**SOFTWARE TRAINING REPORT**

ON

**Advanced JAVA**

Submitted in partial fulfillment of the

requirements for the award of

**Degree of Bachelor of Technology in Computer Science & Engineering**



**Submitted To: Submitted By:**

**Mrs. SUKHDEEP KAUR RITIK GUPTA**

**Assistant professor 1702815**

**Department of Computer Science & Engineering**

**CHANDIGARH ENGINEERING COLLEGE, LANDRAN**



**DECLARATION**

I hereby declare that the Software Training Report entitled **Chatting Application** is an authentic record of my own work as requirements of 6 - months Software Training during the period from 01-02-2021 to 21-07-2021 for the award of degree of B. Tech. (Computer Science & Engineering, **CHANDIGARH ENGINEERING COLLEGE** under the guidance of Mr. Ravi Thakur.

**Ritik Gupta**

**1702815**

**Date: 1st July, 2021**

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

**Signatures**

**Examined by:**

1. 2. 3. 4.

**Head of Department**

**(Signature and Seal)**

**ACKNOWLEDGEMENT**

We owe our insightful appreciation and gratitude to our project supervisor **(HOD Mrs. Sukhpreet Kaur)**, who took keen interest and guided us all along in the project work, titled “**Chatting Application”**, till the completion of the project byproviding all the necessary information for developing the project. The project development helped us in research and we got to know a lot of new things in the domain of Data Analysis. We are in truth grateful to him.

**Ritik Gupta**

**1702815**

**About Company: SoftEdutech**

**SoftEdutech** is an India based leading strategic IT Company offering integrated IT solutions with the vision to provide excellence in software solution. We at SoftEdutech bring innovative ideas and cutting edge technologies into business of customers. SoftEdutech is having a rich experience in providing high technology end to end solutions in **MOBILE APP AND WEB DEVELOPMENT.**

The company is managing global clients across various business verticals and align IT strategies to achieve business goals. The various accreditations that we achieved for every service, we offer reflect our commitment towards the quality assurance. **SoftEdutech** has won the **NATIONAL AWARD** for**2015-16** for the highly appreciable contribution in the field of computersfrom **Hon’able Education Minister Of Punjab Dr.Daljit Singh Cheema.**

We feel pleasure to acquaint with our company as a pioneer in the field of Software Development & Industrial Internship. Our Company involves in **Software Development**, **Web Designing, Industrial Training, Corporate Training, Consulting and Promoting outstanding products** of various companies. We have substantial experience in **Analyzing, Designing, Developing, Testing, Implementing and Supporting projects**. We have a global foot prints in providing the EXCELLENCE TECHNOLOGY companies of **US, UK, France, Ireland, Canada and Australia with quality and timely Web Development and SEO services**.

The foundation upon which our team is created is based upon the premise that motivated people and long-standing relationships are the ultimate tools of success and creativity, energy, perseverance and loyalty are just as important as a platinum resume.

We have a team of highly qualified and experienced professionals with proven problem solving, consulting and analytical skills. Our Team consists of Domain Experts & Technical Specialists to provide world class Business development solutions to global clients.

**Philosophy**

* To impart hardcore practical quality training among students/developers about latest technologies trending today.
* To share knowledge of information security and create awareness in the market. The solution to clients' as per the International standard practices and governance.
* To support good business practices through continual employee training and education
* To equip a local team with a strong knowledge of international best practices and international expert support so as to provide practical advisories in the best interests of our clients

**Vision**

To be a Leading information risk assessors by protecting the client organization's information system from threats and vulnerabilities.

To achieve the development objective to the best established practices and recognized standards worldwide and Regulatory as well as statutory obligation of the region in which organization is operating.

**Why SoftEdutech?**

* Hundreds of Clients & Nearly a Decade of Experience
* Goal Oriented, ROI-Driven Focus
* A Streamlined / Quality-Driven Process
* Talented Designers & Expert Developers
* Our Websites & E-marketing Platforms are Easy to Manage
* We Are Dedicated to Our Clients’ Success
* **EXCELLENCE TECHNOLOGY** is steadfast to undertake the projects cutting edge to technology competence and know-how abilities. The project execution is held with dedication and responsibility to perform our best with the essence of knowledge, creativity and skills to the utmost and efficiently.
* At **EXCELLENCE TECHNOLOGY,** we have competence to expand and adjust as per client specific requirements.
* **Skilled Workforce:** At EXCELLENCE TECHNOLOGY you deal with the highly professional and proficient employees.
* **Cost Efficiency:** We help you to reduce the unnecessary investment and ask for the reasonable amount of money.
* **Quality Of the Product:** Our software service sector has been maintaining the highest international standards of quality.
* **Infrastructure:** Well organized team and tools to handle the projects with responsible approach Hardware, Software, Networking, Voice, Conferencing, disaster recovery all infra all you need for international projects.
* **Ongoing Involvement:** EXCELLENCE TECHNOLOGY products are “built for change” as we are well responsive that the necessity to improve a Web solution generally arises even before the solution is out of the door. We deliver long-term product enhancement if desired.
* **Partnership:** EXCELLENCE TECHNOLOGY considers every client a partner. From the initial stages, you are closely involved into the procedure of technical classification, development, and testing.

Table of Contents

1. Introduction
2. Software and Hardware Requirements
3. **Advanced JAVA**
   * 1. **CHAPTER 1: INTRODUCTION**
     2. **CHAPTER 2: COLLECTION**
     3. **CHAPTER 3: MULTITHREADING**
     4. **CHAPTER 4: JAVA NETWORKING**
     5. **CHAPTER 5: AWT**
     6. **CHAPTER 6: SWING**
     7. **CHAPTER 7: JAVA DATABASE CONNECTIVITY**
     8. **CHAPTER 8: JAVA SERVER PAGES**
     9. **CHAPTER 9: JAVA SERVER PAGES**
     10. **CHAPTER 10: APPLET**
4. **Project Snapshots**
5. **Conclusion**
6. **References**

# Introduction

The main objective of this system is develop flawless system, which is able to connect two people connected to the same network 24 hours a day 365 days in a year.  Another aim is that manage hundred of projects in multiple locations or just a few. The another main objective of this system is to keep the communication simple and flawless.

Teleconferencing or Chatting, is a method of using technology to bring people and ideas “together” despite of the geographical barriers. The technology has been available for years but the acceptance it was quite recent. On the Internet, chatting is talking to other people who are using the Internet at the same time you are.

In some cases, a private chat can be arranged between two parties who meet initially in a group chat. Our project is an example of a chat server. It is made up of two applications the client application, which runs on the user’s computer and server application, which runs on any computer on the network. To start chatting client should get connected to server where they can practice two kinds of chatting, public one (message is broadcasted to all connected users) and private one (between any 2 users only) and during the last one security measures were taken.

**Software and Hardware Requirements**

## 2.1 Software Requirements

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system which enables the user to interact with several physical hardware devices.

The minimum software requirement specifications for developing this project are as follows:

Designing frontend : JSP and Servlets

Backend : My SQL

Scripting : Java Script

IDE : My Eclipse

Web Server : Tomcat

## 2.2 Hardware Requirement Specification

The Collection of internal electronic circuits and external physical devices used in building a computer is called Hardware.

The minimum hardware requirement specification for developing this project is as follows:

Processor : Pentium IV

RAM : 512MB RAM

Hard Disk : 10GB

**ADVANCED JAVA**

**Chapter 1: Introduction**

**1. INTRODUCTION TO JAVA**

Java is a programming language initially developed by Sun Microsystems by James Gosling and released as a principal component of Sun Microsystems' Java platform in 1995. Although the language gets much of its syntax from C and C++ it has a less complicated object model and lesser low-level services. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture .Java is a general-purpose, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most standard programming languages in use, and is extensively used from application software to web applications.

There were five primary goals in the creation of the Java language.

* + It should be "simple, object-oriented and familiar".
  + It should be "robust and secure".
  + It should be "architecture-neutral and portable".
  + It should execute with "high performance".
  + It should be "interpreted, threaded, and dynamic".

1. **J2EE**

Short for Java 2 Platform Enterprise Edition, J2EE is a platform-independent, Java- centric environment from Sun for developing building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitier, Web-based applications.

Some of the key features and services of J2EE

At the client tier, J2EE supports pure HTML, as well as Java applets or applications. It relies on Java Server Pages and servlet code to create HTML or other formatted data for the client

Enterprise JavaBeans (EJBs) provide another layer where the platform's logic is stored. An EJB server provides functions such as threading, concurrency, security and memory management. These services are transparent to the author.

Java Database Connectivity (JDBC), which is the Java equivalent to ODBC, is the standard interface for Java databases.

The Java servlet API enhances consistency for developers without requiring a graphical user interface.

**CHAPTER 2:COLLECTION**

Collection framework provides a well-designed set of interface and classes for storing and manipulating groups of data as a signal unit a collection. It provides a standard programming interface to many of the most common abstractions, without blundering the programmer with too many procedure and interfaces.

**Collocation tree part**

* **List-**Array List

**1.2. Set-** Tree set, Hash set, Link hash set

**1.3. Map**- Hash Map, Hash Table, Tree Map

**2. Benefits of the Java Collections Framework**

**2.1 Reduces programming effort**

By providing useful data structures and algorithms, the Collections Framework frees you to concentrate on the important parts of your program rather than on the low-level "plumbing" required to make it work. By facilitating interoperability among unrelated APIs, the Java Collections Framework frees you from writing adapter objects or conversion code to connect APIs.

**2.2 Increases program speed and quality**

This Collections Framework provides high-performance, high-quality implementations of useful data structures and algorithms. The various implementations of each interface are interchangeable, so programs can be easily tuned by switching collection implementations. Because you're freed from the drudgery of writing your own data structures, you'll have more time to devote to improving programs' quality and performance.

**2.3. Allows interoperability among unrelated APIs**

The collection interfaces are the vernacular by which APIs pass collections back and forth. If my network administration API furnishes a collection of node names and if your GUI toolkit expects a collection of column headings, our APIs will interoperate seamlessly, even though they were written independently.

**2.4. Reduces effort to learn and to use new APIs**

Many APIs naturally take collections on input and furnish them as output. In the past, each such API had a small sub-API devoted to manipulating its collections. There was little consistency among these ad hoc collections sub-APIs, so you had to learn each one from scratch, and it was easy to make mistakes when using them. With the advent of standard collection interfaces, the problem went away.

**2.5. Reduces effort to design new APIs**

This is the flip side of the previous advantage. Designers and implementers don't have to reinvent the wheel each time they create an API that relies on collections; instead, they can use standard collection interfaces.

**2.6. Fosters software reuse**

New data structures that conform to the standard collection interfaces are by nature reusable. The same goes for new algorithms that operate on objects that implement these interfaces

**CHAPTER 3: MULTITHREADING**

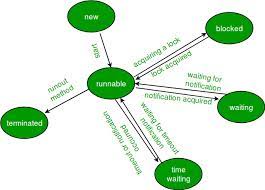
Java is a multi-threaded programming language which means we can develop multithreaded program using Java. A multi-threaded program contains two or more parts that can run concurrently and each part can handle a different task at the same time making optimal use of the available resources especially when your computer has multiple CPUs.

By definition, multitasking is when multiple processes share common processing resources such as a CPU. Multi-threading extends the idea of multitasking into applications where you can subdivide specific operations within a single application into individual threads. Each of the threads can run in parallel. The OS divides processing time not only among different applications, but also among each thread within an application.

Multithreading enables you to write in a way where multiple activities can proceed concurrently in the same program.

**1. Life Cycle of a Thread**

A thread goes through various stages in its life cycle. For example, a thread is born, started, runs, and then dies. The following diagram shows the complete life cycle of a thread.



Following are the stages of the life cycle −

**New** − A new thread begins its life cycle in the new state. Itremains in this state until theprogram starts the thread. It is also referred to as a **born thread**.

**Runnable** − after a newly born thread is started, the threadbecomes runnable. A thread in thisstate is considered to be executing its task.

**Waiting** − Sometimes, a thread transitions to the waiting state while the thread waits for anotherthread to perform a task. A thread transitions back to the runnable state only when another thread signals the waiting thread to continue executing.

**Timed Waiting** − A runnable thread can enter the timed waiting state for a specified interval oftime. A thread in these state transitions back to the runnable state when that time interval expires or when the event it is waiting for occurs.

**Terminated (Dead)** − Arunnable thread enters the terminated state when it completes its task orotherwise terminates. Thread Priorities Every Java thread has a priority that helps the operating system determine the order in which threads are scheduled.

Java thread priorities are in the range between MIN\_PRIORITY (a constant of 1) and MAX\_PRIORITY (a constant of 10). By default, every thread is given priority NORM\_PRIORITY (a constant of 5).

Threads with higher priority are more important to a program and should be allocated processor time before lower-priority threads. However, thread priorities cannot guarantee the order in which threads execute and are very much platform dependent. Create a Thread by Implementing a Runnable Interface If your class is intended to be executed as a thread then you can achieve this by implementing a **Runnable** interface.

You will need to follow three basic steps −

**Step 1-** As a first step, you need to implement a run () method provided by a **Runnable** interface. This method provides an entry point for the thread and you will put your complete business logic inside this method.

Following is a simple syntax of the run () method − public void run ( )

**Step 2-** As a second step, you will instantiate a **Thread** object usingthe following constructor −Thread (Runnable threadObj, String threadName);

**Step 3-** Once a Thread object is created, you can start it by calling **start ()** method, whichexecutes a call to run ( ) method. Following is a simple syntax of start ()

Method − void start ();

**CHAPTER 4: JAVA NETWORKING**

The term Network programming refers to writing programs that execute across multiple devices (Computers), in which the devices are all connected to each other using networks. The Java net package of the J2se API contains a collection of classes and interfaces that provide the low-level communication details, allowing you to write programs that focus on solving problem at hand.

The java net package provides support for the two common network protocols.

1. The java.net package provides support for the two common network protocols

**TCP-** TCP stands for transmission control protocol, which allows for reliable communicationbetween two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.

**UDP (User Datagram Protocol) -** UDP stands for User Datagram Protocol, a connection-lessprotocol that allows for packets of data to be transmitters between applications.

1. This chapter gives a good understanding on the following two subjects −

**URL Processing-** This would be covered separately. Click here to learn about URL Processingin java language.

**Socket Programming-** Sockets provide the communication mechanism between two computersusing TCP. A client program creates a socket on its end of the communication and attempts to connect that socket to a server. When the connection is made, the server creates a socket object on its end of the communication.

The client and the server can now communicate by writing to and reading from the socket.

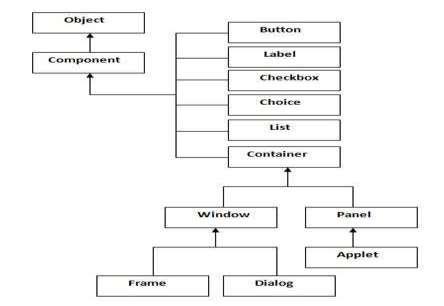
**CHAPTER 5: AWT (ABSTRACT WINDOWING TOOLKIT)**

Java AWT (Abstract Windowing Toolkit) is an API to develop GUI or window-based application in java.

* AWT components are platform-dependent.
* AWT components are heavyweight.
* AWT doesn't support pluggable look and feel.
* AWT provides fewer components than Swing.
* AWT doesn't follow MVC (Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view.

**1. Java AWT Hierarchy**

The hierarchy of Java AWT classes is given below.



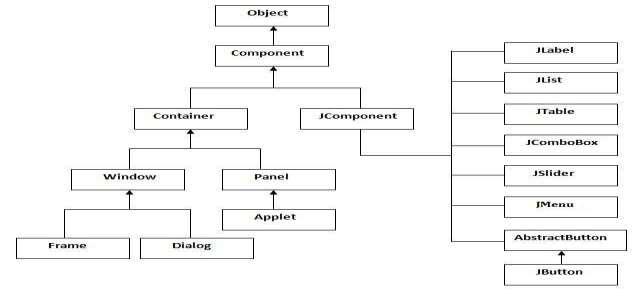
**CHAPTER 6: SWING**

Work Flow **Java Swing** is a part of Java Foundation Classes (JFC) that is **used to create** **window-based applications**. It is built on the top of AWT (Abstract Windowing Toolkit) APIand entirely written in java.

* Java swing components are **platform-independent**.
* Swing components are **lightweight**.
* Swing **supports pluggable look and feel**.
* Swing provides **more powerful components** such as tables, lists, scroll panes, color chooser, tabbed pane etc.
* Swing **follows MVC**.

**1. Hierarchy of Java Swing classes**

The hierarchy of java swing API is given below:



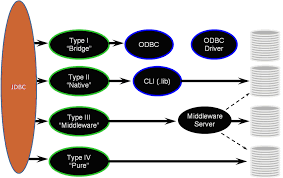
**CHAPTER 7: JAVA DATABASE CONNECTIVITY**

The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases – SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access.

JDBC technology allows you to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment.

1. **JDBC drivers**

There are commercial and free drivers available for most relational database servers. These drivers fall into one of the following types:



**Type 1** That calls native code of the locally available ODBC driver.

**Type 2** That calls database vendor native library on a client side. This code then talks to databaseover network.

**Type 3**, the pure-java driver that talks with the server-side middleware those then talks todatabase.

**Type 4**, the pure-java driver that uses database native protocol

**2. Functionality and implementation**

JDBC allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections. JDBC connections support Creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes-

**Statement -** the statement is sent to the database server each and every time.

**Prepared Statement -** the statement is cached and then the execution path is predetermined onthe database server allowing it to be executed multiple times in efficient manner.

**Callable Statement-** used for executing stored procedure son the

Database. Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database.

These statements do not return any other information. Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types. There is an extension to the basic JDBC API in the javax.sql. The method Class.forName (String) is used to load the JDBC driver class.

**Example-** Class.forName (sun.jdbc.odbc.JdbcOdbcDriver); Used to load the JDBC-ODBCbridge driver.

When a Driver class is loaded, it creates an instance of itself and registers it with the Driver Manager. Now when a connection is needed, one of the DriverManager.getConnection() methods is used to create a JDBC connection

If a database operation fails, JDBC raises a SQLException. There is typically very little one can do to recover from such an error, apart from logging it with as much detail as possible. It is recommended that the SQLException be translated into an application domain exception (an unchecked one) that eventually results in a transaction rollback and a notification to the user.

Finally to summarize, the following points can be stated-

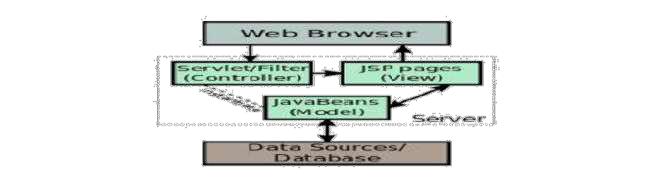
* JDBC API provides a database programming interface for Java programs. A Java program can send queries to a database by using the JDBC driver.
* The java.sql package contains classes that help in connecting to a database, sending SQL statements to the database and processing the query results.
* The Connection Object represents a connection with a database. I t can be initialized using the get Connection () method of the Driver Manager class.
* The Prepared Statement object allows you to execute parameterized queries. It can be initialized using the prepare Statement () method of the Connection object.
* The set String () method sets the query parameters of the PreparedStatementobject.
* The execute Update () method executes the query statement present in the Prepared Statement object and returns the number of rows affected by the query.
* The Result Set Metadata interface is used to obtain information about the columns stored in a Result Set object.

**CHAPTER 8: JAVA SERVER PAGES**

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems [1], JSP is similar to PHP, but it uses the Java programming language. To deploy and run Java Server Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Java Server Pages is a technology which permits software developers to create dynamic request like HTML, XML in order to answer to client request in the net. This technology lets Java code and definite pre-defined procedures to be implanted into static content. The syntax in Java Server Pages includes a supplementary XML tag which is known as JSP actions**.**

It is made use to evoke the utility of the built-in functions. Moreover JSP permits to establish and form the JSP tag libraries which operate as an extension to the standard XML or HTML tags. These JSP tag libraries give a good technique to widen the potentiality of the Web server by providing an independent platform’s compiler compiles the JSPs into Java Servlets



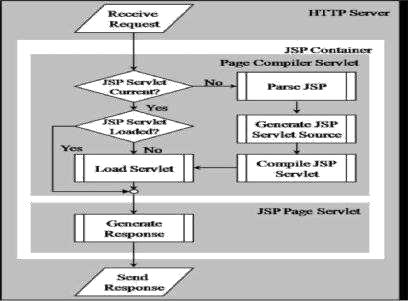
A JSP compiler may possibly create a servlet in Java code and it is later compiled by the Java compiler. It might even directly produce the byte code for the servlet. Java Server Pages can be examined as a high level abstraction of servlets which is practiced as an extension of the Servlet2.1API.

Java Server Pages are HTML pages embedded with snippets of Java code. It is an inverse of a Java Servlet

**JSPs run in two phases**

**Translation Phase-** In translation phase JSP page is compiled into a servlet called JSP PageImplementation class

**Execution Phase-** In execution phase the compiled JSP is processed

****

The Java Server Pages and the Servlets were initially developed at Sun Microsystems. Opening with version 1.2 of the Java Server Page specification the JSPs have been built under the Java Community Process. There are quite a few JSP implicit objects that are represented by the JSP container and it could be mentioned and indicated by the programmers

**Config -** It gives the data of the servlet configuration

**Application -** Data's are shared by the servlets and Java Server Pages in theapplication.

**Exception -** Exceptions are not trapped by the codes in the application.

**Out -** The data's are written with the help of JSP Writer to the response stream.

**Request -** Here the Hypertext Transfer Protocol request the object.

**Response -** Here the Hypertext Transfer Protocol response the object

**Session -** It is helpful to trace the data's and information about a user from one request to anotherrequest.

There are several actions that are performed in JSP actions. A JSP action is nothing but a XML tags that invokes functionality of the built-in web server.

Some of the JSP action is given as follows.

**Jsp: param -** It indicates a parameter which will be added in addition to the request of the existing parameters. It is used inside the jsp: params or jsp: include, jsp: forward blocks

**Jsp: include -** Java Servlet provisionally gives the request and response off to the specific JavaServer Page. The Control will later come again to the existing JSP as soon as the other JSP has completed. With the help of this the JSP code will be distributed among several other JSPs rather than replica.

**Jsp: forward -** This JSP action is used to give off the request/response to the other servlet orJSP. The control will not come back to the existing JSP.

**Jsp: plugin -** The ancient version of web browsers like Internet Explorer and NetscapeNavigator use various tags to embed an applet. This action creates the definite tags that are required for a browser to include an applet.

**Jsp: fallback -** This action is used to confirm that if the browser never gives support to applets.

**Jsp:getProperty -** It obtains a property from the specific JavaBean.

**Jsp: set Property -** It sets a property in the specific JavaBean

**JSP Tags**

**3. JSP scripting elements**

JSP scripting elements let you insert Java code into the servlet that will be generated from the current JSP page. There are three forms:

Expressions of the form <%= expression %> that are evaluated and inserted into the output,

Script lets of the form <% code %> that are inserted into the

Servlet’s service method, and

Declarations of the form <%! Code %> that are inserted into the body of the servlet class, outside of any existing method

**4. JSP Directive**

A JSP Directive affects the overall structure of the servlet class. It usually has the following form :< %@ directive attribute="value" %> However, you can also combine multiple attribute settings for a single directive, as follows:

<%@ directive attribute1="value1"attribute2="value2"...attribute="value" %>

There are two main types of directive: page, which lets you do things like import classes, customize the servlet superclass, and the like; and include, which lets you insert a file into the servlet class at the time the JSP file is translated into a servlet. The specification also mentions the table directive, which is not supported in JSP version 1.0, but is intended to let JSP authors define their own tags. It is expected that this will be the main new contribution of JSP 1.Implicit objects in jsp are the objects that are created by the container automatically and the container makes them available to the developers, the developer do not need to create them explicitly. Since these objects are created automatically by the container and are accessed using standard variables; hence, they are called implicit objects.

JSP Implicit Objects are as follows:

**Request implicit object**

The JSP implicit request object is an instance of a java class that implements the

Javax.servlet.http.HttpServletRequest interface. It represents the request made by the client.

The request implicit object is generally used to get request parameters, request attributes, header information and query string values.

**response implicit object**

The JSP implicit response object is an instance of a java class that implements the

Javax.servlet.http.HttpServletResponse interface. It represents the response to be given to the client. The response implicit object is generally used to set the response content type, add cookie and redirect the response.

**Out implicit object**

The JSP implicit out object is an instance of the javax.servlet.jsp.JspWriter class. It represents the output content to be sent to the client. The out implicit object is used to write the output content.

**Session implicit object**

The JSP implicit session object is an instance of a java class that implements the javax.servlet.http.HttpSession interface. It represents a client specific conversation. The session implicit object is used to store session state for a single user.

**Application implicit object**

The JSP implicit application object is an instance of a java class that implements the javax.servlet.ServletContext interface. It gives facility for a JSP page to obtain and set information about the web application in which it is running.

**Exception implicit object**

The JSP implicit exception object is an instance of the java.lang.Throwable class. It is available in JSP error pages only. It represents the occurred exception that caused the control to pass to the JSP error page.

**Config implicit object**

The JSP implicit config object is an instance of the java class that implements javax.servlet.ServletConfig interface. It gives facility for a JSP page to obtain the initialization parameters available.

**Page implicit object**

The JSP implicit page object is an instance of the java.lang.Object class. It represents the current JSP page. It is not advisable to use this page implicit object often as it consumes large memory.

**Page Context implicit object**

The JSP implicit page Context object is an instance of the javax.servlet.jsp.PageContext abstract class. It provides useful context information. Also it contains the reference to implicit objects.

**CHAPTER 9: SERVLET**

Servlets are protocol and platform independent server-side software components, written in Java. They run inside a Java enabled server or application server, such as the Web Sphere Application Server. Servlets are loaded and executed within the Java Virtual Machine (JVM) of the Web server or application server, in much the same way that applets are loaded and executed within the JVM of the Web client.

Since servlets run inside the servers, however, they do not need a graphical user interface (GUI).

In this sense, servlets are also faceless objects.

Servlets more closely resemble Common Gateway Interface (CGI) scripts or programs than applets in terms of functionality. As in CGI programs, servlets can respond to user events from an HTML request, and then dynamically construct an HTML response that is sent back to the client.

**1) The Java Servlet API**

The Java Servlet API is a set of Java classes which define a standard interface between a Web client and a Web servlet. Client requests are made to the Web server, which then invokes the servlet to service the request through this interface.

The API is composed of two packages:

javax.servlet

javax.servlet.http

The Servlet interface class is the central abstraction of the Java

Servlet API. This class defines the methods which servlets must implement, including a service () method for the handling of requests. The Generic Servlet class implements this interface, and defines a generic, protocol-independent servlet.

To write an HTTP servlet for use on the Web, we will use an even

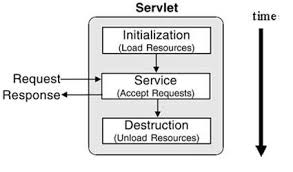
More specialized class of Generic Servlet called HttpServlet.HttpServlet provides additional methods for the processing of HTTP requests such as GET (do Get method) and POST (do Post method). Although our servlets may implement a service method, in most cases we will implement the HTTP specific request handling methods of do get and do Post.

**2) Servlet Life Cycle**

The life cycle of a servlet can be categorized into four parts:

**Loading and Instantiation-** The servlet container loads the servlet during startup or whenthe first request is made. The loading of the servlet depends on the attribute <load-onstartup> of

Web.xml file. If the attribute <load-on-startup> has a positive value then the servlet is load with loading of the container otherwise it load when the first request comes for service. After loading of the servlet, the container creates the instances of the servlet



**Initialization-** After creating the instances, the servlet container calls the init () method andpasses the servlet initialization parameters to the init () method. The init() must be called by the servlet container before the servlet can service any request. The initialization parameters persist until the servlet is destroyed. The init() method is called only once throughout the life cycle of the servlet. The servlet will be available for service if it is loaded successfully otherwise the servlet container unloads the servlet.

**Servicing the Request:** After successfully completing the initialization process, the servletwill be available for service. Servlet creates separate threads for each request. The servlet container calls the service () method for servicing any request. The service () method determines the kind of request and calls the appropriate method (doGet () or doPost ()) for handling the request and sends response to the client using the methods of the response object.

**Destroying the Servlet-** If the servlet is no longer needed for servicing any request, theservlet container calls the destroy () method. Like the init() method this method is also called only once throughout the life cycle of the servlet. Calling the destroy() method indicates to the servlet container not to sent the any request for service and the servlet releases all the resources associated with it. Java Virtual Machine claims for the memory associated with the resources for garbage collection.

**CHAPTER 10: APPLET**

Work An **applet** is a Java program that runs in a Web browser. An applet can be a fully functional Java application because it has the entire Java API at its disposal.

There are some important differences between an applet and a standalone Java application, including the following −

An applet is a Java class that extends the java.applet.Applet class.

A main() method is not invoked on an applet, and an applet class will not define main().

Applets are designed to be embedded within an HTML page.

When a user views an HTML page that contains an applet, the code for the applet is downloaded to the user's machine.

A JVM is required to view an applet. The JVM can be either a plug-in of the Web browser or a separate runtime environment.

The JVM on the user's machine creates an instance of the applet class and invokes various methods during the applet's lifetime.

Applets have strict security rules that are enforced by the Web browser. The security of an applet is often referred to as sandbox security, comparing the applet to a child playing in a sandbox with various rules that must be followed.

Other classes that the applet needs can be downloaded in a single Java Archive (JAR) file.

**Life Cycle of an Applet**

Four methods in the Applet class give you the framework on which you build any serious applet.

1. **init** − This method is intended for whatever initialization is need for your applet. It is calledafter the param tags inside the applet tag have been processed.
2. **start** − Thismethod is automatically called after the browser calls the init method. It is alsocalled whenever the user returns to the page containing the applet after having gone off to other pages.
3. **stop** −This method is automatically called when the user moves off the page on which theapplet sits. It can, therefore, be called repeatedly in the same applet.
4. **destroy** − This method is only called when the browser shutsdown normally. Becauseapplets are meant to live on an HTML page, you should not normally leave resources behind after a user leaves the page that contains the applet.
5. **paint** − Invoked immediately after the start() method, and also anytime the applet needs torepaint itself in the browser. The paint()method is actually inherited from the java.awt.

A "Hello, World" Applet

Following is a simple applet named HelloWorldApplet.java – **Example-**

import java.applet.\*;

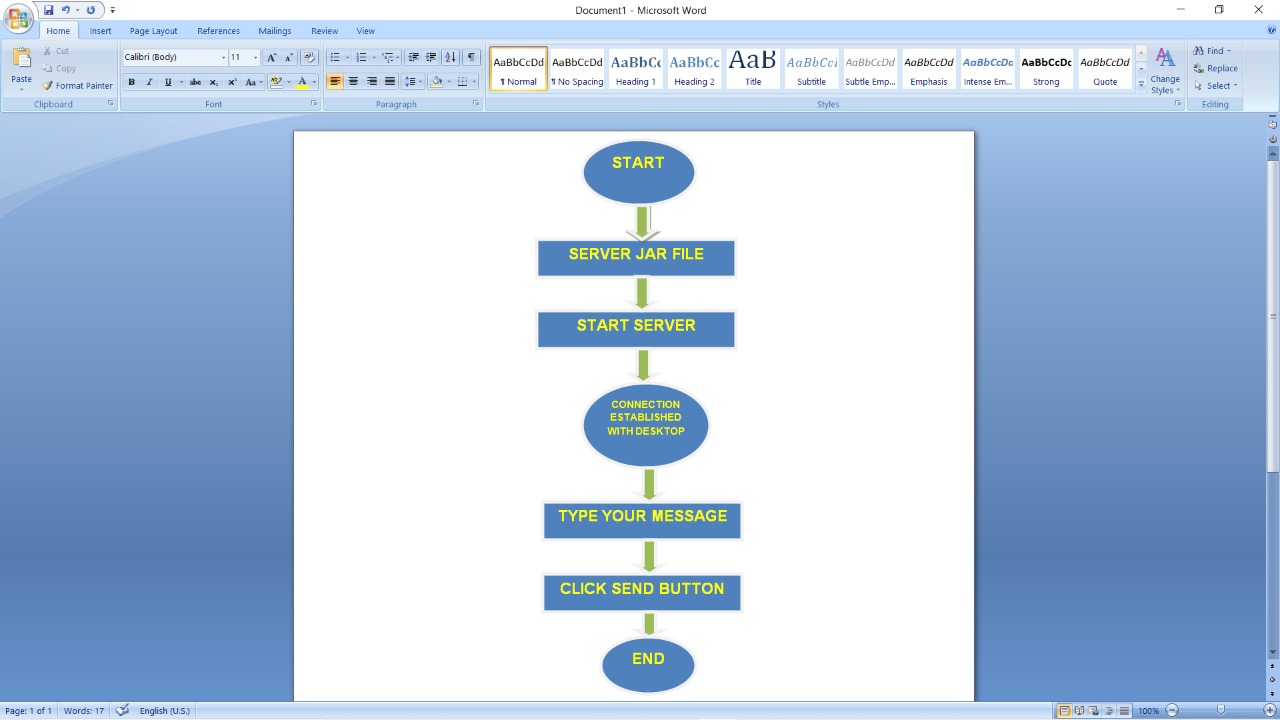
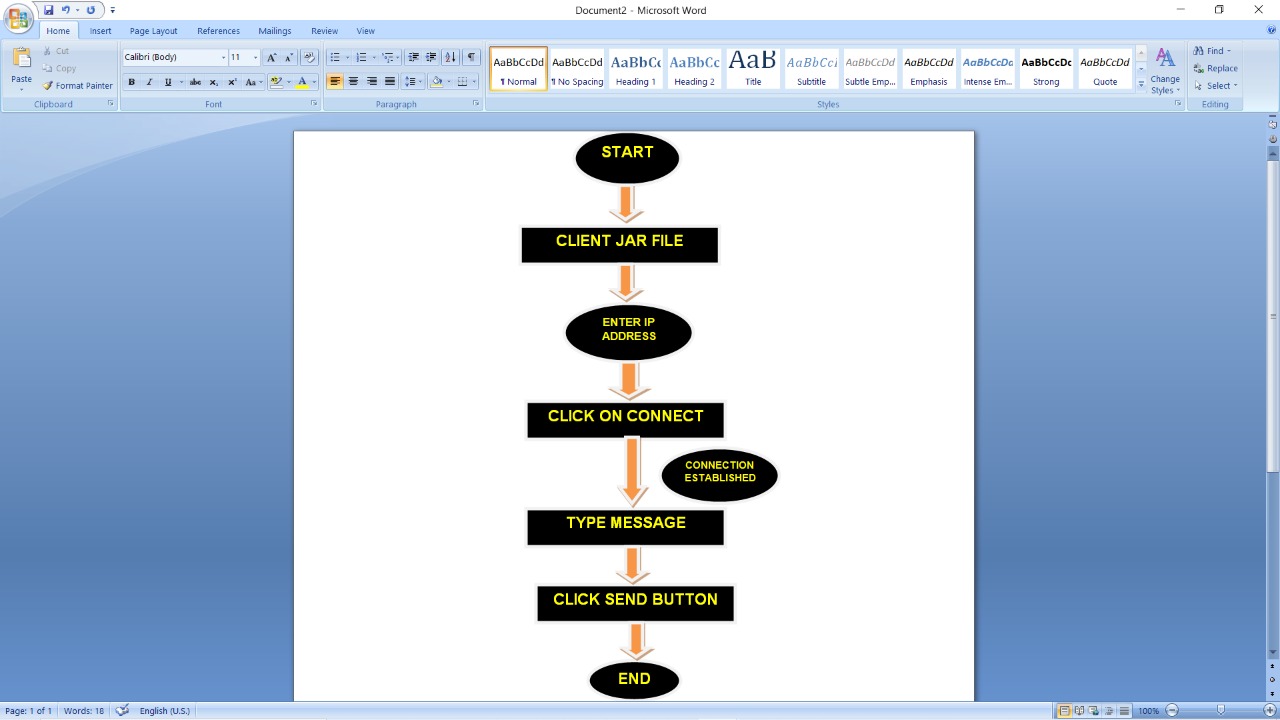
import java.awt.\*;

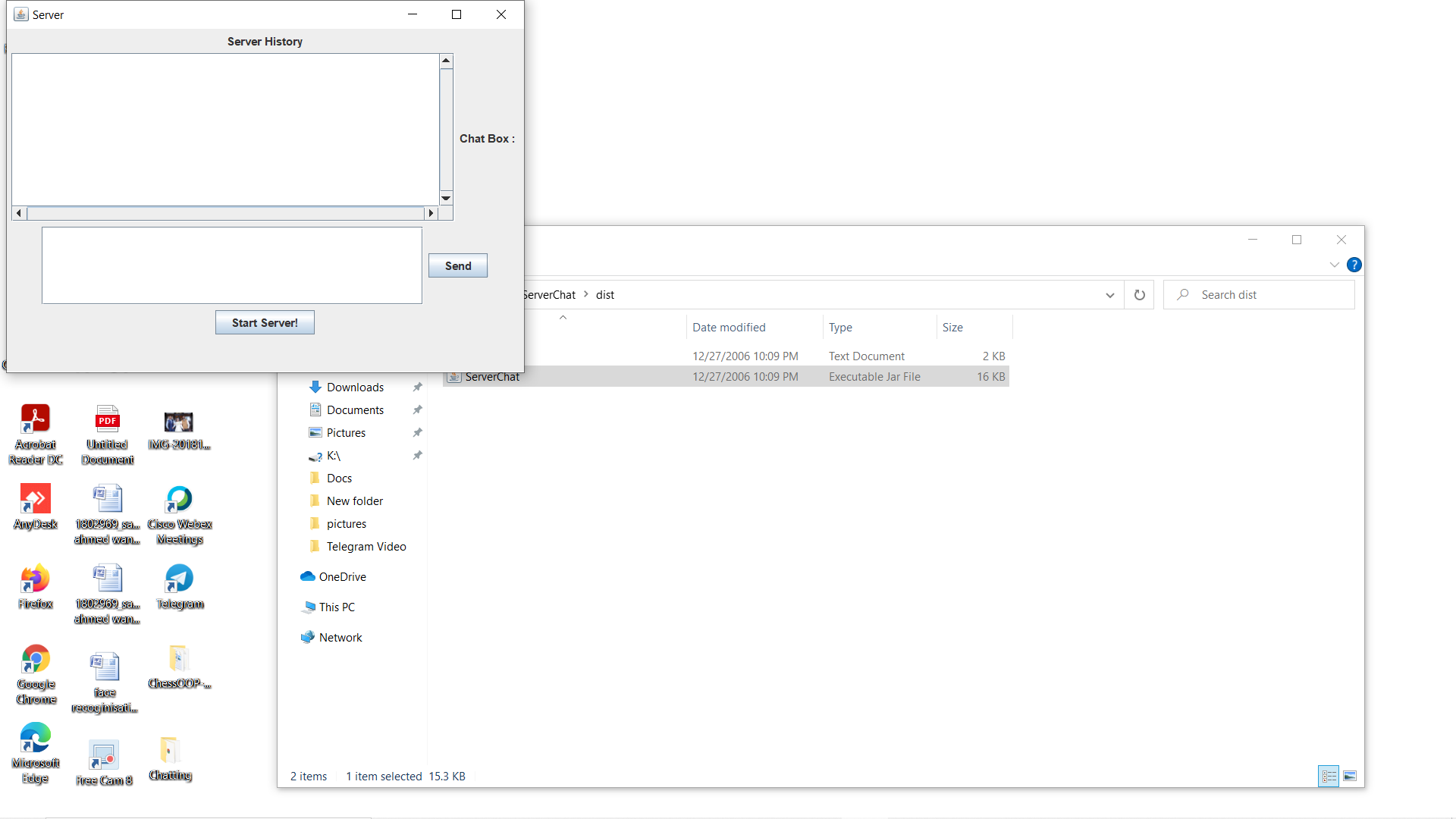
public class HelloWorldApplet extends Applet { public void paint (Graphics g) { g.drawString ("Hello World", 25, 50);

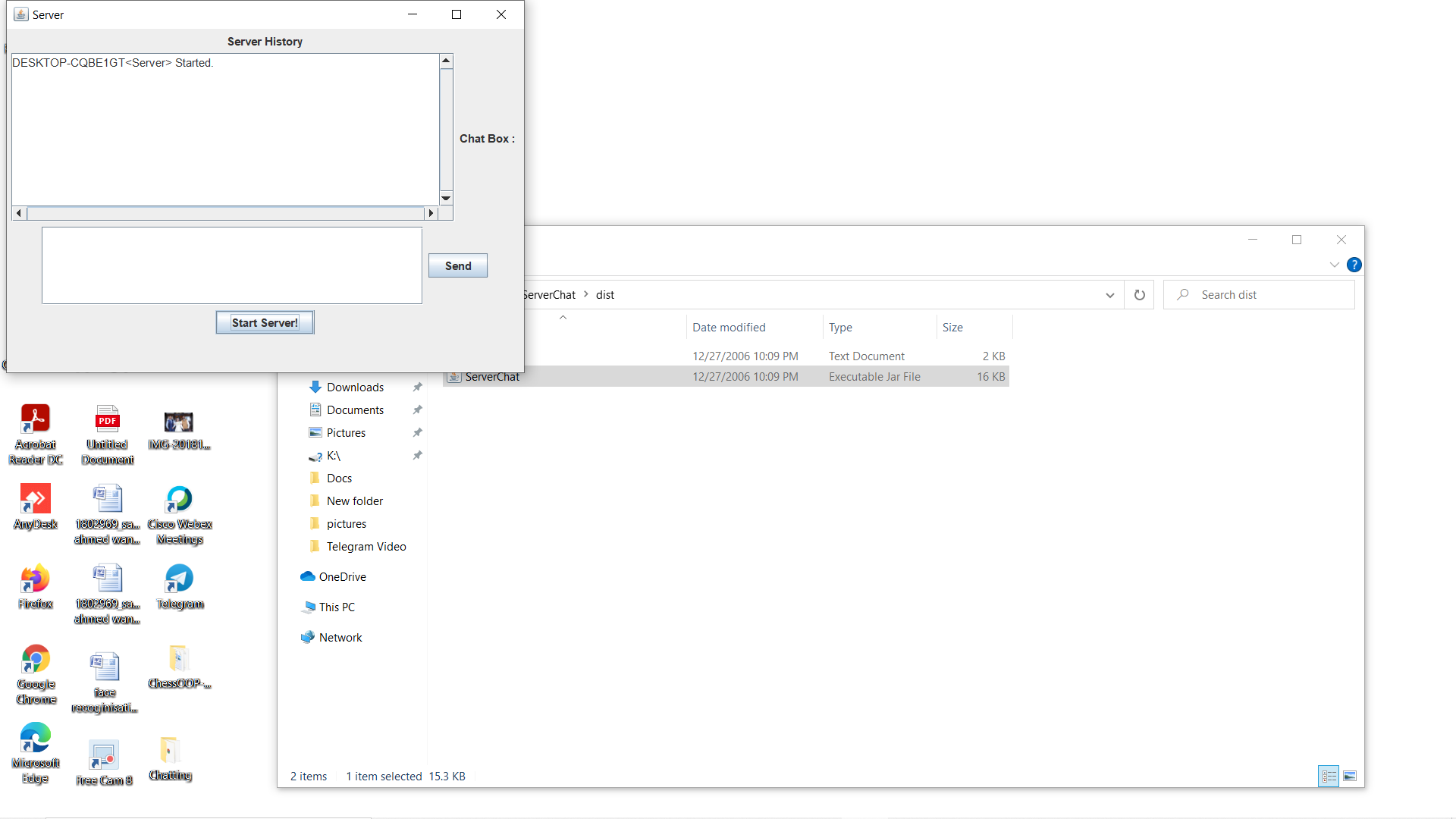
}

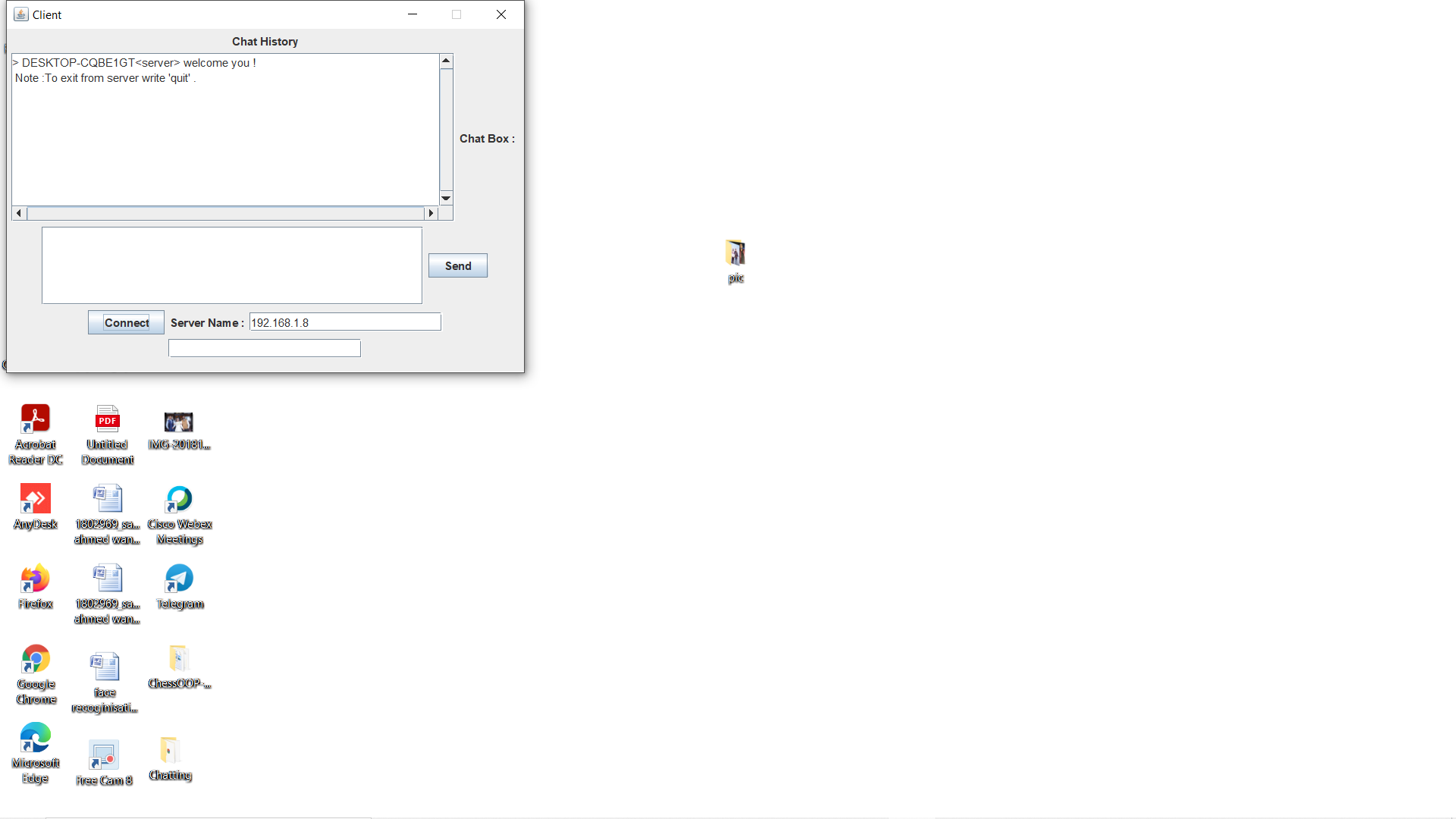
}

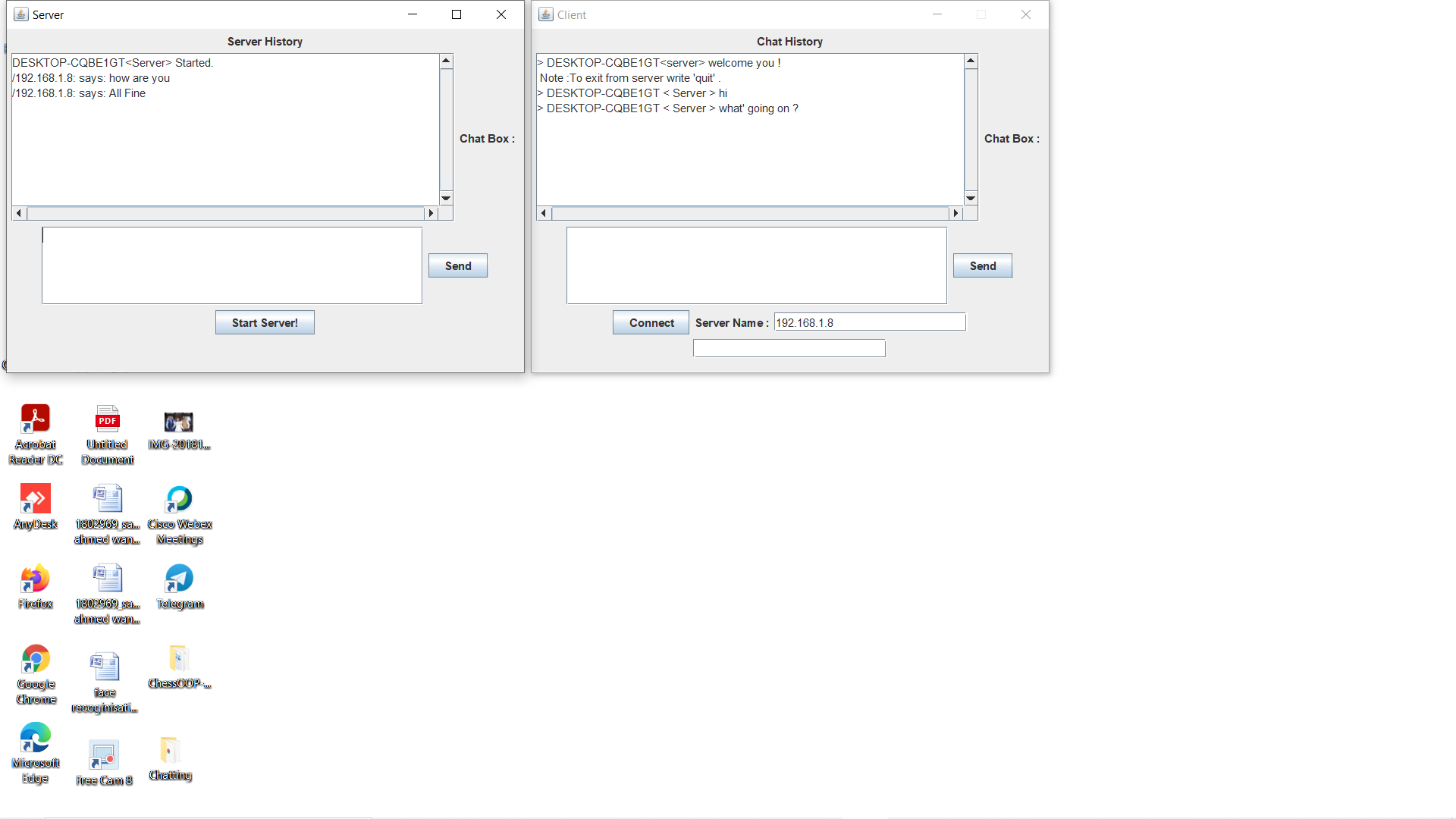
**PROJECT SNAPSHOTS**



****







**Conclusion**

As a matter of fact this project took us one year to be completed and this year is divided into two sections, six months of collecting information and six months of writing java code.

First, the theoretical section, where we have learned lots of things in networking, such as the TCP/IP protocol (number one in network protocols) and how it works, sockets, and network programming which is one of the best programming domains in Computers.

Second, the technical section which let me becomes more familiar with a new programming language such as java and its network components and support to sound package.

Having a chat server as a final year project obliged you and indirectly to go into the tiny details in networking and no one will teach you these details. Lots of experience was gained and another beautiful face of networking was discovered.

The psychological part we have learned from that project is that, nothing in life comes easily and at the same time nothing is impossible only it needs time, patience, and hard working.

**REFERENCES**

**REFERENCE BOOKS REFERRED:**

* Complete Reference Java-Herbert Schildt
* Java 6 (Black Book)-Kognet Solutions Inc.
* Software Engineering-Roger S. Pressman

**WEB SITE FOR REFERENCES:**

* http://www.eol.ucar.edu/software/java/jdk1.5
* http://java.sun.com/j2se/1.3/pdf
* http://download.oracle.com/javase/tutorial