1. **Introduction**

Integrated Development Environmentis a software program that aims at providing the Java software programmers an easy and a very efficient way to develop programs with ease.

**1.1 Aim**

Our project aims at development of applications in on the Java platform with the help of automated tools and simple compiling and debugging functionalities. It also provides the functionality for easy communication with the user of the software as well as the administrator.

**1.2 Objective**

Integrated Development Environment has the main objective of providing the user or the Java program developer the functionality to develop java software with ease and without the hassles of accessing the command prompt for debugging, compiling and the execution of our java software. It also has the prime objective to make the communication between the administrator and the user easy. One of the objectives that makes our project unique is the function or facility to transfer the byte code directly via our application to anyone through a gmail account gateway.

**2.1 Software Requirements**

1. NetBeans IDE version 6.9.1
2. JDK version 1.6
3. MySQL Query Browser Version 1.2.12

**2.2 Hardware Requirements**

a) RAM : 256 MB Minimum.

Recommended: 512GB

b) Required Hard Disk Space : 200 MB or more of available space required on

installation drive

c) Operating System : Windows 7 Service Pack 1, Windows XP

Service Pack 2, Windows Server 2003 Service

Pack 1, or Windows Vista

**3.1 JAVA**

Java is a [programming language](http://en.wikipedia.org/wiki/Programming_language) originally developed by [James Gosling](http://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) (now part of [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation)) and released in 1995 as a core component of Sun Microsystems' [Java platform](http://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](http://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B) but has a simpler [object model](http://en.wikipedia.org/wiki/Object_model) and fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [bytecode](http://en.wikipedia.org/wiki/Java_bytecode) ([class file](http://en.wikipedia.org/wiki/Class_(file_format))) that can run on any [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture). Java is a general-purpose, concurrent, class-based, 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 popular programming languages in use, particularly for client-server web applications. The original and [reference implementation](http://en.wikipedia.org/wiki/Reference_implementation_(computing)) Java [compilers](http://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](http://en.wikipedia.org/wiki/Library_(computing)) were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the [Java Community Process](http://en.wikipedia.org/wiki/Java_Community_Process), Sun relicensed most of its Java technologies under the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License). Others have also developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](http://en.wikipedia.org/wiki/GNU_Compiler_for_Java) and [GNU Classpath](http://en.wikipedia.org/wiki/GNU_Classpath).

**3.2 Design Features**

### 3.2.1 Java Platform

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called [Java bytecode](http://en.wikipedia.org/wiki/Java_bytecode), instead of directly to platform-specific [machine code](http://en.wikipedia.org/wiki/Machine_code). Java bytecode instructions are analogous to machine code, but are intended to be [interpreted](http://en.wikipedia.org/wiki/Interpreter_(computing)) by a [virtual machine](http://en.wikipedia.org/wiki/Virtual_machine) (VM) written specifically for the host hardware. [End-users](http://en.wikipedia.org/wiki/End-user) commonly use a [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java [applets](http://en.wikipedia.org/wiki/Applet).Standardized libraries provide a generic way to access host-specific features such as graphics, [threading](http://en.wikipedia.org/wiki/Thread_(computer_science)), and [networking](http://en.wikipedia.org/wiki/Computer_network).

A major benefit of using bytecode is porting. However, the overhead of interpretation

means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time compilers were introduced from an early stage that compile bytecodes to machine code during runtime.

#### 3.2.2 Implementations

[Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) officially licensed the Java Standard Edition platform for [Linux](http://en.wikipedia.org/wiki/Linux), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X), and [Solaris](http://en.wikipedia.org/wiki/Solaris_(operating_system)). In the past Sun licensed Java to Microsoft but the license expired without renewal. Because Windows does not ship with a Java software platform, a network of third-party vendors and licensees develop them for Windows and other operating system/hardware platforms.

Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with [Microsoft](http://en.wikipedia.org/wiki/Microsoft) after Sun claimed that the Microsoft implementation did not support [RMI](http://en.wikipedia.org/wiki/Java_remote_method_invocation) or [JNI](http://en.wikipedia.org/wiki/Java_Native_Interface) and had added platform-specific features of their own. Sun sued in 1997, and in 2001 won a settlement of US$20 million, as well as a court order enforcing the terms of the license from Sun. As a result, Microsoft no longer ships Java with [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), and in recent versions of Windows,[Internet Explorer](http://en.wikipedia.org/wiki/Internet_Explorer) cannot support Java applets without a third-party plugin. Sun, and others, have made available free Java run-time systems for those and other versions of Windows.

Platform-independent Java is essential to the [Java EE](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) strategy, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications, such as [Web services](http://en.wikipedia.org/wiki/Web_service), [Java Servlets](http://en.wikipedia.org/wiki/Java_Servlet), and [Enterprise JavaBeans](http://en.wikipedia.org/wiki/Enterprise_JavaBean), as well as with [embedded systems](http://en.wikipedia.org/wiki/Embedded_system) based on [OSGi](http://en.wikipedia.org/wiki/OSGi), using [Embedded Java](http://en.wikipedia.org/wiki/Embedded_Java) environments. Through the [GlassFish](http://en.wikipedia.org/wiki/GlassFish) project, Sun is working to create a fully functional, unified [open source](http://en.wikipedia.org/wiki/Open_source) implementation of the Java EE technologies.

Sun also distributes a superset of the JRE called the [Java Development Kit](http://en.wikipedia.org/wiki/Java_Development_Kit) (commonly known as the JDK), which includes development tools such as the [Java compiler](http://en.wikipedia.org/wiki/Java_compiler), [Javadoc](http://en.wikipedia.org/wiki/Javadoc), [Jar](http://en.wikipedia.org/wiki/JAR_(file_format)), and [debugger](http://en.wikipedia.org/wiki/Debugger).

#### 3.2.3 Performance

Programs written in Java have a reputation for being slower and requiring more memory than those written in [C](http://en.wikipedia.org/wiki/C_(programming_language)). However, Java programs' execution speed improved significantly with the introduction of [Just-in-time compilation](http://en.wikipedia.org/wiki/Just-in-time_compilation) in 1997/1998 for [Java 1.1](http://en.wikipedia.org/wiki/Java_version_history),the addition of language features supporting better code analysis (such as inner classes, StringBuffer class, optional assertions, etc.), and optimizations in the [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) itself, such as [HotSpot](http://en.wikipedia.org/wiki/HotSpot) becoming the default for Sun's JVM in 2000. Currently (November 2011), Java 2.0 code has approximately half the performance of C code.

Some platforms offer direct hardware support for Java; there are microcontrollers that can run Java in hardware instead of a software JVM, and [ARM](http://en.wikipedia.org/wiki/ARM) based processors can have hardware support for executing Java bytecode through its [Jazelle](http://en.wikipedia.org/wiki/Jazelle) option.

### 3.2.4 Automatic memory management

Java uses an [automatic garbage collector](http://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) to manage memory in the [object lifecycle](http://en.wikipedia.org/wiki/Object_lifetime). The programmer determines when objects are created, and the Java runtime is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the [unreachable memory](http://en.wikipedia.org/wiki/Unreachable_memory) becomes eligible to be freed automatically by the garbage collector. Something similar to a [memory leak](http://en.wikipedia.org/wiki/Memory_leak) may still occur if a programmer's code holds a reference to an object that is no longer needed, typically when objects that are no longer needed are stored in containers that are still in use. If methods for a nonexistent object are called, a "null pointer exception" is thrown.

One of the ideas behind Java's automatic memory management model is that programmers can be spared the burden of having to perform manual memory management. In some languages, memory for the creation of objects is implicitly allocated on the[stack](http://en.wikipedia.org/wiki/Stack_(data_structure)), or explicitly allocated and deallocated from the [heap](http://en.wikipedia.org/wiki/Dynamic_memory_allocation). In the latter case the responsibility of managing memory resides with the programmer. If the program does not deallocate an object, a [memory leak](http://en.wikipedia.org/wiki/Memory_leak) occurs. If the program attempts

to access or deallocate memory that has already been deallocated, the result is undefined and difficult to predict, and the program is likely to become unstable and/or crash. This can be partially remedied by the use of [smart pointers](http://en.wikipedia.org/wiki/Smart_pointer), but these add overhead and complexity. Note that garbage collection does not prevent "logical" memory leaks, i.e. those where the memory is still referenced but never used.

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily. Explicit memory management is not possible in Java.

Java does not support C/C++ style [pointer arithmetic](http://en.wikipedia.org/wiki/Pointer_(computing)), where object addresses and unsigned integers (usually long integers) can be used interchangeably. This allows the garbage collector to relocate referenced objects and ensures type safety and security.

As in [C++](http://en.wikipedia.org/wiki/C%2B%2B) and some other object-oriented languages, variables of Java's [primitive data types](http://en.wikipedia.org/wiki/Primitive_data_type) are not objects. Values of primitive types are either stored directly in fields (for objects) or on the [stack](http://en.wikipedia.org/wiki/Stack-based_memory_allocation) (for methods) rather than on the heap, as commonly true for objects (but see [Escape analysis](http://en.wikipedia.org/wiki/Escape_analysis)). This was a conscious decision by Java's designers for performance reasons. Because of this, Java was not considered to be a pure object-oriented programming language. However, as of Java 5.0, [autoboxing](http://en.wikipedia.org/wiki/Autoboxing) enables programmers to proceed as if primitive types were instances of their wrapper class.

Java contains multiple types of garbage collectors. By default, HotSpot uses the [Concurrent Mark Sweep collector](http://en.wikipedia.org/wiki/Concurrent_Mark_Sweep_collector), also known as the CMS Garbage Collector. However, there are also several other garbage collectors that can be used to manage the Heap. For 90% of applications in Java, the CMS Garbage Collector is good enough.

**3.3 MySQL**

MySQL stands for Structured Query Language. MySQL Server is a relational database server. It is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of MySQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time).

**SOFTWARE LIFE CYCLE** is the series of identifiable stages that a software product undergoes during its life time. It is a descriptive and diagrammatic representation of s/w life cycle. It represents all the activities required to make a software product transit through its life cycle phases.

In our project named “Integrated Development Environment” we are using LINEAR SEQUENTIAL MODEL. It suggests a systematic and sequential approach to software development that begins at system level and progress through analysis, design, coding, testing and support.

REQUIREMENT ANALYSIS AND SPECIFICATION

DESIGN

FEASIBILITY

STUDY

IMPLEMENTATION

MAINTENANCE

TESTING

**Figure1. Shows various stages of LINEAR SEQUENTIAL MODEL.**

The various stages are:

1. Feasibility Study
2. Requirement Analysis and Specification
3. Design
4. Implementation
5. Testing
6. Maintenance

**4.1 Feasibility Study**

Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points:

* Technical Feasibility
* Operational Feasibility

1. Technical Feasibility:

The project entitled “Integrated Development Environment” is technically feasible because the project was developed in High end technologies like Java Core which runs on all the modern day operating systems namely Apple MAC, Microsoft Windows and even Linux.

b) Operational Feasibility:

Integrated Development Environment is operationally feasible because it makes use of present day coding methodologies and present day databases to implement all the functionalities in an orderly manner so as the user is able to create his softwares on the go and with much ease. Also all the operational aspects of the project are user friendly and are well known by most of the people in the technical fraternity.

**4.2 Requirement analysis and Specification**

**Table of Contents**

1. Introduction

1. 1.1. Purpose
2. 1.2. Scope
3. 1.3. Definitions
4. 1.4. Overview
5. 1.5. References
6. 2. General description

2.1. Perspective of our system

* 1. 2.1.1. User interface
  2. 2.1.2. Hardware interface

2.1.3. Software interface

2.1.4. Memory constraint

2.2 User characteristics

2.3. General constraints

1. 3. Specific Requirements

3.1. Functional Requirements

3.2. Technical Requirements

3.3. Environment

3.3.1. Operation Environment

3.3.2. Development Environment

3.4. Performance Requirement

1. 3.5. Attributes

3.5.1. Quality Attributes

3.5.2. System Attributes

3.5.2.1. Security

3.5.2.2. Maintainability

3.6. Logical database requirements

3.7. Other Requirements

**Preface**

The purpose of this document is to specify the project plan to develop the Query Management System. The name of our project is “Integrated Development environment”. This document outlines a brief plan about how the project is to be shaped and also includes the milestones and deliverables. The document will serve as a guide for the student, developing the product as part of the project. Updates of this document will serve to record the progress of the project.

**1. Introduction**

Integrated Development Environment is a comprehensive and integrated system which holds up all the information at one place and has the capability to be modified by a remote access to the system which is done most probably in our system via the usage of sms facility which can be used for sending out information as well as for reading the received message and thereby acting upon the specified task which has been specified in the message. The receiving and sending of the message is facilitated through a computer via the usage of AT commands through which upon the availability of an interfaced connection with the mobile the computer has the ability to operate the mobile using the AT commands which we can configure as per our needs. Query Management System also keeps a database which is accessed upon receiving a message and the functions which have been predefined are made to access the system and the changes are made in the database correspondingly. This remote access of the database marks the main feature of the system through which one can see and alter his/her records in the database as per his/her choice. To limit the use of any misuse of the information a user will be given a randomly generated password to make sure that the access has been provided to the right person and has been authenticated to the correct person.

* 1. Purpose

The purpose of creating this one of a kind Integrated Development Environment is to make software development in the Java core language easy and well defined. It makes the user do away with the conventional method of using the command prompt to compile the code. It also makes the interpreting and the running of the byte code easier too. With the user friendly graphical user interface in the software making softwares in the language like Java becomes a piece of cake. Also with the intrellisense feature, writing the codes become more predictive and with its in-built dictionary more efficient.

* 1. Scope

Our software has its scope to all the novice Java core application developers who want to create stand alone applications on the java platform. The future scope of this project is farfetched with extending its functionality to enable the development of applications in the Java advanced field too.

1.3 Overview

Integrated development environment (IDE) (also known as integrated design environment, integrated debugging environment or interactive development environment) is a [software application](http://en.wikipedia.org/wiki/Software_application) that provides comprehensive facilities to [computer programmers](http://en.wikipedia.org/wiki/Computer_programmer) for [software development](http://en.wikipedia.org/wiki/Software_development). An IDE normally consists of:

* A [source code editor](http://en.wikipedia.org/wiki/Source_code_editor)
* A [compiler](http://en.wikipedia.org/wiki/Compiler) and an [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29)
* [Build automation](http://en.wikipedia.org/wiki/Build_automation) tools
* A [debugger](http://en.wikipedia.org/wiki/Debugger)

It is very much realistic, having full provision of Graphical User Interfaces ( GUI ) that helps the user to make a instant reaction to different events by just clicking. The provision for shortcuts has also been defined in case the user is comfortable with the shortcuts.

The application provides the provision to user for creating applications with the java technology that has in build complier and Interpreter support to compile the code that will generate the .class file and correspondingly will show the output in the textbox or the console.

**2. General description**

2.1. Perspective of our system

2.1.1 USER INTERFACE

Computer desktop system or a laptop computer.

2.1.2 HARDWARE INTERFACE

Normal PC, USB Cable

2.1.3 SOFTWARE INTERFACE

Front end -> Java core

Back end -> MYSQL server

2.1.4 MEMORY CONSTRAINTS:

At least 256 MB RAM and 50 MB free space on hard disk will be required for running the application.

2.2 User characteristics:

1. Knowledgeable user
2. Need to have basic computer knowledge

2.3 Constraints

The user has to have basic knowledge of coding in the Java language. Elementary knowledge of the programming paradigms and methodologies are needed.

**3. Specific requirements:**

This section contains the software requirements to a level of detail sufficient to enable designers to design the system, and the testers to test the system.

3.1. Functional Requirements

The purpose of this section is to obtain agreement regarding what the system will need to do when interacting with the user.

3.2. Technical Requirements

The purpose of this section is to obtain agreement regarding the platforms to be used for deploying the working application and for developing the working application. This is input to assist managerial decisions (i.e., purchasing) as well as technical analysis (i.e., constraints imposed by the platform).

3.3 Environment

3.3.1 Operational Environment:

To run the application in the production mode, one database server is needed and one operating system.

3.3.2 Development Environment:

Requires JDK version 1.6 with JVM and Java Runtime Environment and remaining same as Operational Environment

3.4 Safety & Reliability Requirements:

By incorporating a robust and proven SQL into the system, reliable performance and

integrity of data is ensured. There must be a power backup for server system.

3.5 Software Attributes

3.5.1 Quality attributes

Menu-driven programs with user friendly interface with simple hyperlinks. It is very easy to use. Backup mechanisms are considered for maintainability of software as well as database. As it is object oriented reusability exists.

3.5.2 System attributes

3.5.2.1 Security

The application will be hidden if the user wants it to be.

3.5.2.2 Maintainability

The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

3.5.2.3 Portability

The application will be easily portable on any window or MAC or Linux based system. Because its implemented in Java, its very much portable and easy to deploy.

3.6 Logical database requirements

The following information will be placed in a database:

1. Login Id and password of the users

* 1. 2. Login Id and password of the administrator
  2. 3.Queries made by the users
  3. 4. Reply by the administrators to the users

3.7 Other Requirements

None

**4.3 Design**

The design phase begins after all the requirements are gathered. The designing phase aims at developing the graphical user interface i.e. what the software/application will look like to the user and the back end designing i.e. the database designing. The primary objective of the design phase is to create a design that satisfies the agreed application requirements. In the design phase the SDLC process continues to move from the "what" questions of the analysis phase to the "how" questions.

 The requirements prototype that was developed earlier during the analysis phase is gradually improved and extended to include all the specified functions of the application.

The design phase hence contains:

* Front-end Designing
* Backend Designing

1. Front-end Designing:

This is the graphical user interface designing. Integrated Development Environment’s front-end is designed in Net Beans version 7.0.1 application. It is user friendly. Integrated Development Environment has several buttons on its front end i.e. file, help, file editing, window, run file etc. All the functions can easily be understood by the user as all the buttons are labeled in layman language and have their usual meaning.

1. Backend Designing:

This is the designing of the functions of Integrated Development Environment that are not visible to the user and which function in the background to make it possible for the user to access and remotely view and change the data fields specified in the system. Each user will have an associated password which he will have to mention in order to gain access to the system and thereby make it possible for him/her to make changes to the values in the database. The functions associated in the database are that of

1. ADD – to add another user
2. DELETE – to remove an existing user
3. MODIFY – to change certain things to the user
4. QUERY- to view queries by the user

**Use Case Diagram**

Admin

User

**Zero Level DFD**

.java file

BYTE CODE (.class file)

Can run on different platforms

**One Level DFD**

ADMINISTRATOR

Response to the user

Issues from the user

Report error codes or issues

Open/create projects

Mail byte code

Byte code (.class files)

Interpreted output

USER

**Fig**

**ER Diagram**

Queries

manages

Admin

Account

Creates

User

**4.4 Implementation**

Implementation phase starts after the designing of the software/application is done. In other words the design phase is the base for implementation. The implementation stage is defined as the system or system modifications being installed and made operational in a production environment. The implementation phase marks the beginning of the coding phase.

The coding of the front-end and backend can be done using the Java(core) and for backend MY SQL can be used.

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development.Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s systems design had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering.

**4.4.1 Compiler**

Complier has been embedded for the written code to be compiled and a byte code(.java file) is created in the same directory.

Java Virtual Machine

Java Interpreter

Byte Code

(Classname.class)

Source Code

(filename.java)

Operating System

Java Compiler

Text Editor

Fig :- 2.2 Interpreter

The process to compile a java program

Fig 2.2 Compiler

**4.4.2 Interpreter**

We can run Java on most platforms provided a platform must has a Java interpreter. That is why Java applications are platform independent. Java interpreter translates the Java bytecode into the code that can be understood by the Operating System. Basically, A Java interpreter is a software that implements the Java virtual machine and runs Java applications. As the Java compiler compiles the source code into the Java bytecode, the same way the Java interpreter translates the Java bytecode into the code that can be understood by the Operating System.

When a Java interpreter is installed on any platform that means it is JVM (Java virtual machine) enabled platform. It (Java Interpreter) performs all of the activities of the Java run-time system. It loads Java class files and interprets the compiled byte-code.The interpreter also serves as a specialized compiler in an implementation that supports dynamic or "just in time," compilation which turns Java byte-code into native machine instructions.

The application provides the provision to user for creating applications with the java technology that has in build complier and Interpreter support to compile the code that will generate the .class file and correspondingly will show the output in the textbox or the console.

* + 1. **Modules of IDE**

Different modules in this application are :-

* SIGN IN AND SIGN UP

Users that are already registered for using this application can use it directly by providing their access Id and Password while others can Sign Up in a simple fashion in order to have their access.

* WRITING CODE

A text area is provided for users to write their logical code using java only.

* COMPILER

Complier has been embedded for the written code to be compiled and a byte code(.java file) is created in the same directory.

* INTERPRETER

Interpreter has been embedded to Interpret the byte code created by the Compiler.

* INTELLISENSE (STATIC)

Intellisense has been provided that lower down the task of writing the entire syntax. ie it incorporates reusability of the correct syntax.

* REPORT ISSUE

The users can report their queries to the administrator of our application.

This is the login page for the administrator. Only the authenticated user can request for the queries.

* KEYBOARD SHORTCUTS

Users can have the knowledge of all the shortcuts used in the project and can employ the same to save time and effort.

* ADMIN LOGIN

pass the screen and have the authority to response to the reply of different users.

* TRANSFER CODE VIA WEB

Here user has the provision for transferring the code created to his email Id or can mail to anybody. A google gateway has been provided to transfer the code.

**4.5 Maintenance**

There are two goals of Maintenance in SDLC: Increase the ability of the software and avoidance of failures.

**Better Adaptability of the Software** – SDLC’s idea of maintenance is very beneficial not only to the developers but also for the software itself.   With the use of available data, developers will learn something new.   In this account, they will be able to apply this knowledge to the software. Maintenance will ensure that the software will last for a very long time.

**Avoidance of Failures** – Maintenance does not only ensure that the problem will be fixed but maintenance should prevent the same event from happening again. Software will always have that “learning” property as long as they are handled well by the developers.   Developers will often work on the problem and at the same time give updates on the system to prevent failures.

**5.1 Modifications and Improvement over the existing implementations**

* The present implementation of the project has the intellisense feature implemented but is static. In the future we aim to make the intellisense feature more dynamic
* Also in the future we aim to implement various palettes so that the user can drag and drop various objects like the text box, the panel, window panes, etc.

**Conclusion**

IDE is designed to maximize programmer productivity by providing tightly-knit components with similar [user interfaces](http://en.wikipedia.org/wiki/User_interface). However, because an IDE is a complicated piece of software by its very nature, this higher productivity only occurs after a lengthy learning process.

It has piece together command line utilities in a cohesive unit, which theoretically reduces the time to learn a language, and increases developer productivity. It is also thought that the tight integration of development tasks can further increase productivity. For example, code can be parsed while being written, providing instant feedback on syntax errors.

**Features of the Project**

* Embedded compiler and Interpreter
* Security
* Portability
* Graphical user interface ( GUI )
* Database Connectivity
* features like undo, redo etc.
* Transfer byte code via web

**Bibliography**

1. http://apl.jhu.edu/~hall/java/IDEs.html

2. http://faq.programmerworld.net/

3. http://netbeans.org

4. http://www.cs.unc.edu/~jbs/images/java/java-dataflow.GIF

5. http://www.javaworld.com/javaworld/jw-05-1997/jw-05-indepth.html

6. http://www.jcreator.com/

7. http://www.roseindia.net/java/java-introduction/javatools/java-inetrpreter.shtml

**NewFrame1.java**

public class NewJFrame1 extends javax.swing.JFrame {

/\*\* Creates new form NewJFrame1 \*/

public NewJFrame1() {

initComponents();

//COMPILER

private void jMenuItem19ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTextArea2.setText("");

String commands[]= {"C:\\Program Files\\Java\\jdk1.6.0\_15\\bin\\javac","-classpath","C:\\",d+f};

//String commands[]= {"C:\\Program Files\\Java\\jdk1.6.0\_15\\bin\\javac","-classpath","C:\\","C:\\IDE\\hello.java"};

try

{

Process ps=Runtime.getRuntime().exec(commands);

Scanner sc=new Scanner(ps.getErrorStream());

boolean flag=true;

long count=0;

while(sc.hasNextLine())

{

jTextArea2.append(sc.nextLine()+"\n");

count = count+1;

//value = value + sc.nextLine();

flag=false;

}

//jLabel9.setText(value);

if(flag)

{

jTextArea2.setText("");

jTextArea2.append("compiled");

jLabel9.setText("");

}

String err ="";

if(jTextArea2.getText()!="compiled")

{

String var = jTextArea2.getText();

err = var.substring(var.length()-9,var.length());

}

jLabel9.setVisible(true);

jLabel9.setText(err);

/\*String co = String.valueOf(count);

jLabel9.setVisible(true);

jLabel9.setText(co);\*/;sc.close();

}

catch(Exception e)

{

System.out.println(e.getMessage());

}

}

private void jMenuItem17ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextArea1KeyPressed(java.awt.event.KeyEvent evt) {

int k=evt.getKeyCode();

switch(k)

{

case KeyEvent.VK\_SPACE:

}

}

//INTERPRETER

private void jMenuItem16ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//String commands[]= {"C:\\Program Files\\Java\\jdk1.6.0\_15\\bin\\java","-classpath","C:\\","Array2"};

jTextArea2.setText("");

{

String s=jTextArea1.getText();

String classname[] = new String[1000];

StringTokenizer st = new StringTokenizer(s);

for (int i = 0; st.hasMoreTokens(); i++)

{

classname[i] = st.nextToken();

}

String commands[]= {"C:\\Program Files\\Java\\jdk1.6.0\_15\\bin\\java","-classpath",d,classname[1]};

try

{

Process ps=Runtime.getRuntime().exec(commands);

Scanner sc=new Scanner(ps.getErrorStream());

boolean flag=true;

while(sc.hasNextLine())

{

jLabel9.setVisible(true);

jLabel9.setText(sc.nextLine());

// jTextArea1.setText(sc.nextLine());

//System.out.println(sc.nextLine());

flag=false;

}

if(flag)

{

jLabel1.setText("done!!!!");

System.out.println("Done !!");

}

/////

InputStream stderr = ps.getInputStream();

InputStreamReader isr = new InputStreamReader(stderr);

BufferedReader br = new BufferedReader(isr);

String line = null;

while ( (line = br.readLine()) != null)

{

jTextArea2.append(line+"\n");

System.out.println(line);

jLabel9.setText((""));

//System.out.println("");

}

sc.close();

}

catch(Exception e)

{

System.out.println(e.getMessage());

}

}

// jTextArea1.setText("you have not compiled or error in compilation");

}

//OPEN

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {

jTextArea1.setEnabled(true);

FileDialog fd = new FileDialog(this,"OPEN",FileDialog.LOAD);

fd.setVisible(true);

String dir= fd.getDirectory();

//

d= fd.getDirectory();

//

String file= fd.getFile();

//

f= fd.getFile();

//

String s = dir+file;

setTitle(file);

jTextField2.setText(d);

jTextField1.setText(f);

try{

FileInputStream fin = new FileInputStream(s);

int ch;

String s11="";

FileInputStream fstream = new FileInputStream(s);

while ((ch = fin.read()) != -1)

s11=s11+(char)ch;

fin.close();

String j = s11;

jTextArea1.setText(j);

// jTextField1.setText(j);

}

catch(Exception e)

{

System.out.println(e);

} // TODO add your handling code here:

}

**Front.java**

private void jLabel5MousePressed(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

new CreateAccount().setVisible(true);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

/////// temporary bhej rhe hain ise main page pe....

String s1=jTextField1.getText();

char ch1[]=jPasswordField1.getPassword();

String s2=String.valueOf(ch1);

Connection con=null;

Statement st;

ResultSet rs;

try

{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st=con.createStatement();

rs=st.executeQuery("select name from user where username='"+s1+"' and password='"+s2+"'");

if(rs.next())

{

//set the final variable for username that is to be used at other pages..

username = jTextField1.getText();

setVisible(false);

new NewJFrame1().setVisible(true);

}

if(!rs.next())

{

jTextField1.setText("");

jPasswordField1.setText("");

jLabel7.setVisible(true);

jLabel1.setText("username or password incorrect...");

}

rs.close();

con.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

new Adminlogin().setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new front().setVisible(true);

}

});

}

**Transfer.java**

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String[] to={jTextField3.getText()};

String[] cc={"gurjeet\_1991@yahoo.co.in"};

String[] bcc={"gurjeet\_1991@yahoo.co.in"};

//This is for google

sendMail(jTextField1.getText(),jTextField2.getText(),"smtp.gmail.com","465","true",

"true",true,"javax.net.ssl.SSLSocketFactory","false",to,cc,bcc,

"mail contain .class file attached","This is my style...of reply..If u send virus mails..",jTextField4.getText());

}

public void call()

{

jLabel6.setVisible(true);

jLabel6.setText("Invalic Email Id or password");

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\*java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new tranfer().setVisible(true);

}

});\*/

}

public synchronized boolean sendMail(String userName,String passWord,String host,String port,String starttls,String auth,boolean debug,String socketFactoryClass,String fallback,String[] to,String[] cc,String[] bcc,String subject,String text, String uparsefile){

try

{

Properties props = new Properties();

//Properties props=System.getProperties();

props.put("mail.smtp.user", userName);

props.put("mail.smtp.host", host);

if(!"".equals(port))

props.put("mail.smtp.port", port);

if(!"".equals(starttls))

props.put("mail.smtp.starttls.enable",starttls);

props.put("mail.smtp.auth", auth);

if(debug){

props.put("mail.smtp.debug", "true");

}else{

props.put("mail.smtp.debug", "false");

}

if(!"".equals(port))

props.put("mail.smtp.socketFactory.port", port);

if(!"".equals(socketFactoryClass))

props.put("mail.smtp.socketFactory.class",socketFactoryClass);

if(!"".equals(fallback))

props.put("mail.smtp.socketFactory.fallback", fallback);

Session session = Session.getDefaultInstance(props, null);

session.setDebug(debug);

MimeMessage msg = new MimeMessage(session);

msg.setText(text);

msg.setSubject(subject);

//////////////

String att = uparsefile;

DataSource ds = new FileDataSource(att);

msg.setDataHandler(new DataHandler(ds));

msg.setFileName("att");

///////////////

// msg.setFileName("C:\\Users\\GURJEET\\Desktop\\11111.txt");

msg.setFrom(new InternetAddress("p\_sambasivarao@sutyam.com"));

for(int i=0;i<to.length;i++){

msg.addRecipient(Message.RecipientType.TO, new InternetAddress(to[i]));

}

for(int i=0;i<cc.length;i++){

msg.addRecipient(Message.RecipientType.CC, new InternetAddress(cc[i]));

}

for(int i=0;i<bcc.length;i++){

msg.addRecipient(Message.RecipientType.BCC, new InternetAddress(bcc[i]));

}

msg.saveChanges();

Transport transport = session.getTransport("smtp");

transport.connect(host, userName, passWord);

transport.sendMessage(msg, msg.getAllRecipients());

transport.close();

return true;

}

catch (Exception mex)

{

call();

mex.printStackTrace();

return false;

}

}

**AdminLogin.java**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(jTextField1.getText().equals("pratik") || jTextField2.getText().equals("gurjeet"))

{

new Adminwork().setVisible(true);

dispose();

}

jLabel3.setVisible(true);

jLabel3.setText("invalid enteries");

**AdminWork.java**

public class Adminwork extends javax.swing.JFrame {

/\*\* Creates new form Adminwork \*/

public Adminwork() {

initComponents();

jLabel2.setVisible(false);

Connection con = null;

Statement st;

ResultSet rs;

try

{

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

con = DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st = con.createStatement();

rs = st.executeQuery("Select username from codes");

while(rs.next())

{

jComboBox1.addItem(rs.getString("username"));

}

//else

// jOptionPane1.showMessageDialog(this, "Wrong details entered.", "Error", javax.swing.JOptionPane.ERROR\_MESSAGE);

rs.close();

con.close();

}

catch(Exception e)

{

System.out.println(e.getMessage());

}

;

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jComboBox1 = new javax.swing.JComboBox();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jLabel1 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jLabel2 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(204, 204, 255));

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "<--- DEFAULT --->" }));

jComboBox1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jComboBox1ActionPerformed(evt);

}

});

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel1.setText("Select the User..");

jButton1.setText("Send Reply..");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel2.setText("jLabel2");

jButton2.setText("Back");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 408, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(17, 17, 17)

.addComponent(jLabel1)

.addGap(35, 35, 35)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(72, 72, 72)

.addComponent(jButton2)

.addGap(18, 18, 18)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 98, Short.MAX\_VALUE)

.addComponent(jLabel2)

.addGap(56, 56, 56))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.

addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(11, 11, 11)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 240, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jLabel2)

.addComponent(jButton2))

.addContainerGap(18, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Connection con=null;

Statement st;

ResultSet rs;

try

{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st=con.createStatement();

rs=st.executeQuery("Select \* from codes where username='"+jComboBox1.getSelectedItem()+"'");

if(rs.next())

{

String n = rs.getString("code");

jTextArea1.setText(n);

}rs.close();

con.close();

}

catch(Exception e)

{

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Connection con;

PreparedStatement st;

ResultSet rs;

try{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st=con.prepareStatement("insert into replytouser values(?,?)");

String s1= (String) jComboBox1.getSelectedItem();

st.setString(1,s1);

st.setString(2, jTextArea1.getText());

int x=st.executeUpdate();

if(x>0)

{

jLabel2.setVisible(true);

jLabel2.setText("send..");

}

}

catch(Exception e)

{}

}

**Create account**

package ide;

import java.sql.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*\*

\*

\* @author GURJEET

\*/

public class CreateAccount extends javax.swing.JFrame {

/\*\* Creates new form CreateAccount \*/

public CreateAccount() {

initComponents();

jLabel1.setVisible(false);

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel7 = new javax.swing.JLabel();

jPasswordField2 = new javax.swing.JPasswordField();

jPasswordField1 = new javax.swing.JPasswordField();

jLabel4 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jLabel6 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jTextField1 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(204, 204, 255));

jPanel1.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0, 0)));

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel7.setForeground(new java.awt.Color(51, 0, 153));

jLabel7.setText("Sign Up Form..");

jPasswordField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jPasswordField2ActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel4.setText("Password");

jLabel5.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel5.setText("Confirm Password");

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel3.setText("Username");

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel2.setText("Name");

jButton1.setIcon(new javax.swing.ImageIcon("C:\\Users\\GURJEET\\Desktop\\New folder\\continue.jpg")); // NOI18N

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel6.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

jLabel6.setText("Email Id");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jTextField2.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusGained(java.awt.event.FocusEvent evt) {

jTextField2FocusGained(evt);

}

});

jLabel1.setText("jLabel1");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel4)

.addComponent(jLabel3)

.addComponent(jLabel2)

.addComponent(jLabel5)

.addComponent(jLabel6))

.addGap(42, 42, 42)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel7)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPasswordField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 114, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 122, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 183, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 164, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 251, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(75, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(22, 22, 22)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 45, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel6))

.addGap(25, 25, 25)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGap(11, 11, 11))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String s1=jTextField1.getText();

String s2=jTextField2.getText();

char ch1[]=jPasswordField1.getPassword();

char ch2[]=jPasswordField2.getPassword();

String s3=String.valueOf(ch1);

String s4=String.valueOf(ch2);

String s5=jTextField3.getText();

System.out.println(s1+" "+s2+" "+s3+" "+s4);

if(jTextField1.getText() == null)

{

jLabel1.setText("name field empty..");

}

Connection con=null;

Statement st;

ResultSet rs1,rs2;

try

{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st=con.createStatement();

rs1=st.executeQuery("Select \* from user where username='"+s2+"'");

if(rs1.next())

{

jTextField2.setForeground(Color.red);

jTextField2.setText("Username already exists");

}

else

{

rs2 = st.executeQuery("Select \* from user where email\_id='"+s5+"'");

if(rs2.next())

{

jTextField3.setForeground(Color.red);

jTextField3.setText("User with this email id already exists");

rs2.close();

}

else

{

if(s3.equals(s4))

{

PreparedStatement pst=con.prepareStatement("Insert into user values(?,?,?,?)");

pst.setString(1, s1);

pst.setString(2, s2);

pst.setString(3, s3);

pst.setString(4, s5);

int x=pst.executeUpdate();

if(x>0)

{

dispose();

//new NewJFrame1().setVisible(true);

}

}

else

{

jLabel1.setText(" \* the two fiels do not match");

//jPasswordField1.setText("the two password fields do not match");

jLabel1.setVisible(true);

}

}

}

rs1.close();

con.close();

}

catch(Exception e)

{

System.out.println(e);

}

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jPasswordField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField2FocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

jTextField2.setText(jTextField1.getText()+"\_cmc");

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new CreateAccount().setVisible(true);

}

});

}

**Error.java**

public class error extends javax.swing.JFrame {

/\*\* Creates new form error \*/

public error() {

initComponents();

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

jLabel1.setIcon(new javax.swing.ImageIcon("G:\\myfolder\\my projects(java)\\editor\\exit.gif")); // NOI18N

jLabel1.setText("YOU HAVE NOT SELECTED YOUR PROJECT TYPE");

jButton1.setText("O.K.");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addGap(165, 165, 165)

.addComponent(jButton1)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(76, 76, 76)

.addComponent(jLabel1)

.addGap(50, 50, 50)

.addComponent(jButton1)

.addContainerGap(127, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

setVisible(false);

new NewProject().setVisible(true);

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new error().setVisible(true);

}

});

}

**Response.java**

package ide;

import ide.\*;

import java.sql.\*;

/\*\*

\*

\* @author GURJEET

\*/

public class Responses extends javax.swing.JFrame {

/\*\* Creates new form Responses \*/

public Responses() {

initComponents();

}

/\*\* This method is called from within the constructor to

\* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jLabel1 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setFont(new java.awt.Font("AR ESSENCE", 1, 18));

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18));

jLabel1.setText("Response to your Query");

jButton1.setText("Check");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Back to Main");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 204, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 381, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(149, 149, 149)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton2)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addContainerGap())

);

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con=null;

Statement st;

ResultSet rs;

try

{

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

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/IDE","root","123456");

st=con.createStatement();

rs=st.executeQuery("Select \* from replytouser where username='"+new front().username+"'");

if(rs.next())

{

String n = rs.getString("reply");

jTextArea1.setText(n);

}rs.close();

con.close();

}

catch(Exception e)

{

}

//dispose(); // TODO add your handling code here:

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

dispose();

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Responses().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

// End of variables declaration

}

**B.1 User Authentication**

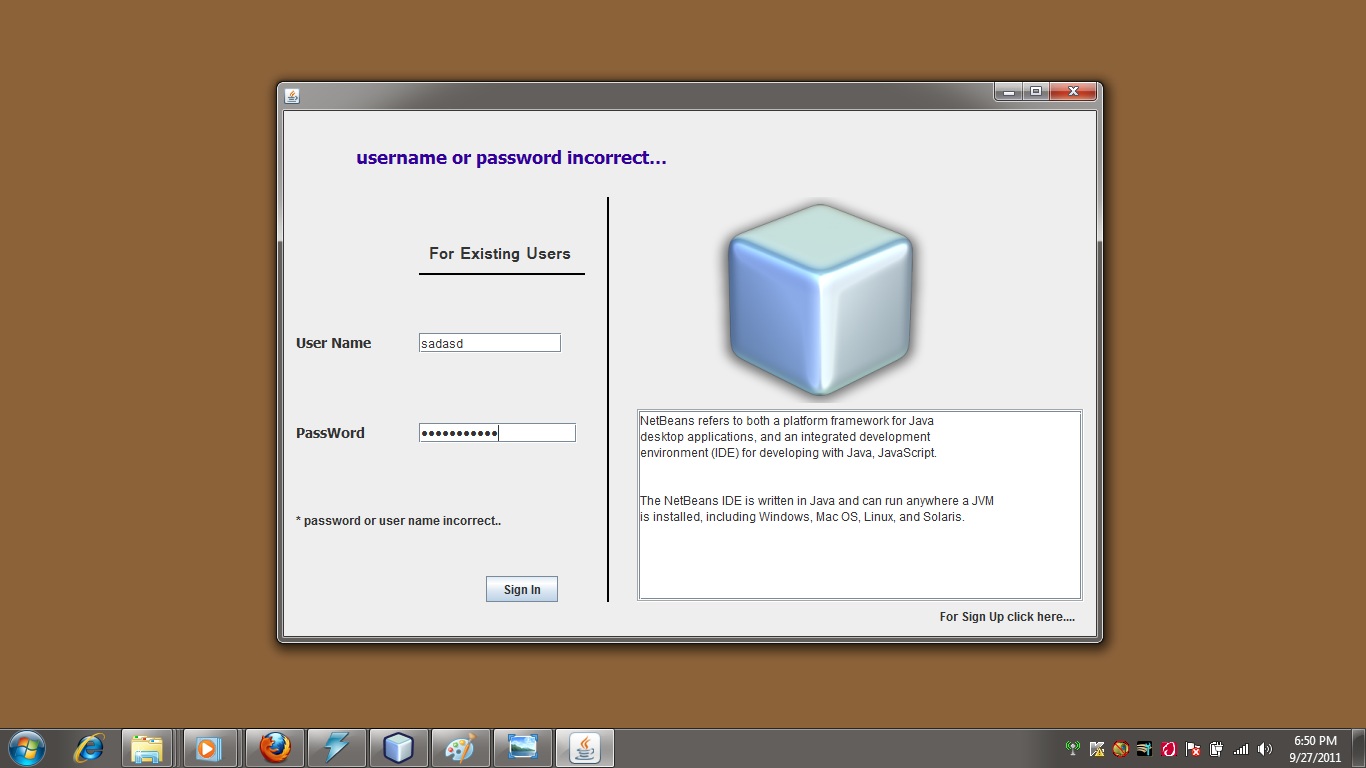
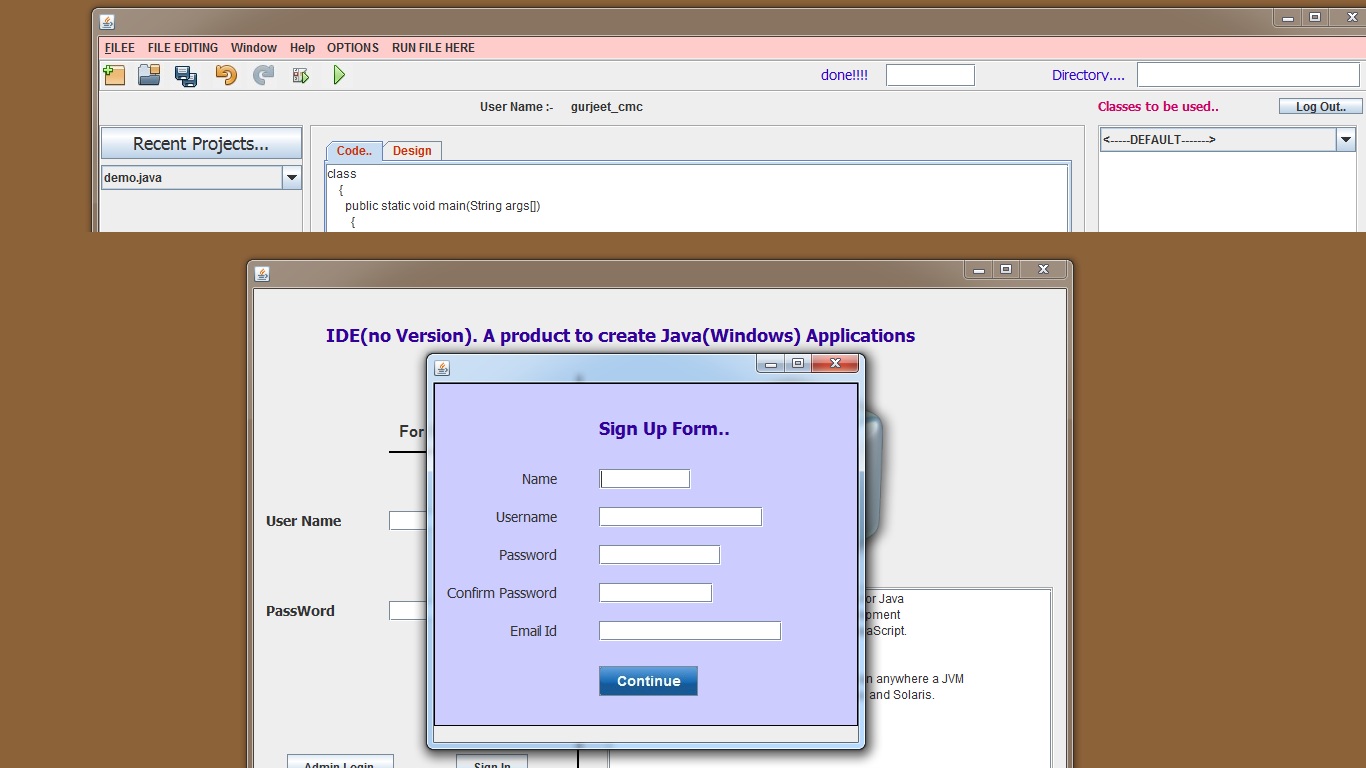
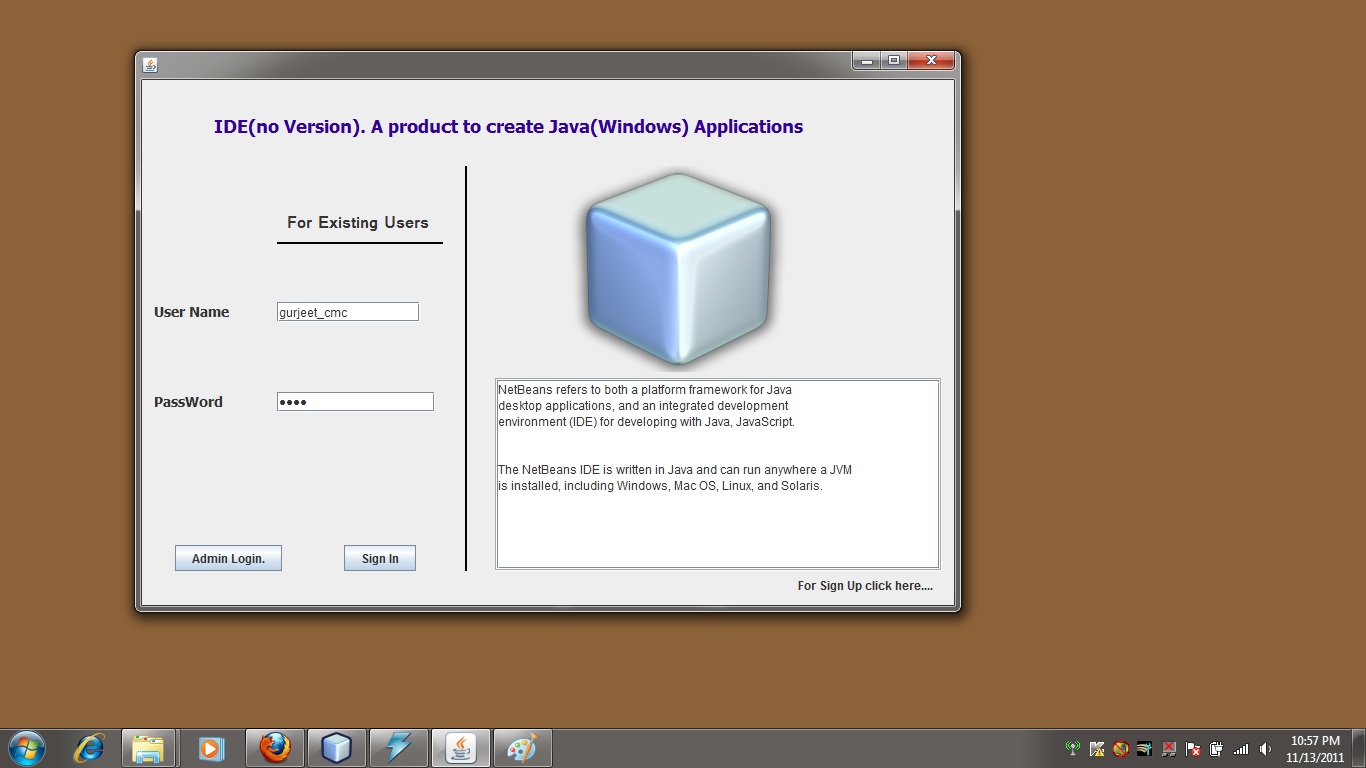
****

Fig :- 2.4 Authentication

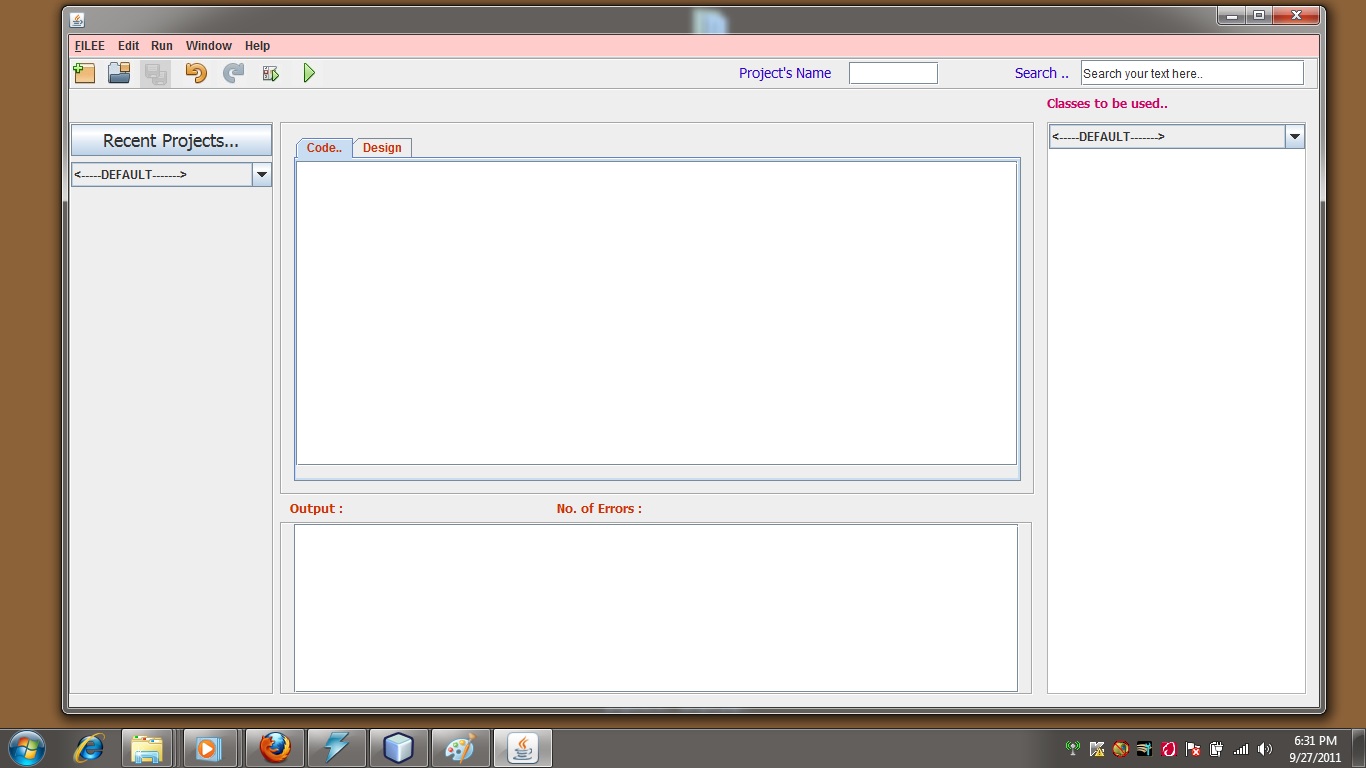
**B.2Sign-Up Form**

****

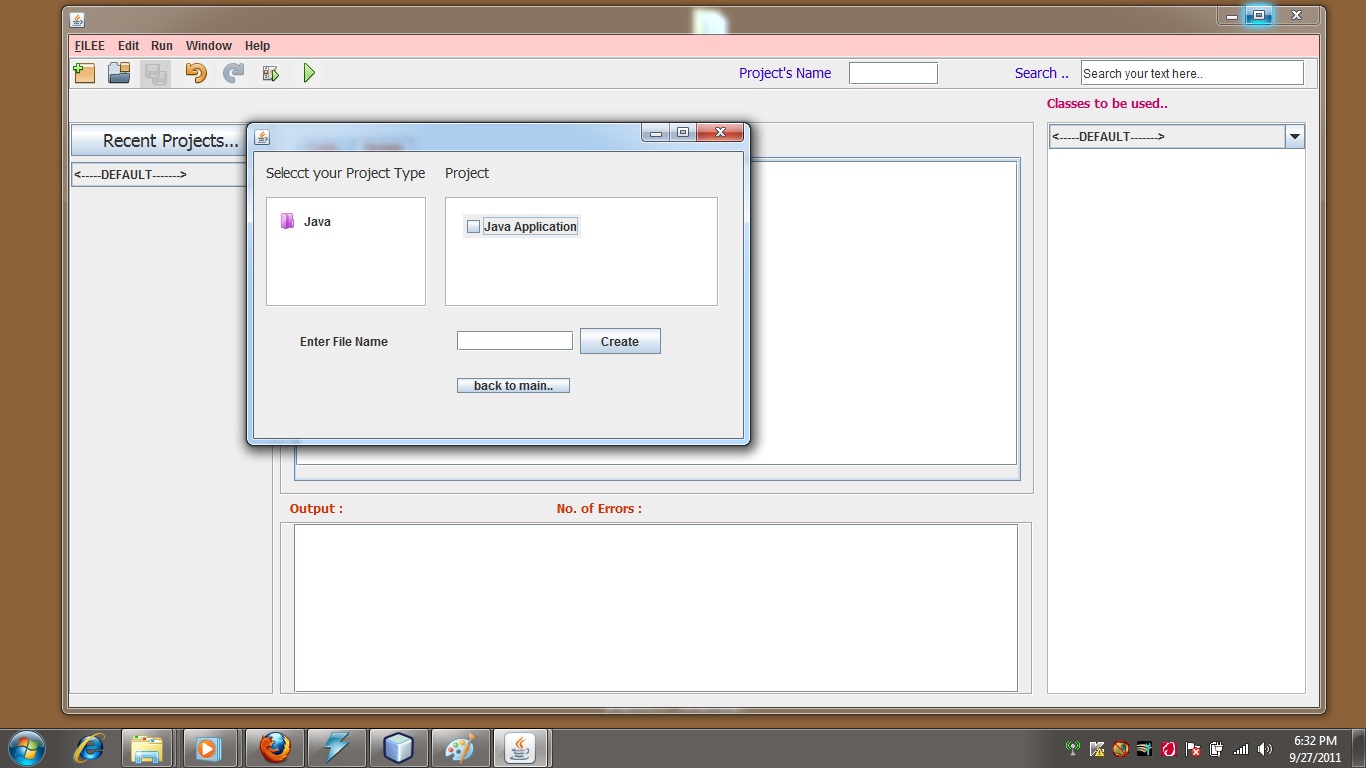
****

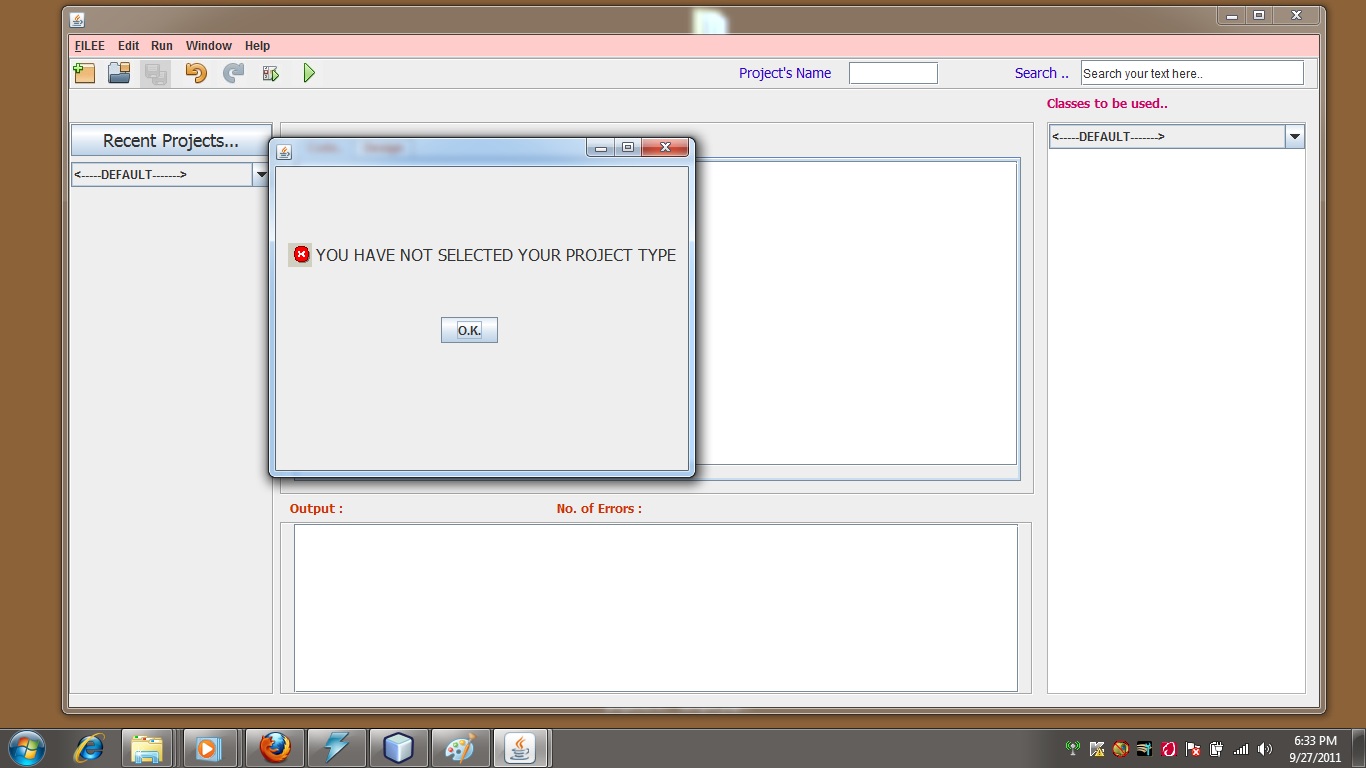
**B.3 Main Interface**

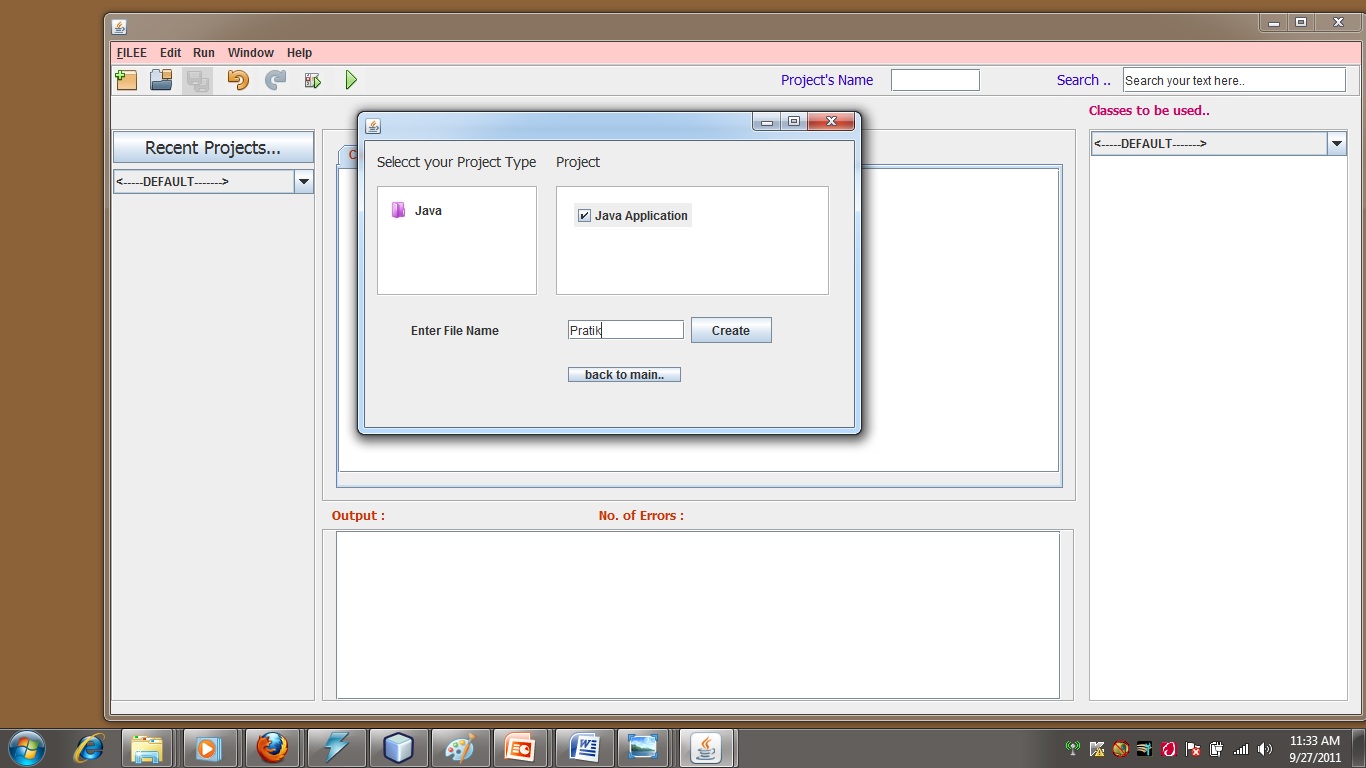
This is the main interface where user can do writing compiling etc. More over recent projects provision has been enabled to give the view of recent projects to the users.

****

**B.4 Creating a new project**



****

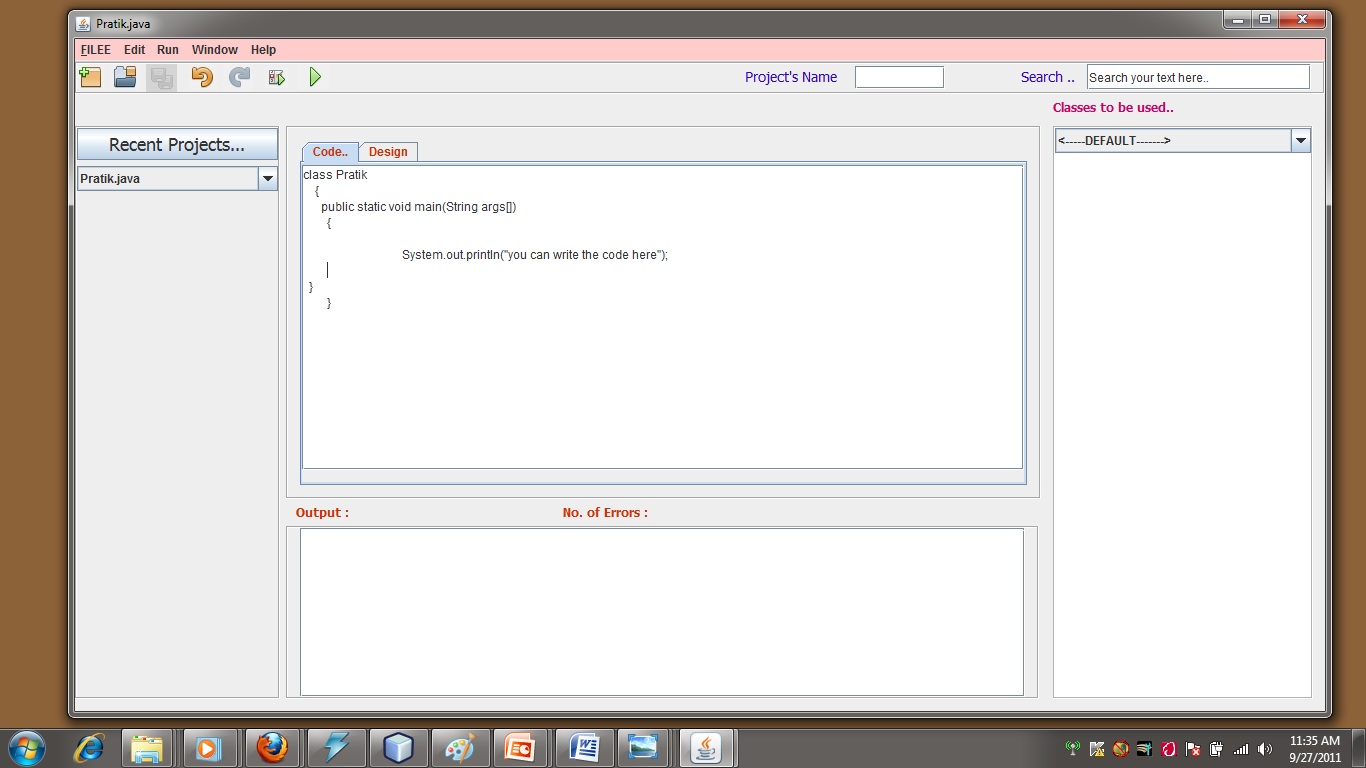
****

**B.5 Write Code here**

A textbox has been provided where user can write his code with the help of GUI ( Graphical User Interfaces ).

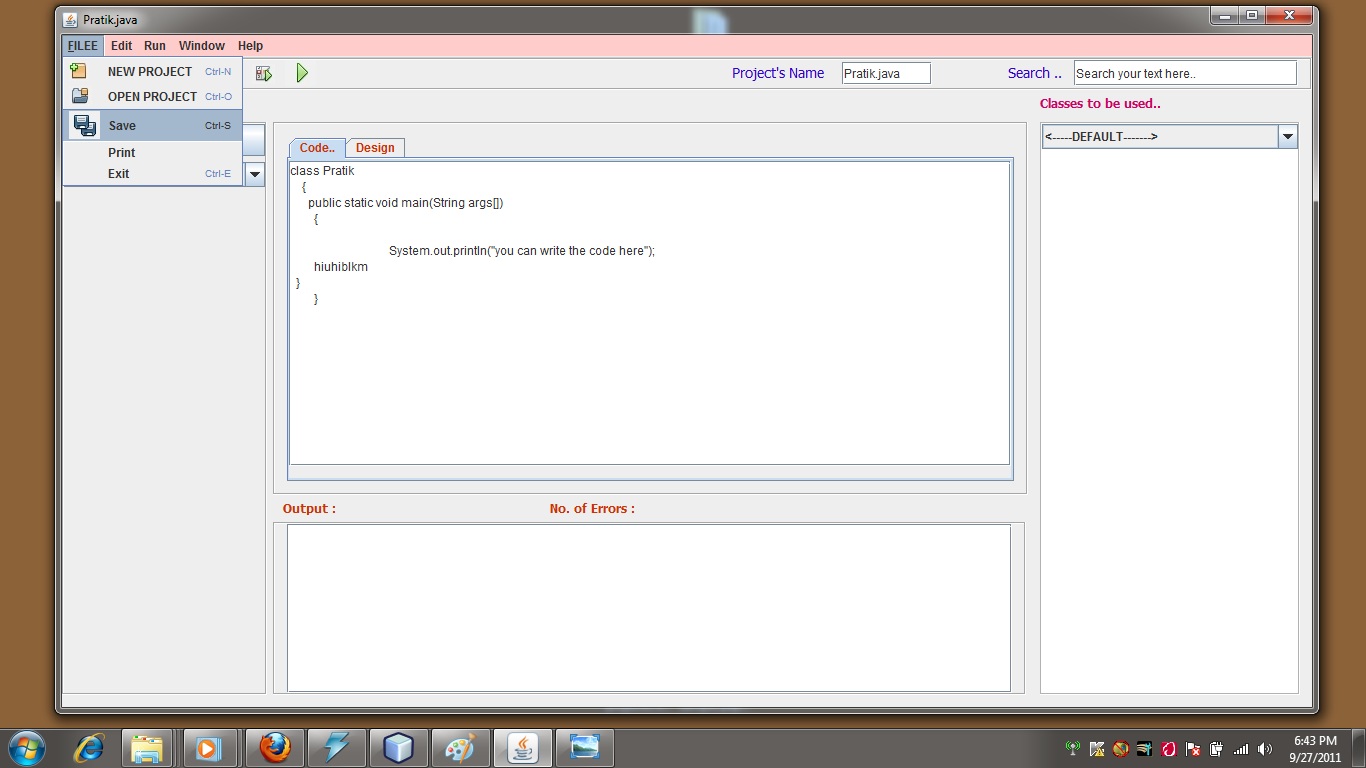
After the code has been written, user can compile the program. An another textbox is provided where the errors are going to be displayed after compilation. If the compilation has been successful then a corresponding .class file will be generated in the default directory of the system.

After Successful compilation of the code, Interpreter has been called to execute the .class file generated by the compiler previously and correspondingly the output is displayed in the text area provided.

****

**B.6 Save the Project**

You can save your program at the backend also rather than storing in the system directories. That is mysql has been used for the purpose that keeps record of all the files that are generated with our application.

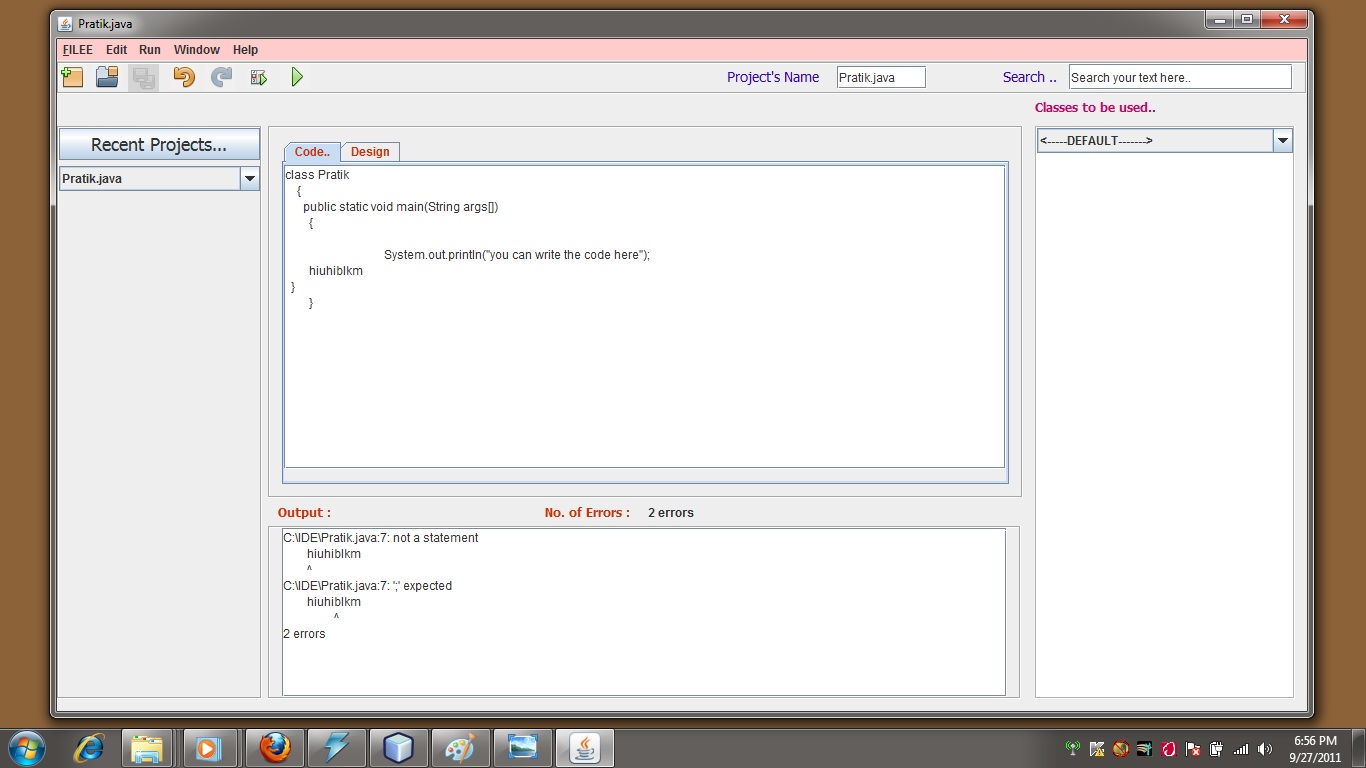
****

**B.7 Code Compilation**

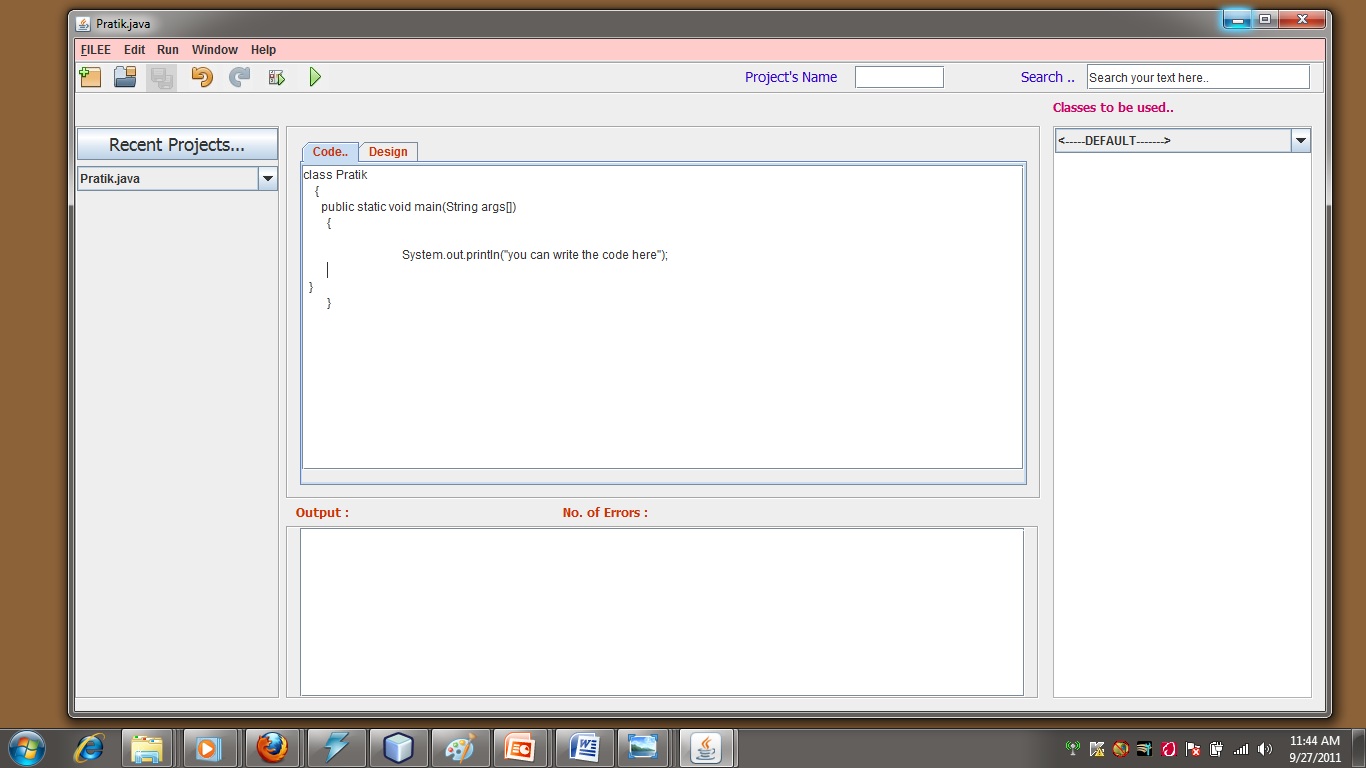
In Java, programs are not compiled into executable files. They are compiled into Bytecode which the JVM then executes at runtime. Java source code is compiled into bytecode when we use the javac compiler. The bytecode gets saved on the disk with the file extension .class. When the program is to be run, the bytecode is converted, using the Just-In-Time(JIT) compiler. The result is machine code which is then fed to the memory and is executed.

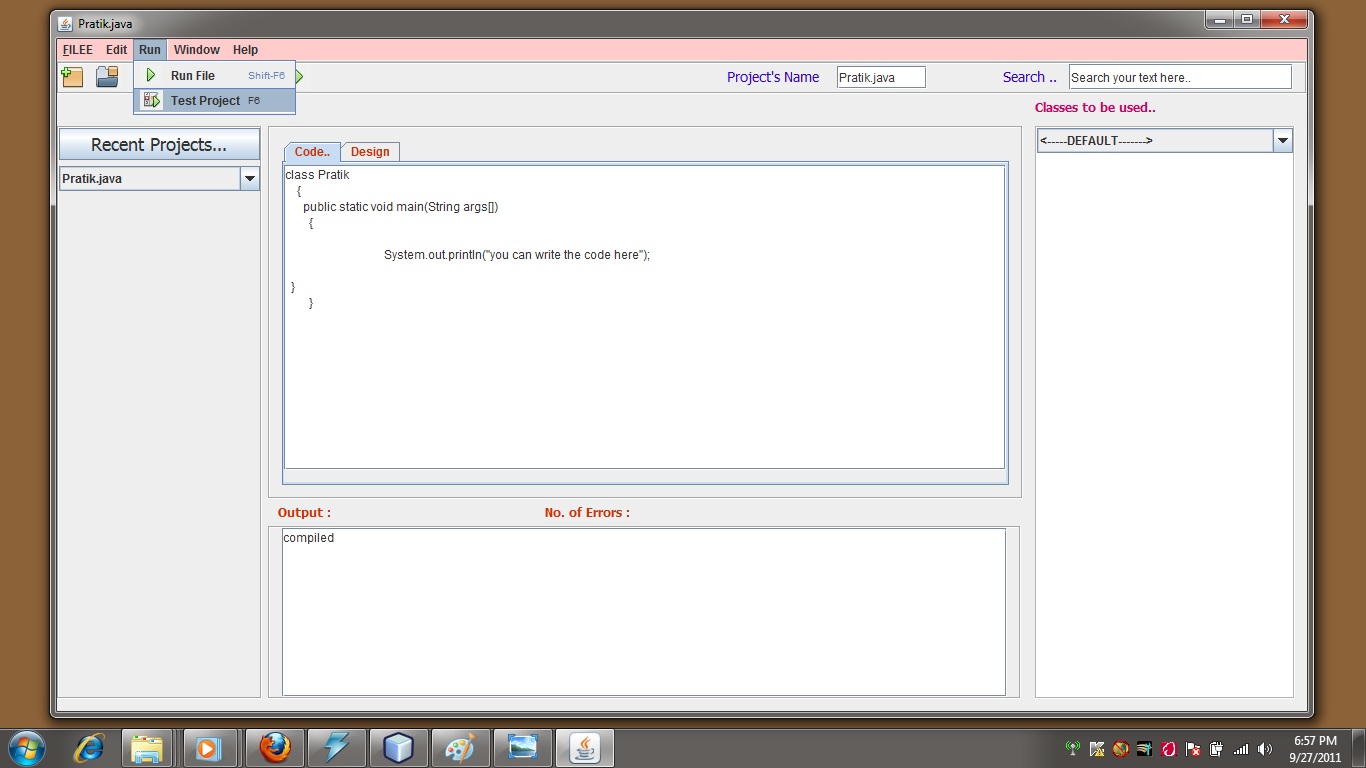
So Java has four step compilation:

* compiler will check the syntactical error
* create byte-code
* create a blank \*.class file
* merge that byte code to that blank \*.class file

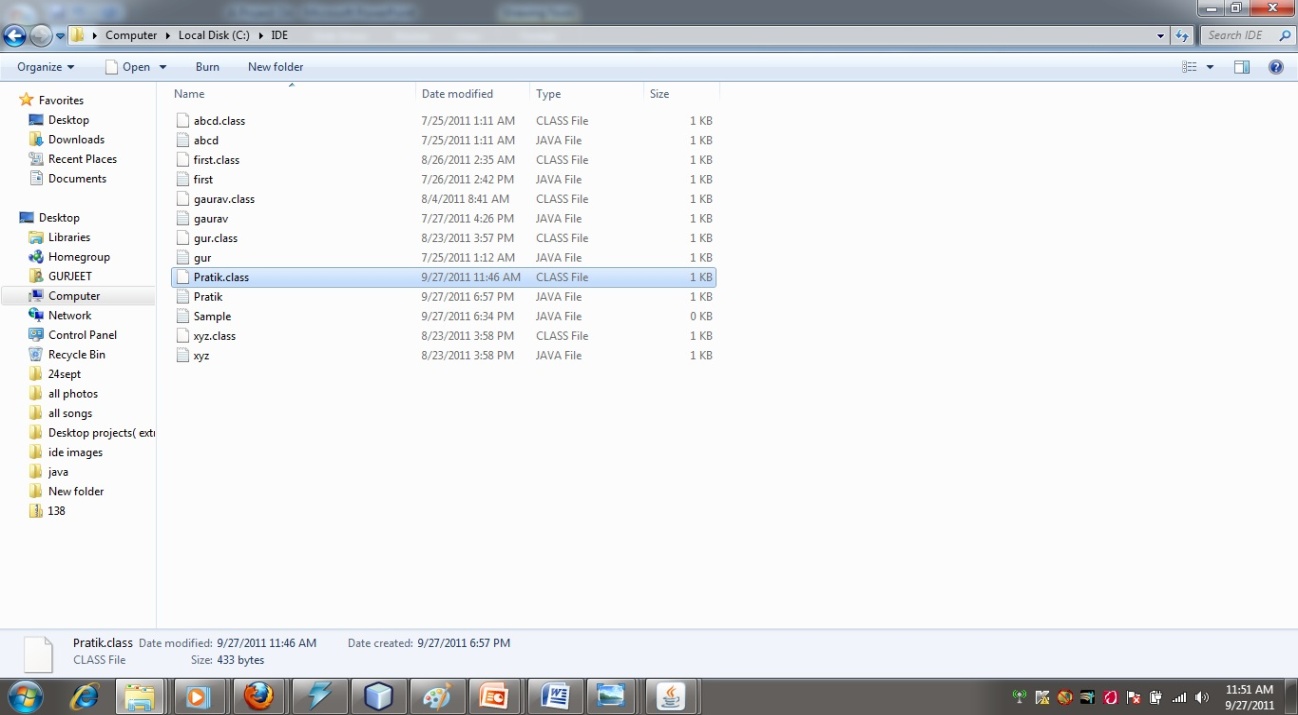
****

**B.8 Error Removing**

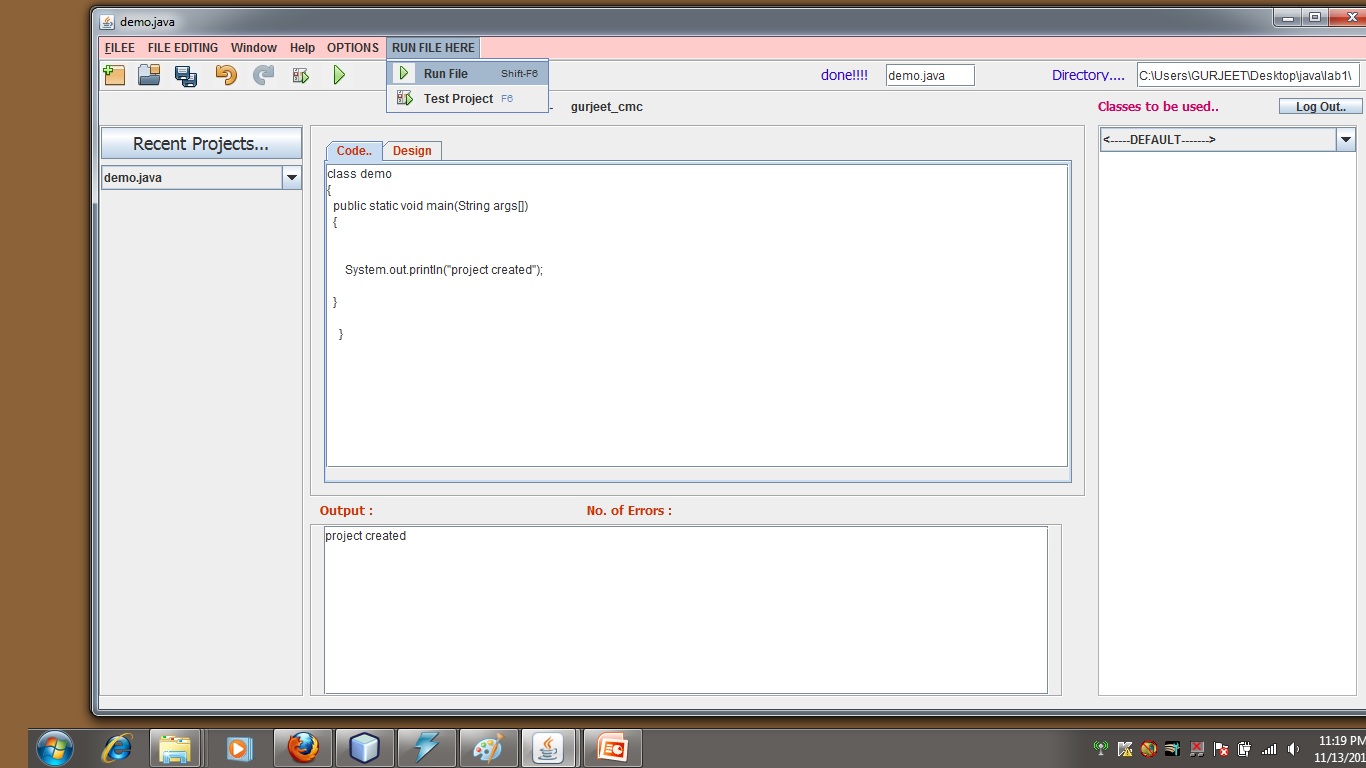
****

****

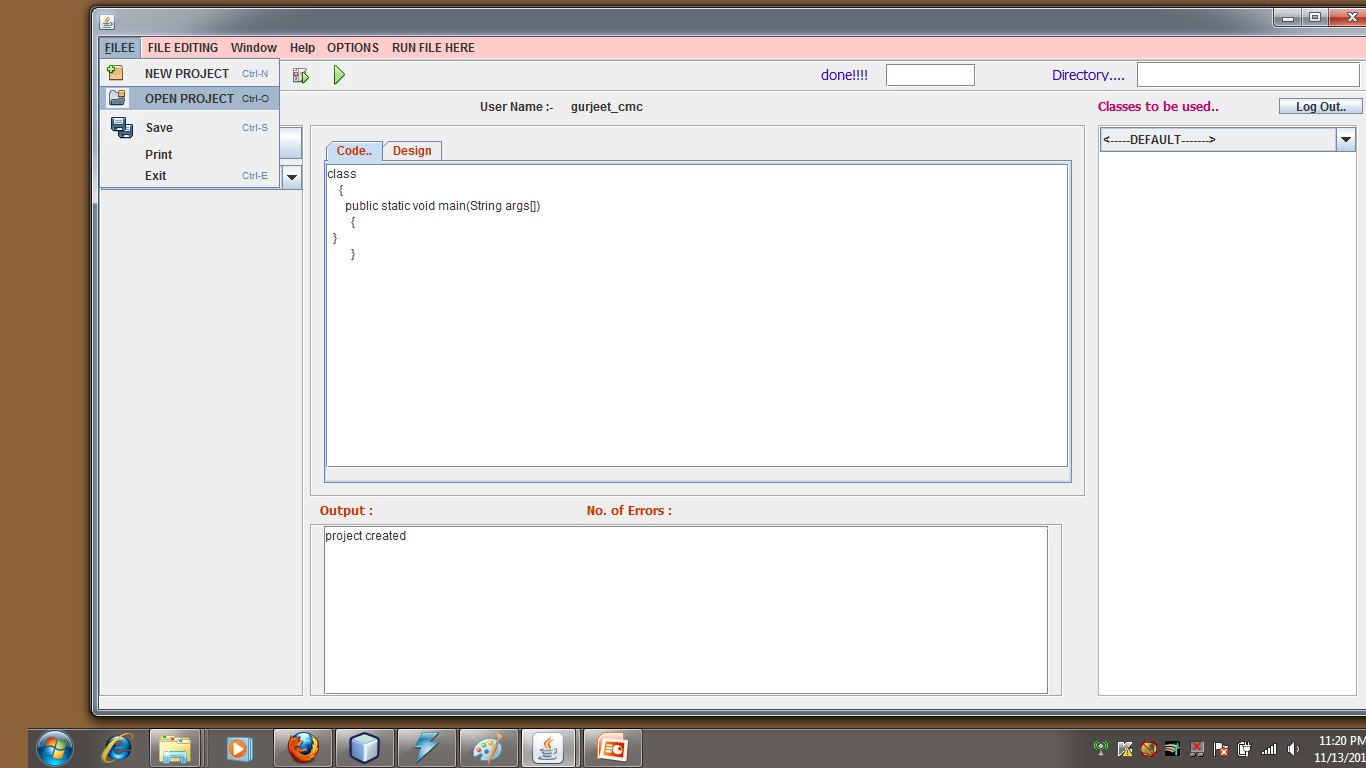
**B.9 Generation of byte code( .class file )**

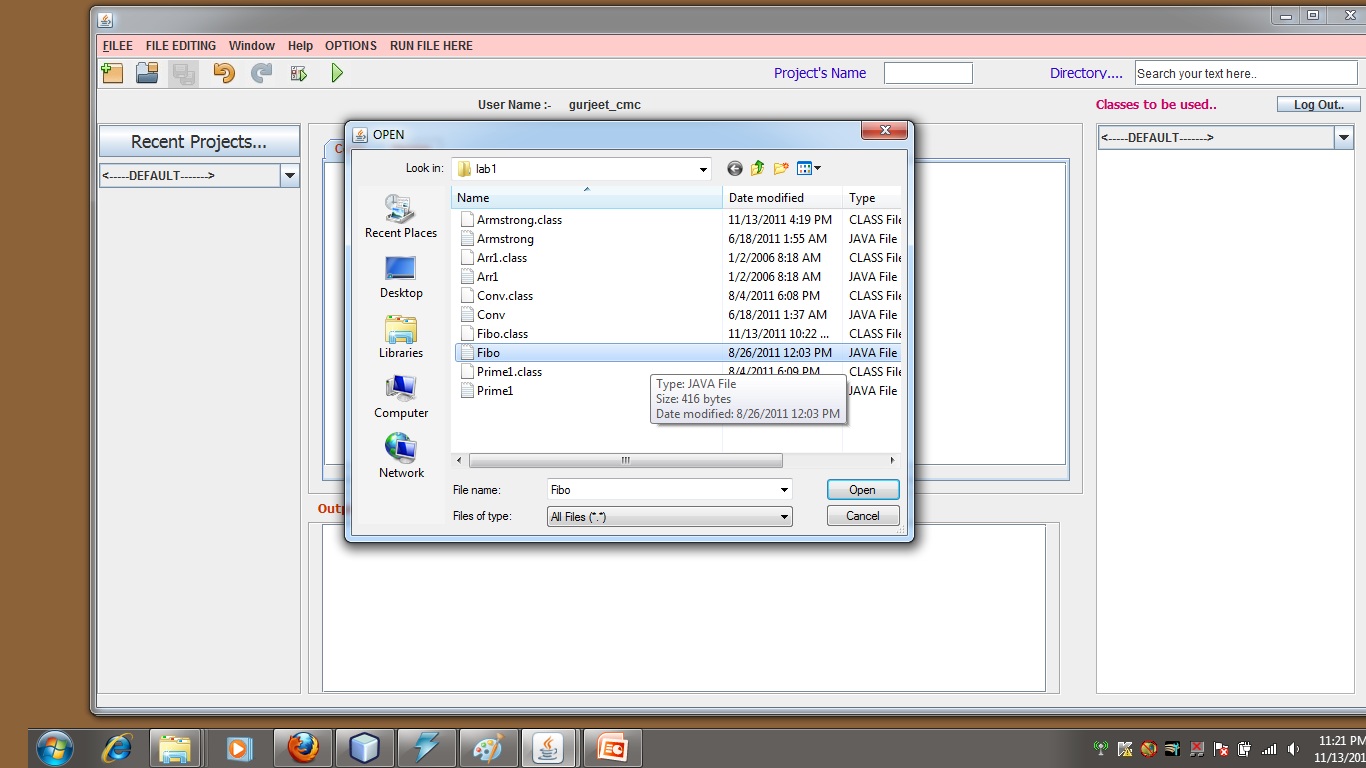
****

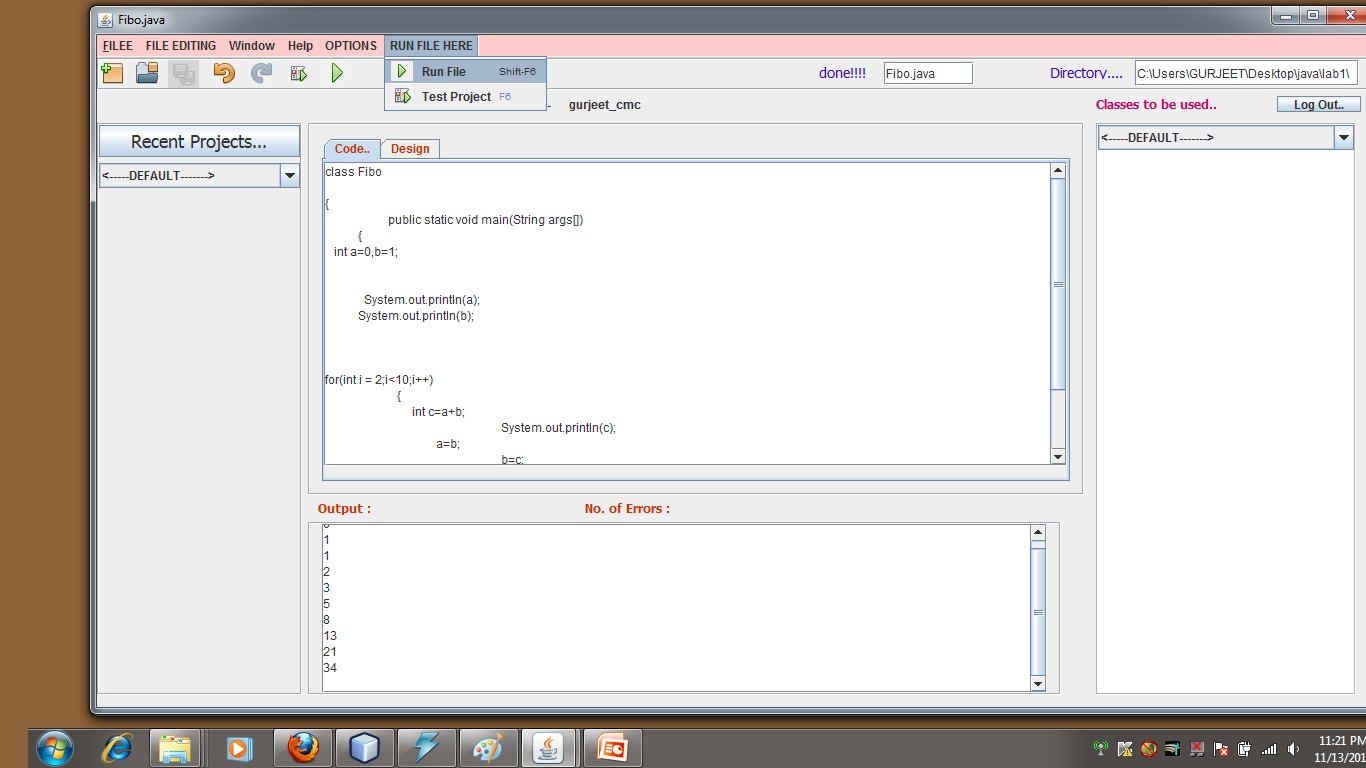
**B.10 Interpreting**

****

**B.11Opening Another Project**

****

****

****

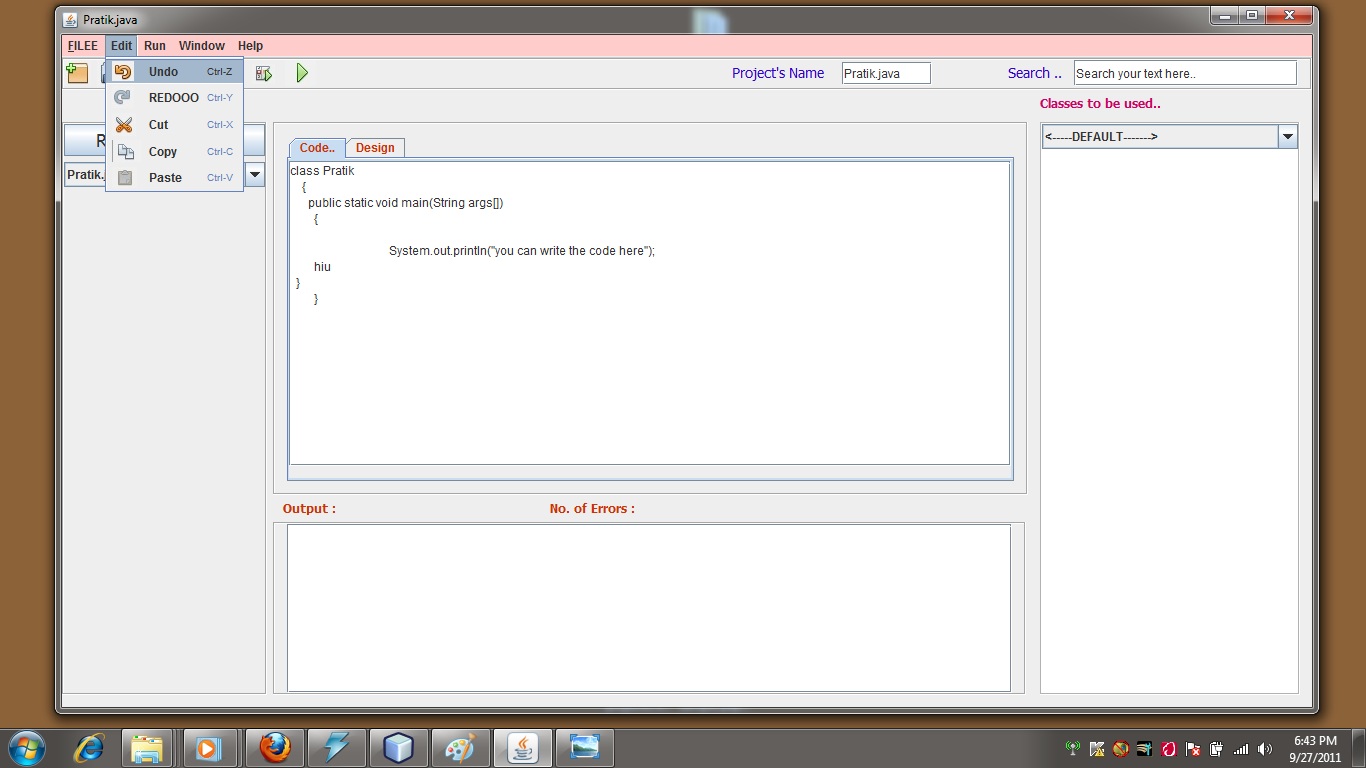
**B.12 Text Manipulation**

**B.12.1 Undo**

It erases the last change done to the document reverting it to an older state. In some more advanced programs such as graphic processing, undo will negate the last command done to the file being edited.

The opposite of undo is **redo**. The redo command reverses the undo or advances the buffer to a more current state.

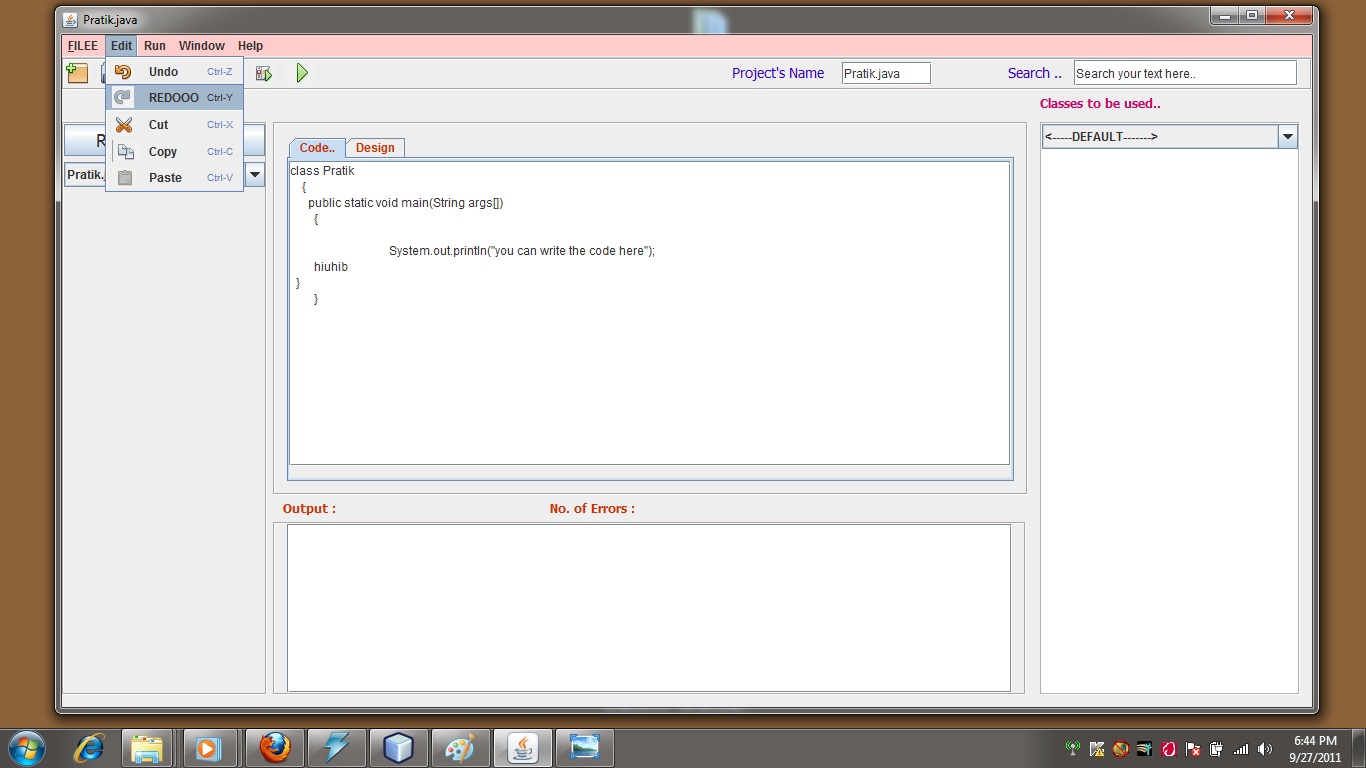
In most Windows applications, the Undo command is activated by pressing Ctrl+Z or Alt+Backspace.



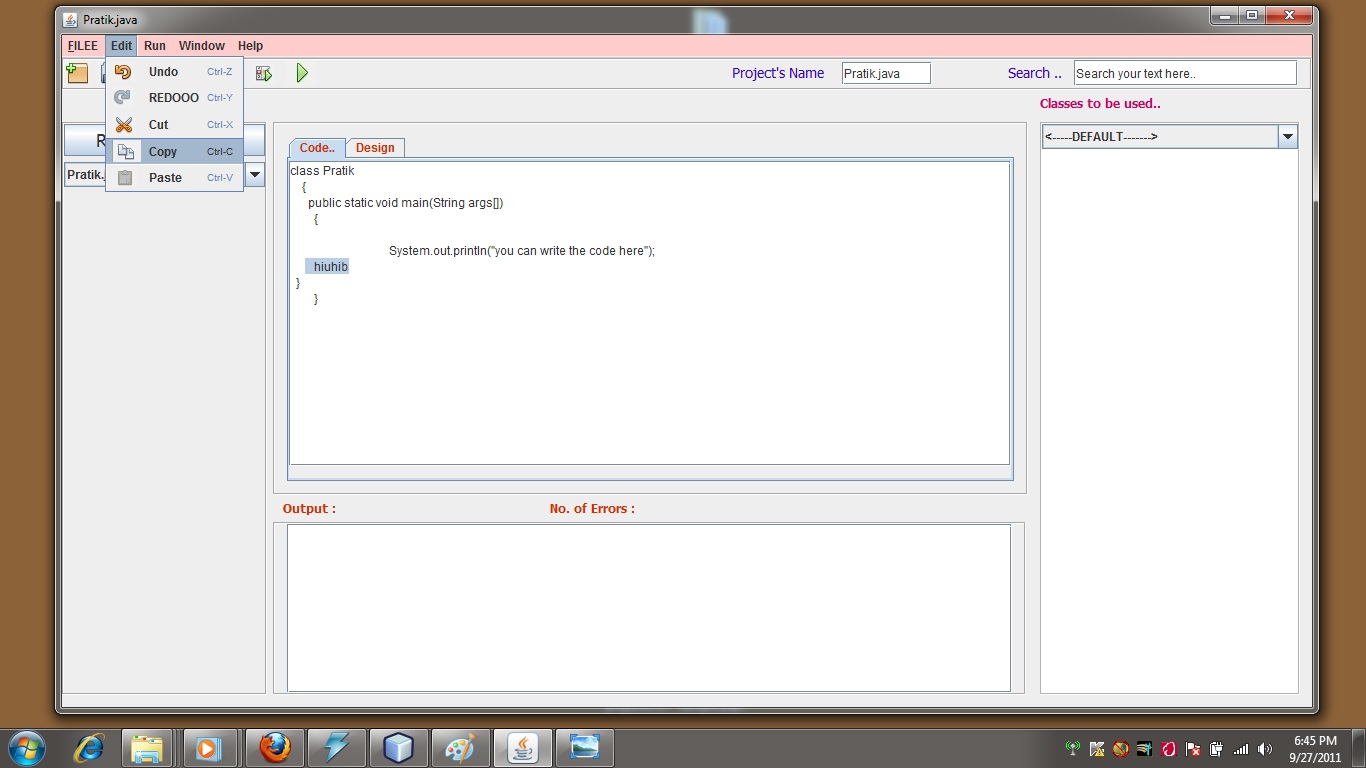
**B.12.2 Redo**

Redo reverts the effects of the undo action. The simplest form of redo is flip-undo, in which using undo after undoing redoes the undone action. In this case, the program flips back and forth between two states when the undo button is pressed.

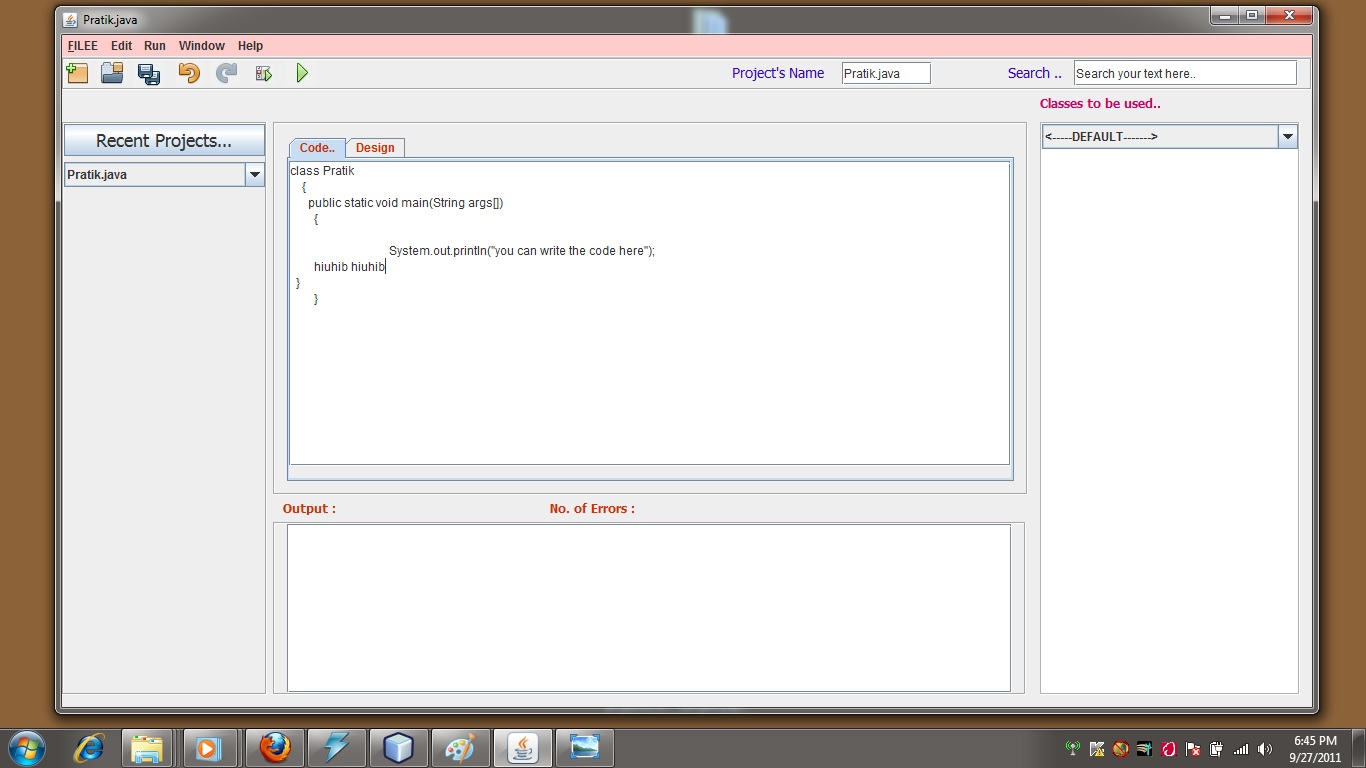
In a more typical redo model, there are separate undo and redo buttons. The redo can be used for each undo action performed. Making a new edit usually clears the redo list. If a branching redo model is used, the new edit branches the action history.



**B.12.3 Copy**



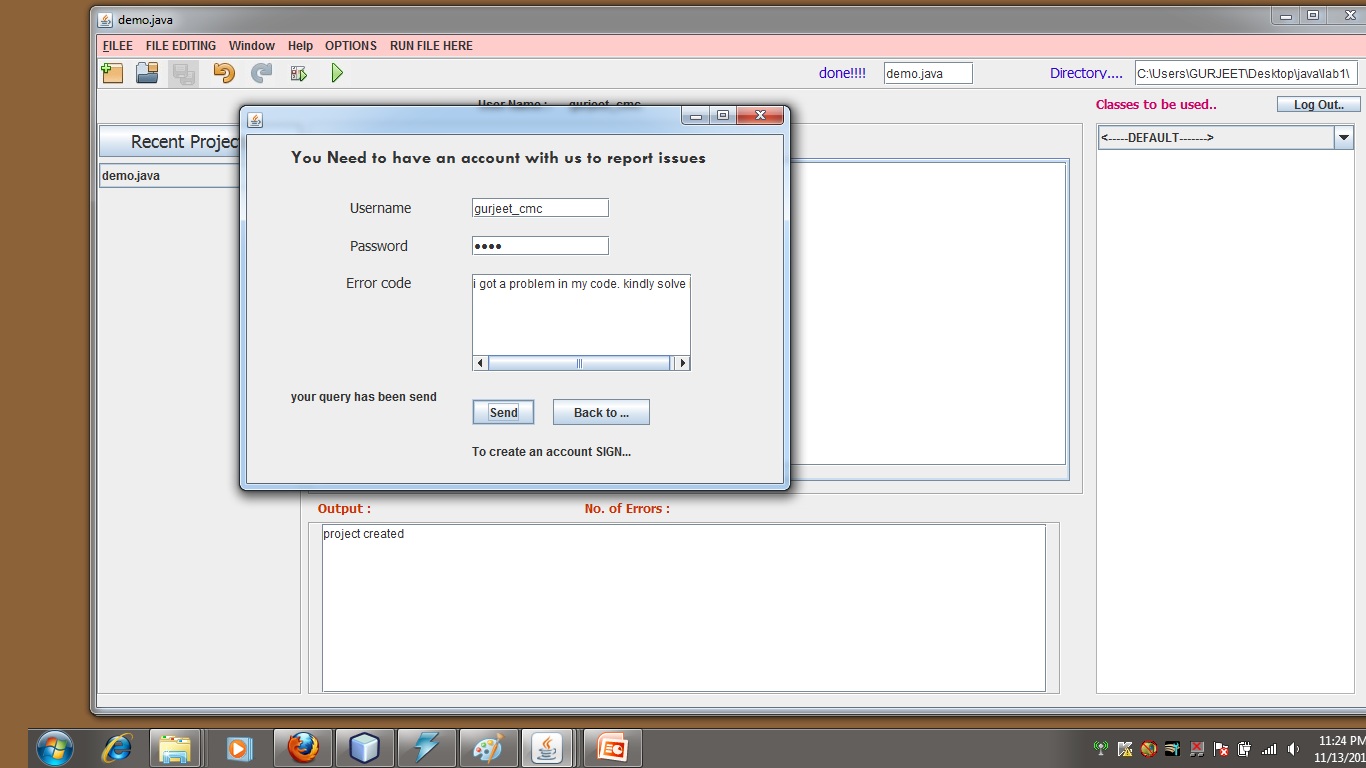
**B.12.4 Paste**

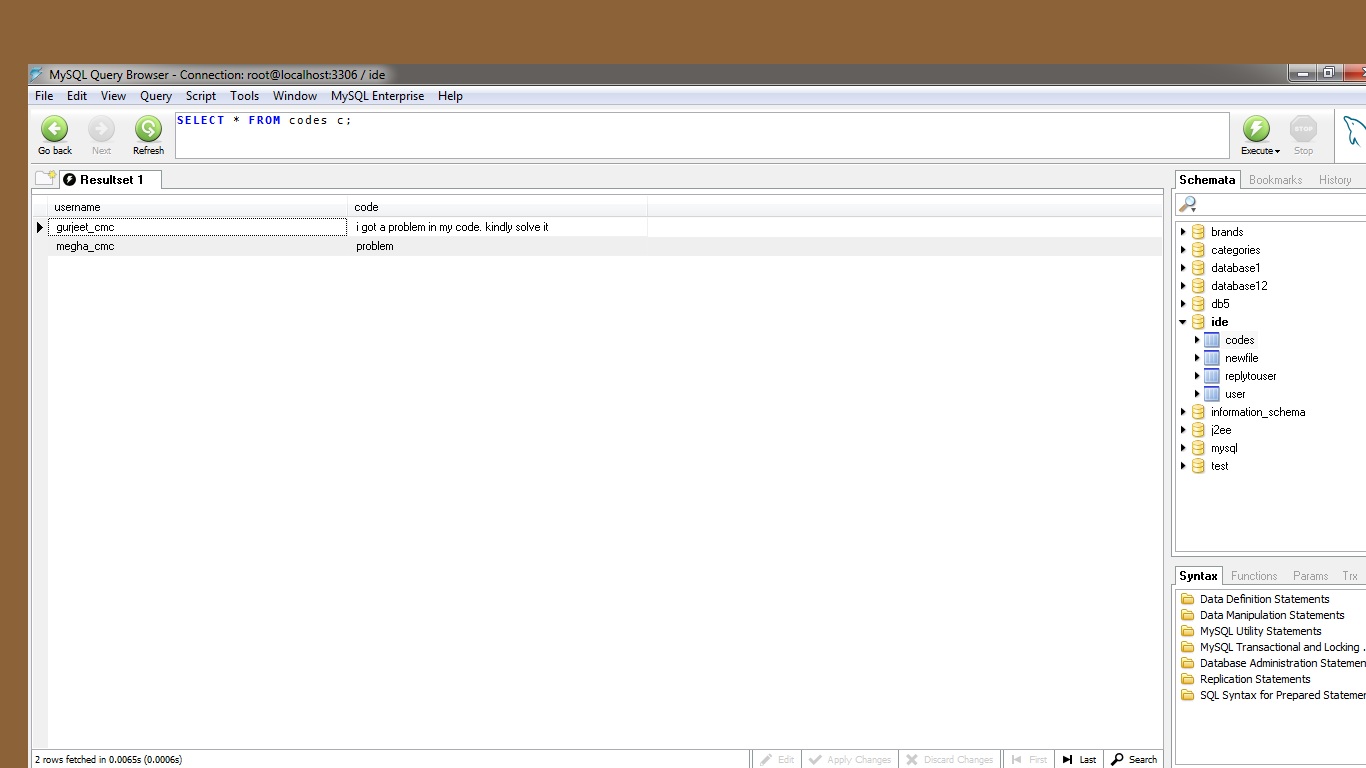
****

**B.13 Short-Cuts Used**

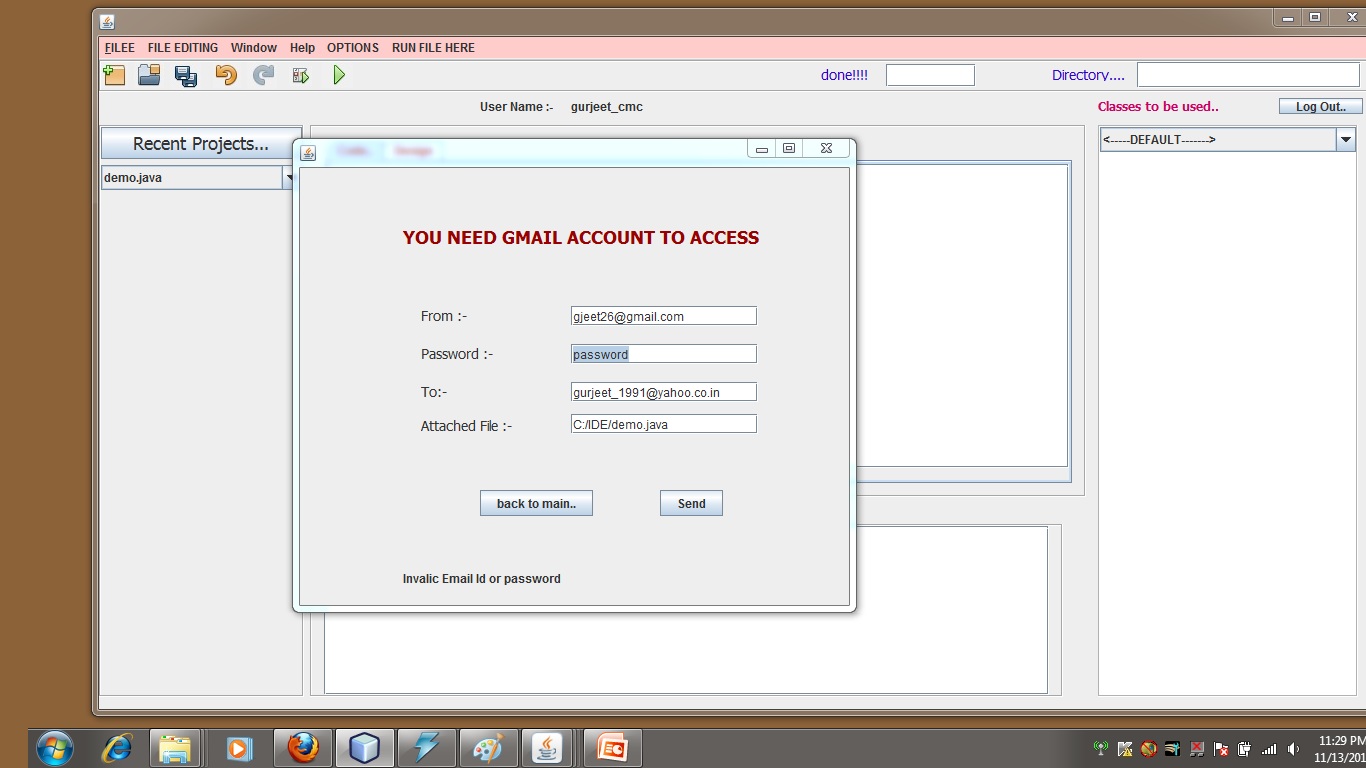
****

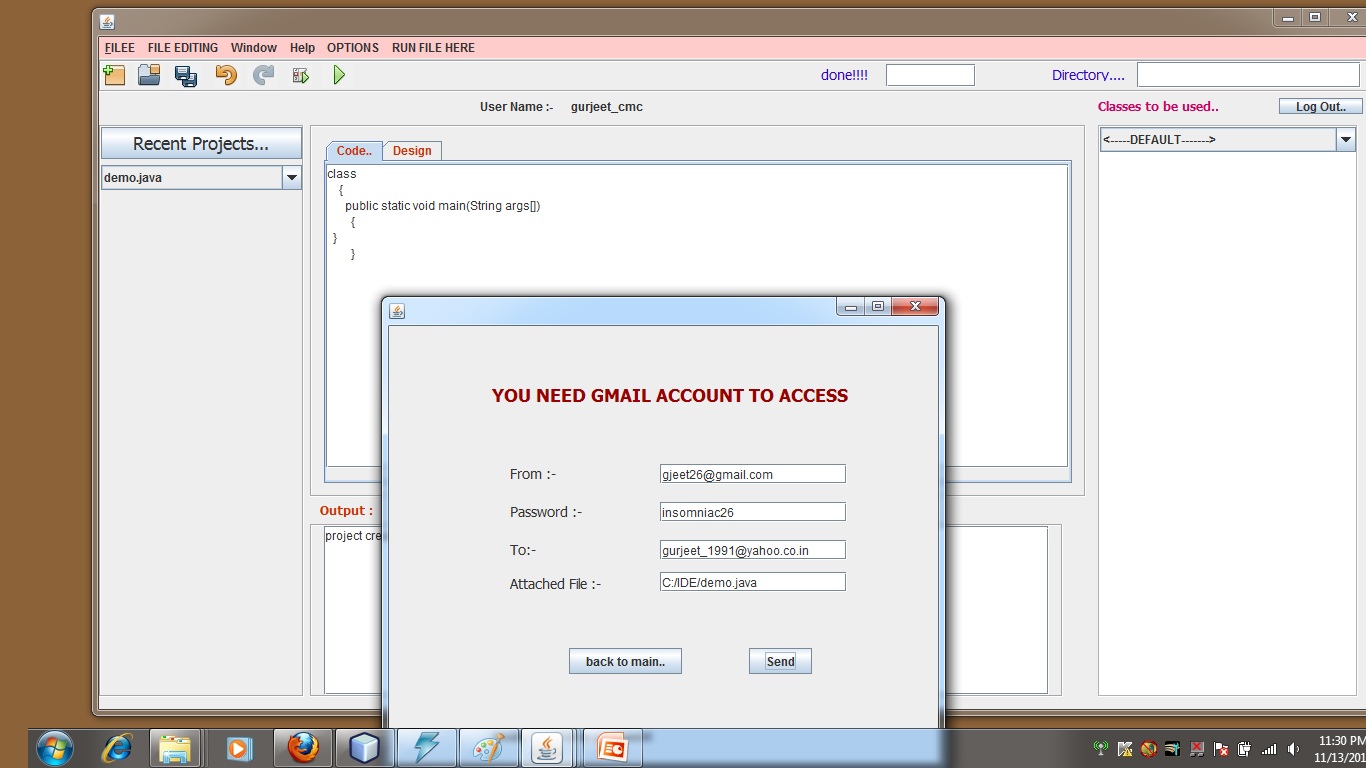
**B.14 Report Issue**

****

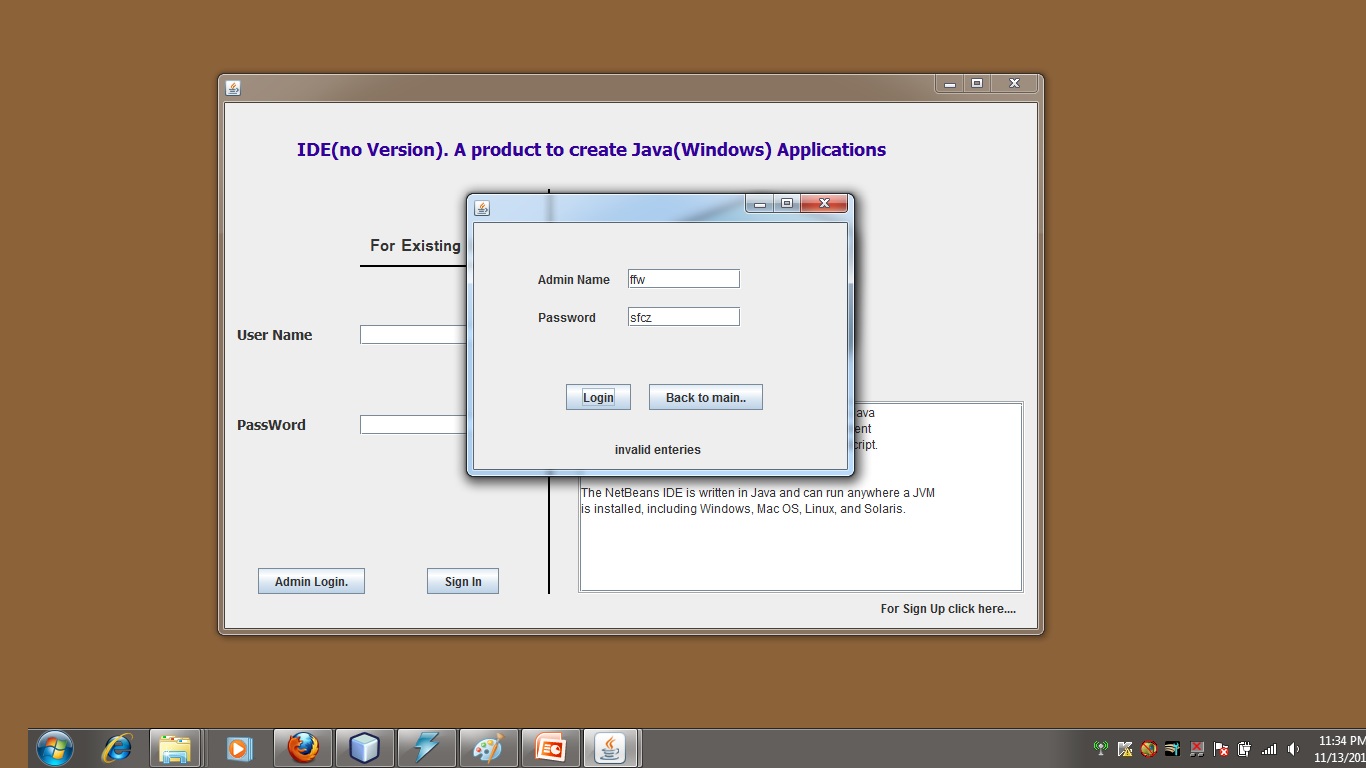
****

**B.15 Sending Code Via Web**

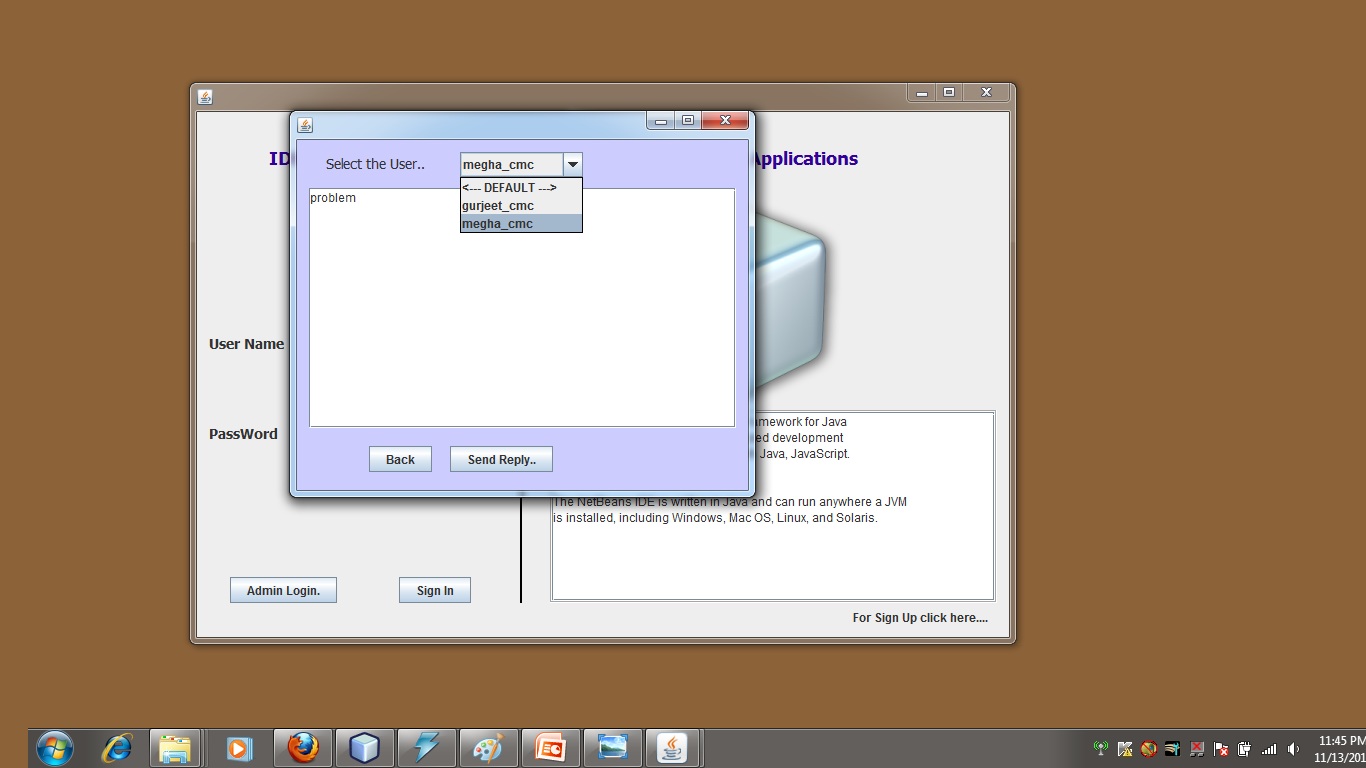
****

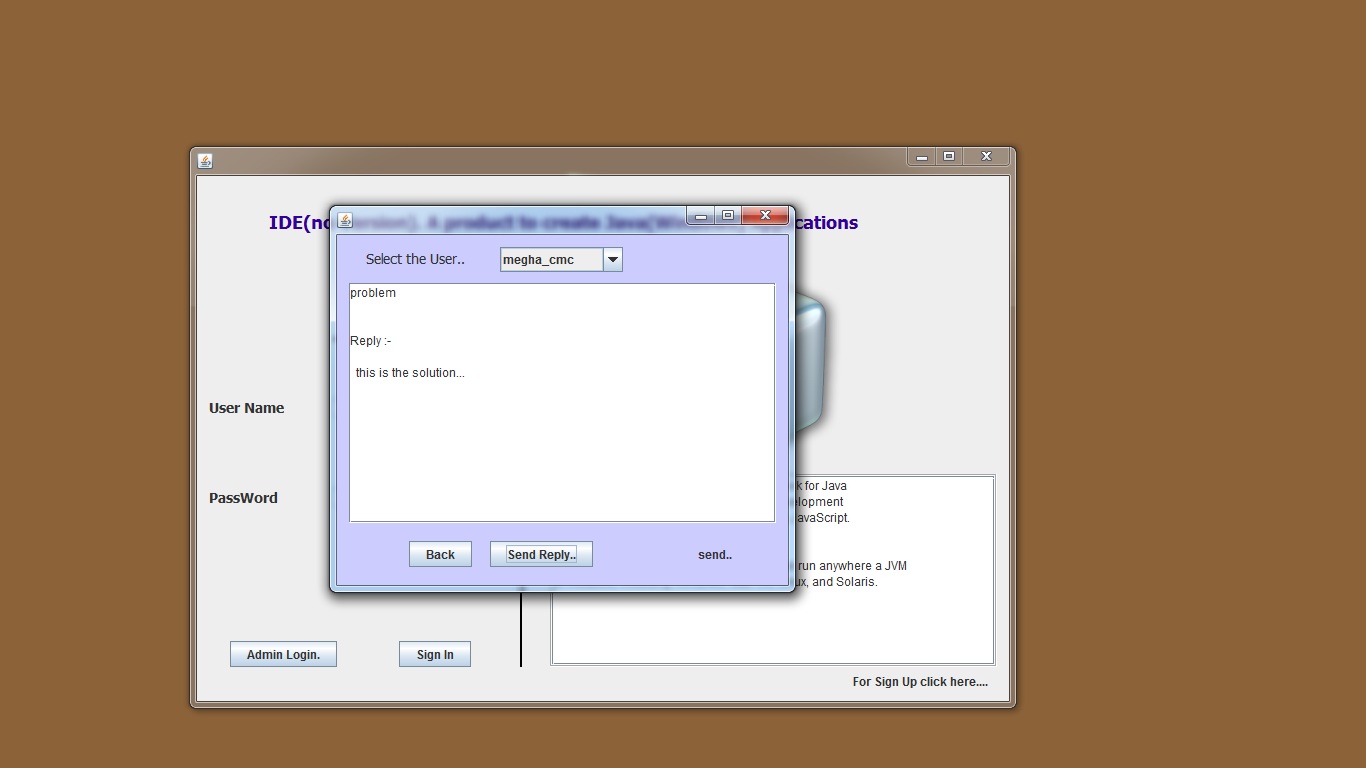
****

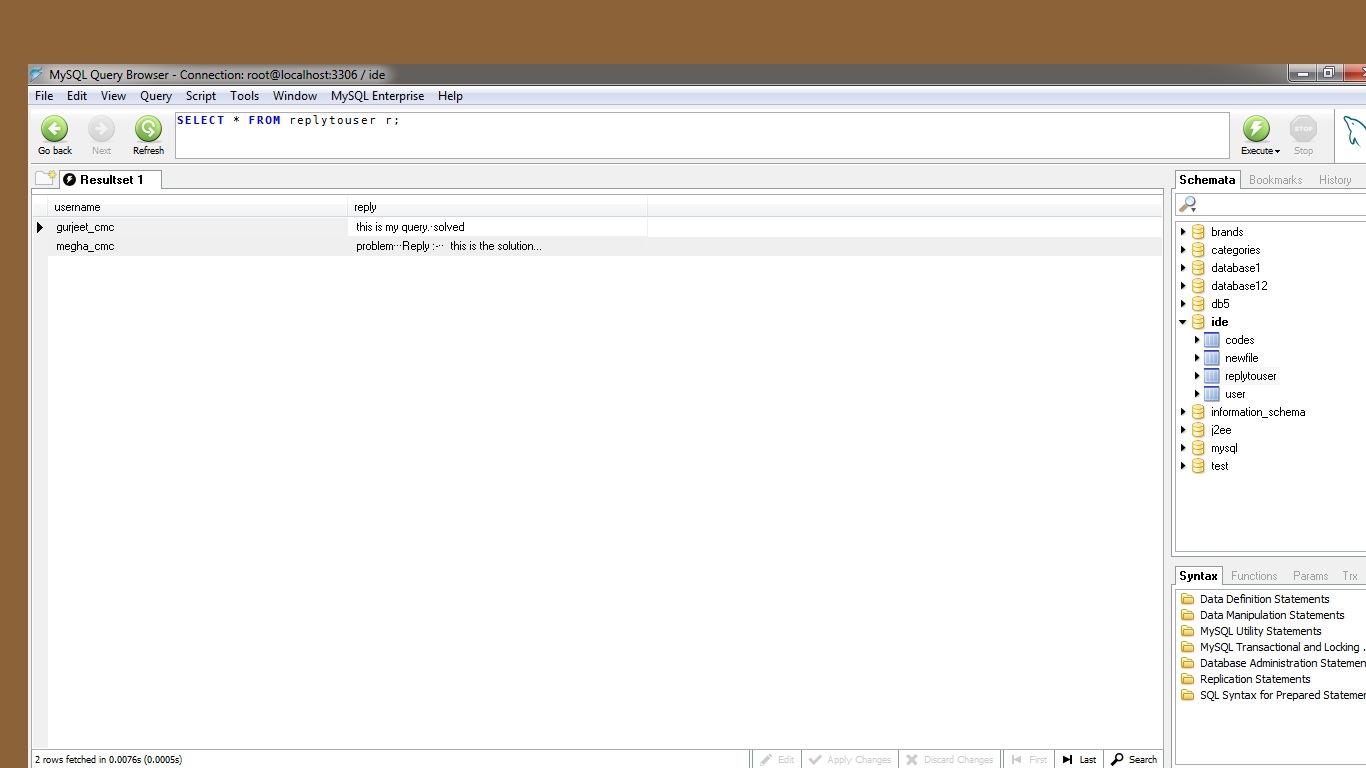
**B.16 Administrator Login Screen**

****

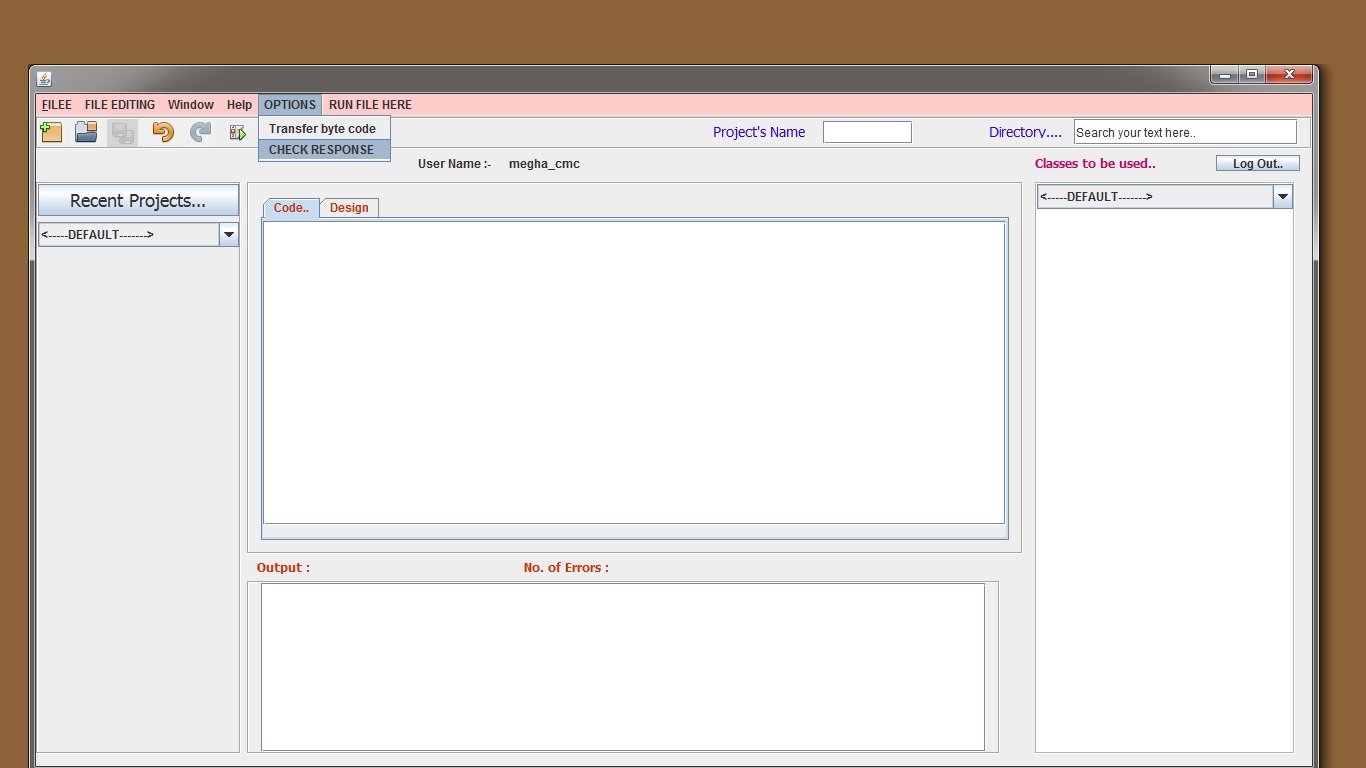
**B.17 Handling Query Screen**

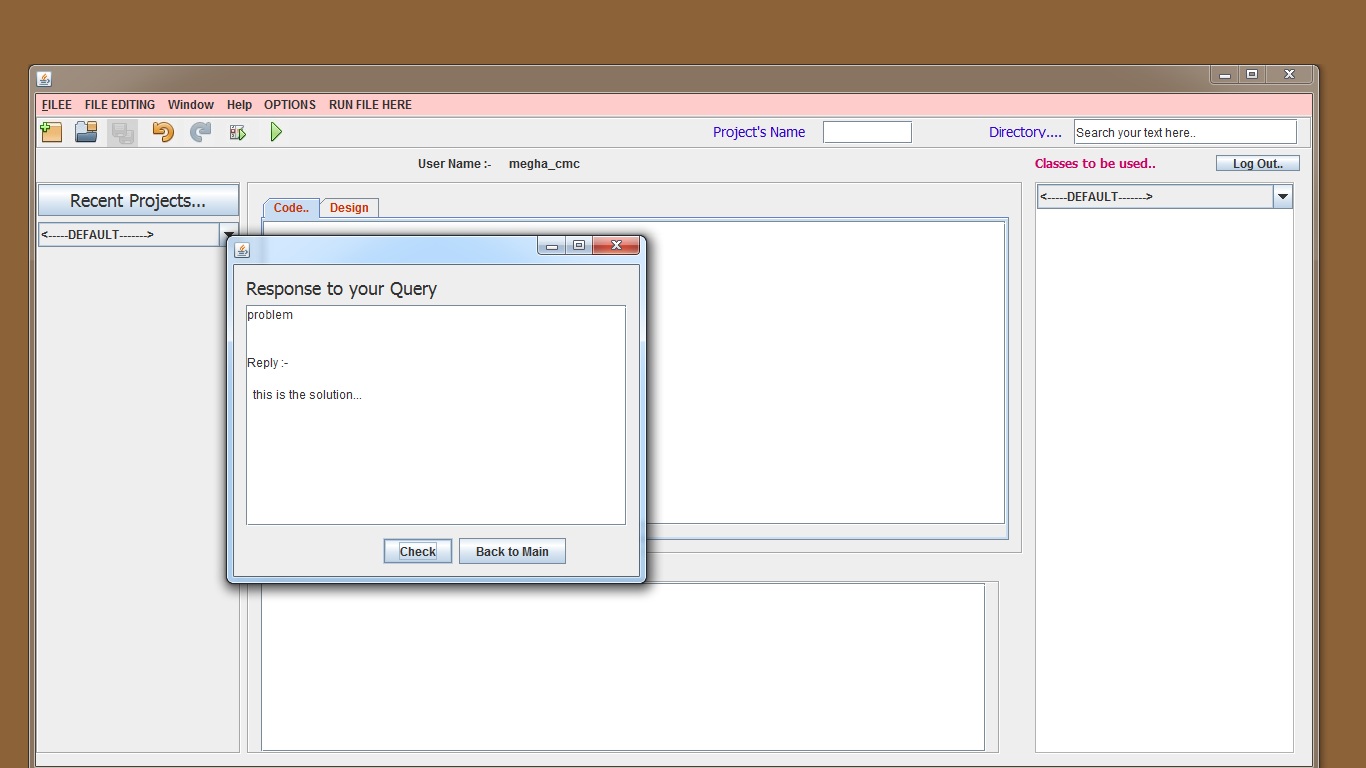
****

****

****

**B.18 Checking Reply At User Side**

****

****