
Now you are ready to perform a **one-time only** setup of Eclipse on your machine.

1. Double-click the shortcut to Eclipse that you just created above.

The following splash screen will appear



and then an **Eclipse Launcher** pop-up window will appear.



1. In the **Workspace** text box, your name should appear between **C:\Users\** and **\eclipse-workspace**, instead of **Pattis**.

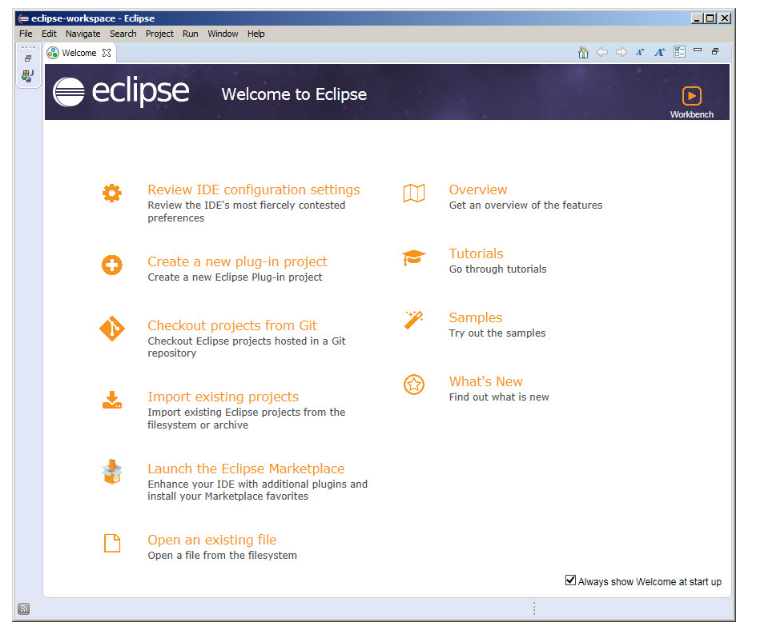
Leave **unchecked** the **Use this as the default and do not ask again** box. Although you will use this same workspace for the entire quarter (checking projects in and out of it), it is best to see this **Workspace Launcher** pop-up window each time you start Eclipse, to remind you where your workspace is located.

In fact, it is a good idea to create on your desktop a shortcut to your workspace folder; but you must click **OK** (see below) before Eclipse creates this folder and you can create a shortcut to it.

1. Click **Launch**.

Progress bars will appear as Eclipse loads.

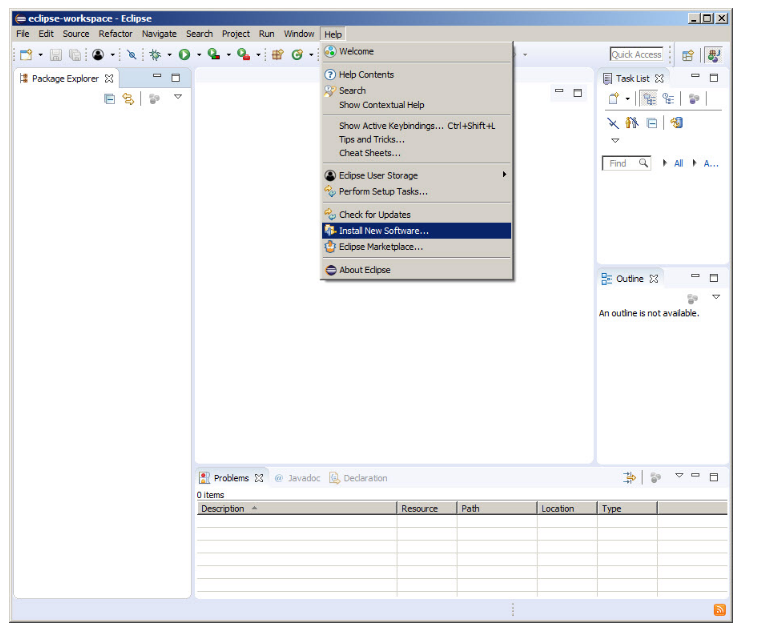
Eventually the Eclipse workbench will appear with a **Welcome** tab covering it.



1. Terminate (click **X** on) the **Welcome** tab.

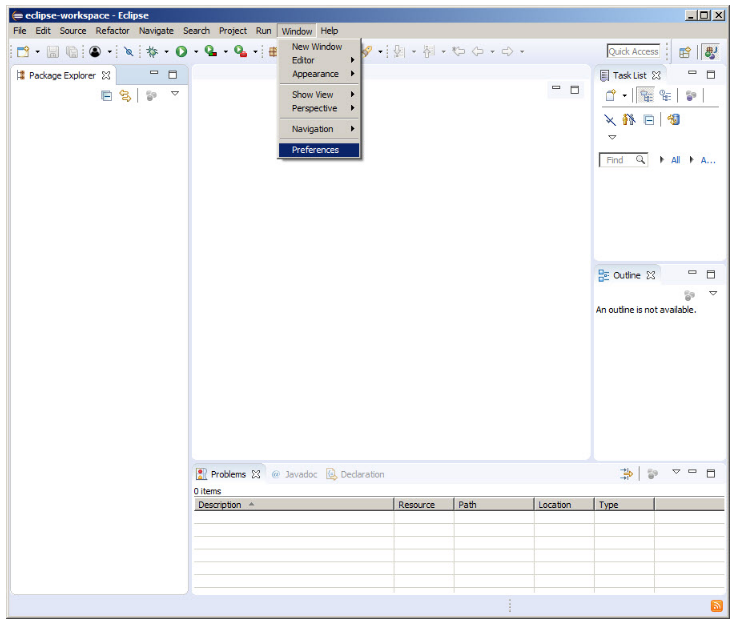
You will not see the **Welcome** tab when you start Eclipse again, after this first time.

1. Click **Help** (on the far right of the line below this window's blue title **eclipse-workspace - Eclipse**) and then click **Install New Software...** in its pull-down menu, as shown below.

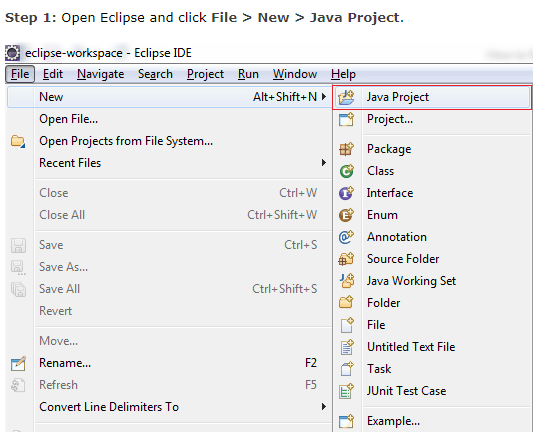


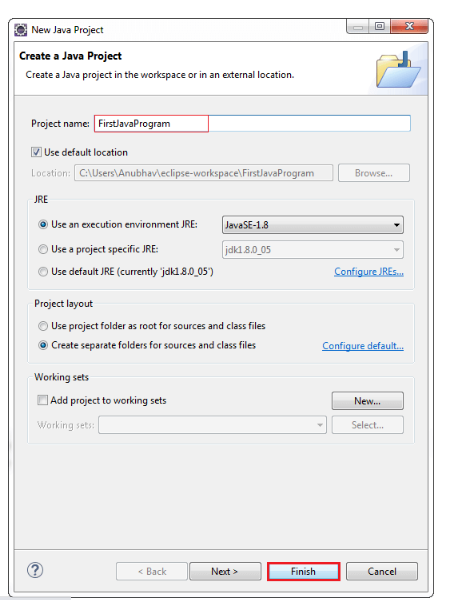
The Install pop-up window will appear.

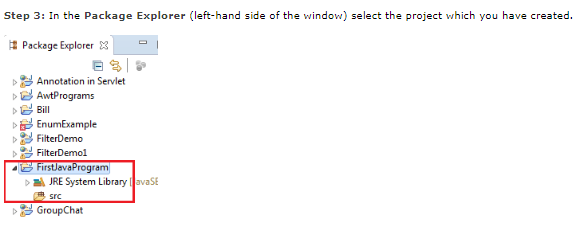
Click Window (to the left of Help on the far right of the line below this window's blue title eclipse-workspace - Eclipse) and then click Preferences in its pull-down menu, as shown below.

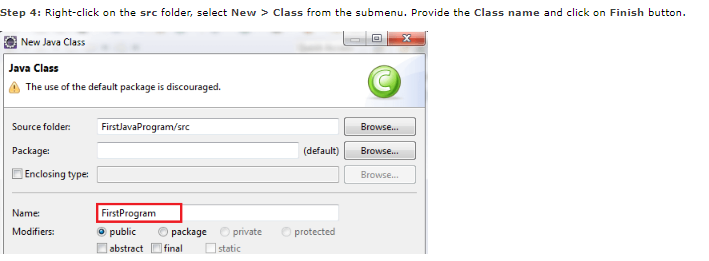


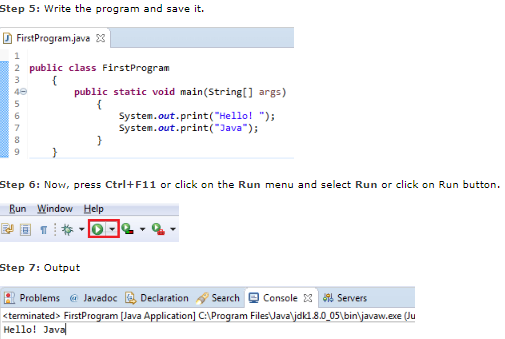
* Create a project.
* Right-click on your project and choose **Properties**.
* Click on **Java Build Path**.
* Click on the **Libraries** tab.
* Click on **Add External JARs...**.
* Navigate to **junit.jar**. It should be in a location such as **...Eclipse 3.0.1\plugins\org.junit\_3.8.1\junit.jar**.
* Select **junit.jar**, click on **Open**, click on **OK**.
* To create a test class:
* Open the New wizard (**File > New > Other...**)
* Select **Java > JUnit** in the left pane and **TestCase** in the right pane and click **Next**.
* Enter the name of your test class and click **OK**.
* To run your test class, select your test class and choose **Run as > JUnit Test** from the **Run** drop-down menu in the toolbar.











**STEPS FOR EXECUTING THE PROJECTS**

**Step 1:**

Open Eclipse and set the workspace

**Step2:**

Right Click on the project Run As and Run On Tomcat Server

**Step3:**

In middle we got tomcat error that time we need to change port number

**Step4:**

Copy url in Google chrome and Run