Institute of Computer Technology

ACADEMIC PROJECT (3rd Sem.)

Subject:- OOP

PROJECT NAME:-

FileGuard

Submitted by:-

Vyom Modi

En.no:- 23162171009

&

Divyaraj Parmar

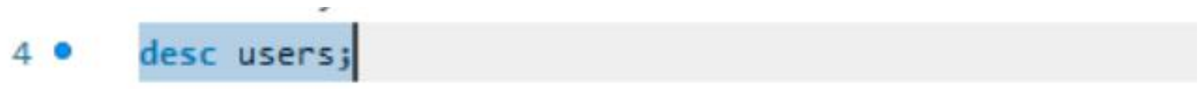
En.no:-23162171013

Submitted to:-

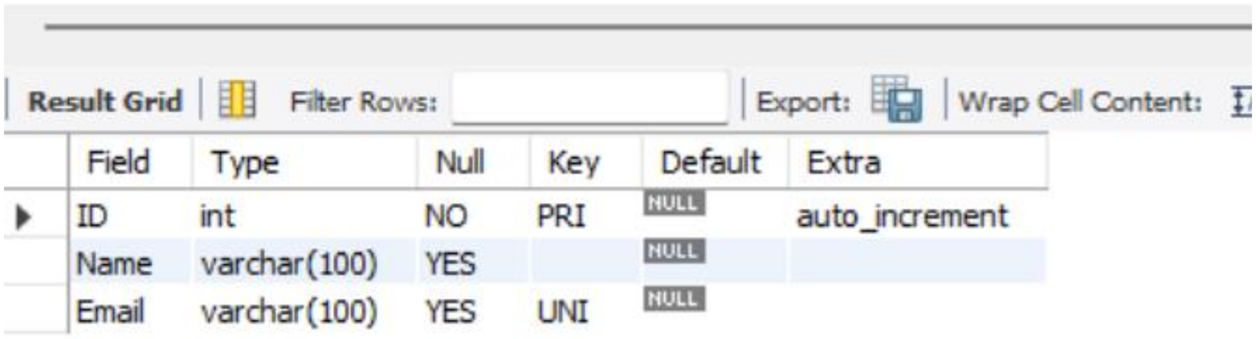
Santanu Sasmal Sir

My sql:-

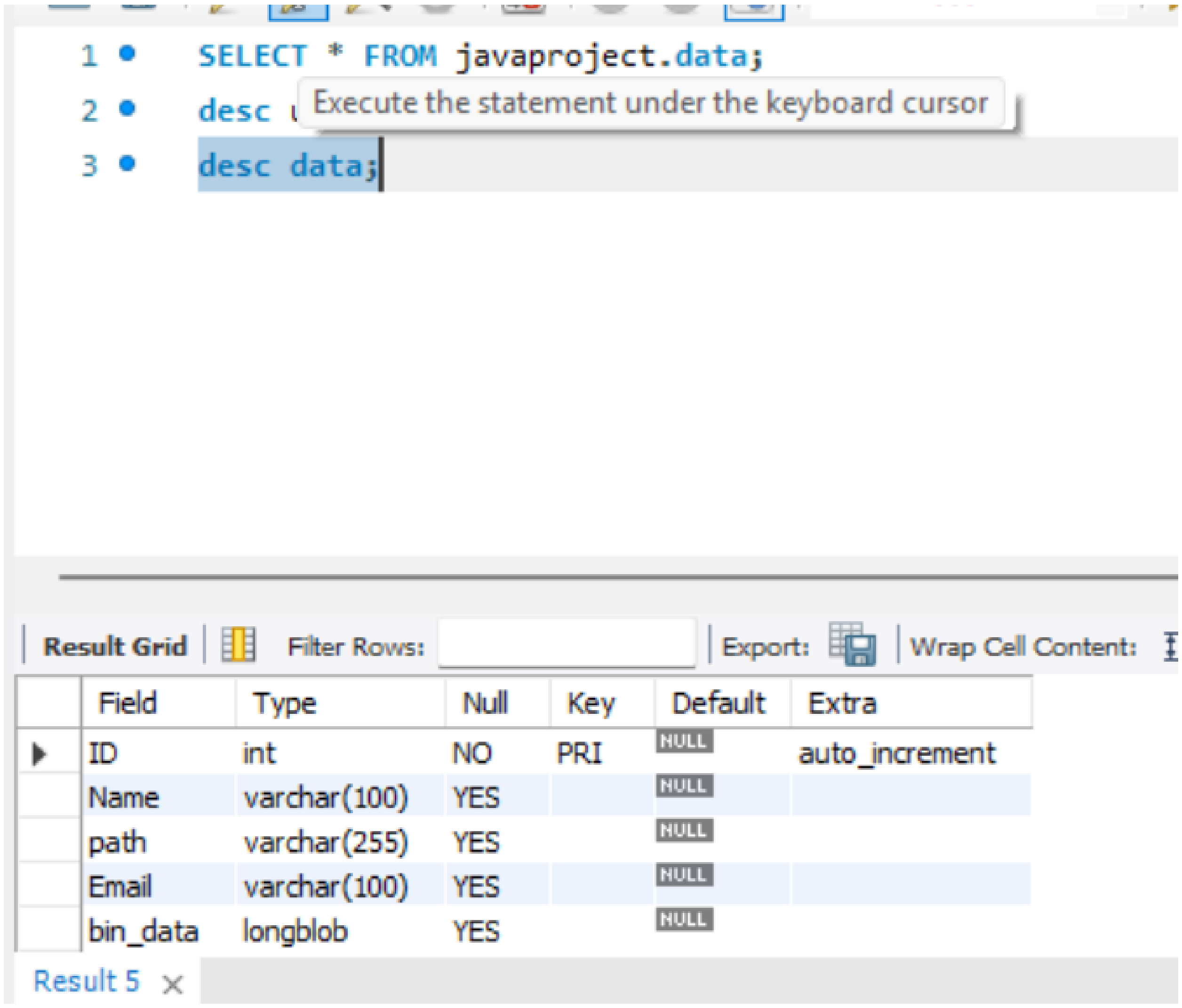
Make sql like this.



```
desc users;
```



Field	Type	Null	Key	Default	Extra
ID	int	NO	PRI	NULL	auto_increment
Name	varchar(100)	YES		NULL	
Email	varchar(100)	YES	UNI	NULL	



```
SELECT * FROM javaproject.data;
desc u
desc data;
```

Field	Type	Null	Key	Default	Extra
ID	int	NO	PRI	NULL	auto_increment
Name	varchar(100)	YES		NULL	
path	varchar(255)	YES		NULL	
Email	varchar(100)	YES		NULL	
bin_data	longblob	YES		NULL	

Mysql:-

```
use javaproject;

desc data;
CREATE TABLE data (
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NULL,
    path VARCHAR(255) NULL,
    Email VARCHAR(100) NULL,
    bin_data LONGBLOB NULL,
);

desc users;
```

```sql
CREATE TABLE users (
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NULL,
    Email VARCHAR(100) NULL UNIQUE,
);
```

DB Package

SqlConnection.java

```java
package DB;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class SqlConnection {
	public static Connection connection;
	public static Connection getConnection() {
		try {
			Class.forName("com.mysql.cj.jdbc.Driver");
			connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/JavaProject?useSSL=false",
"root", "your_pass"); // change your password. here
			System.out.println("Connected Succesfully.");
		} catch (ClassNotFoundException | SQLException e) {
			e.printStackTrace();
		}
		return connection;
	}

	public static void closeConnection() {
		if (connection != null) {
			try {
				connection.close();
			} catch (SQLException e) {
				e.printStackTrace();
			}
		}
	}
}
```

Purpose:

- **getConnection()**: Establishes a connection to the MySQL database using the provided JDBC URL, username, and password.
- **closeConnection()**: Closes the database connection once it's no longer needed.

DAO Package

UserDAO.java

```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import DB.SqlConnection;
import Model.User;

public class UserDAO { // DAO Full Form:- Data Access Object
/*	isExists(String email):

		Purpose: Checks if a user with the specified email exists in the
database.
		Parameters: String email - the email to be checked.
		Returns: boolean - true if the email exists, otherwise false.
 */

	public static boolean isExists(String email) throws SQLException {
		Connection connection = SqlConnection.getConnection();
		PreparedStatement ps = connection.prepareStatement("select email from
users");
		ResultSet rs = ps.executeQuery();

		while (rs.next()) {
			String e = rs.getString(1);
			if (e.equals(email)) {
				return true;
			}
		}
		return false;
	}
/*
	saveUser(User user):

		Purpose: Saves a new user to the database.
		Parameters: User user - the user object to be saved, containing the
user's name and email.
		Returns: int - the number of rows affected by the SQL INSERT operation.
*/
	public static int saveUser(User user) throws SQLException {
		Connection connection = SqlConnection.getConnection(); /
		PreparedStatement ps = connection.prepareStatement("Insert into users
values(default, ?, ?)"); /
		ps.setString(1, user.getName());
		ps.setString(2, user.getEmail());
		return ps.executeUpdate();
	}
}
```

DataDAO.java

```
package dao;

/* Work of this file
	getAllFiles(String email):
This method retrieves all the files from the database based on the provided email. It
executes a SQL query and stores the result in a list of Data objects.

hideFile(Data file):
This method hides a file by inserting it into the database. It stores the file’s
binary data (bin_data) in the database and then deletes the file from the file system
to "hide" it.

unhide(int id):
This method retrieves the file (based on its ID) from the database, writes its binary
data back to the file system (restores the file), and then deletes the record from
the database, effectively "unhiding" the file.

 */

import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Reader;
import java.sql.Clob;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import DB.SqlConnection;
import Model.Data;
```

```
       // getAllFiles function me sari files ko sql me se show kere ge. by email id.
public class DataDAO {
       public static List<Data> getAllFiles(String email) throws SQLException {
              Connection connection = SqlConnection.getConnection();
              PreparedStatement ps = connection.prepareStatement("Select * from data
where email = ?"); //insert the sql query
              ps.setString(1, email);
              ResultSet rs = ps.executeQuery();
              List<Data> files = new ArrayList<>();
              while (rs.next()) {
                     int id = rs.getInt(1);
                     String name = rs.getString(2);
                     String path = rs.getString(3);
                     files.add(new Data(id, name, path));
              }
              return files;
       }

       // we are using this function for hide the files.
       //hum pahele file ko bin_data formate me sql me save kere ge and after vo file
system me se delet kere ge.
       public static int hideFile(Data file)throws SQLException, IOException {
        Connection connection = SqlConnection.getConnection();
        PreparedStatement ps = connection.prepareStatement("insert into data(name,
path, email, bin_data) values(?,?,?,?)");
        ps.setString(1, file.getFileName());
        ps.setString(2, file.getPath());
        ps.setString(3, file.getEmail());
        File f = new File(file.getPath());
        FileReader fr = new FileReader(f);
        ps.setCharacterStream(4, fr, f.length());
        int ans = ps.executeUpdate();
        fr.close();
        if (f.exists()) {
            System.out.println("File exists. Attempting to delete...");
            if (f.delete()) {
                System.out.println("File deleted successfully.");
            } else {
                System.out.println("Failed to delete the file.");
            }
        } else {
            System.out.println("File not found.");
        }
              return ans;
    }


       // unhide function me hume jo file unhide kerni hee uska path SQL me se milega
and after we use FileWriter function and write that file in that location. after that
we delet that file data from sql.
       public static void unhide(int id) throws SQLException, IOException {
            Connection connection = SqlConnection.getConnection();
            PreparedStatement ps = connection.prepareStatement("select path, bin_data
from data where id = ?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            // Check if the result exists
            if (rs.next()) {
                String path = rs.getString("path");
                Clob c = rs.getClob("bin_data");

                Reader r = c.getCharacterStream();
                try (FileWriter fw = new FileWriter(path)) {
                    int i;
                    while ((i = r.read()) != -1) {
                        fw.write((char) i);
                    }
                }

                // Delete the record from the database after successfully un-hiding
the file
                ps = connection.prepareStatement("delete from data where id = ?");
                ps.setInt(1, id);
                ps.executeUpdate();

                System.out.println("Successfully Unhidden");
            } else {
                System.out.println("No file found with the provided ID.");
            }
        }
}
```

Model package

Data.java

```
package Model;

public class Data {
       private int id;
       private String fileName;
       private String path;
       private String email;

       public Data(int id, String fileName, String path, String email) {
              this.id = id;
              this.fileName = fileName;
              this.path = path;
              this.email = email;
       }

       public Data(int id, String fileName, String path) {
              this.id = id;
              this.fileName = fileName;
              this.path = path;
       }
```

```
	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getFileName() {
		return fileName;
	}

	public void setFileName(String fileName) {
		this.fileName = fileName;
	}

	public String getPath() {
		return path;
	}

	public void setPath(String path) {
		this.path = path;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

}
```

User.java

```
package Model;

public class User {
	private String name;
	private String email;
	public User(String name, String email) {
		this.name = name;
		this.email = email;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getEmail() {
```

```
                return email;
        }
        public void setEmail(String email) {
                this.email = email;
        }

}
```

Service Package

GenerateOTP.java

```
package service;

import java.util.Random;
// prepoce:- The GenerateOTP class is designed to generate a 4-digit OTP (One-Time
Password) using a random number generator. Here are some improvements and
suggestions:


public class GenerateOTP {
        public static String getOTP() {
                Random random = new Random();
                return String.format("%04d", random.nextInt(10000)) ;
        }
}
```

SendOTPService.java

```
package service;

import javax.mail.*;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import java.util.Properties;

public class SendOTPService {
    public static void sendOTP(String email, String genOTP) {
        // Recipient's email ID needs to be mentioned.
        String to = email;

        // Sender's email ID needs to be mentioned
        String from = "Your_mail@gmail.com"; // Enter your mail address.

        // Assuming you are sending email from through gmails smtp
        String host = "smtp.gmail.com";

        // Get system properties
        Properties properties = System.getProperties();
        // Setup mail server
        properties.put("mail.smtp.host", host);
        properties.put("mail.smtp.port", "465");
        properties.put("mail.smtp.ssl.enable", "true");
        properties.put("mail.smtp.auth", "true");

        // Get the Session object.// and pass username and password
        Session session = Session.getInstance(properties, new
javax.mail.Authenticator() {

            protected PasswordAuthentication getPasswordAuthentication() {


                return new PasswordAuthentication(from, "zuim katb omuf qasv");//
here Enter Your password or Use a security key

//
https://support.google.com/accounts/answer/6103523?hl=en&co=GENIE.Platform%3DAndroid




            }

        });

        // Used to debug SMTP issues
        session.setDebug(true);

        try {
            // Create a default MimeMessage object.
            MimeMessage message = new MimeMessage(session);

            // Set From: header field of the header.
            message.setFrom(new InternetAddress(from));

            // Set To: header field of the header.
            message.addRecipient(Message.RecipientType.TO, new InternetAddress(to));

            // Set Subject: header field
            message.setSubject("File Enc ka OTP");

            // Now set the actual message
            message.setText("Your One time Password for File Enc app is " + genOTP);

            System.out.println("sending...");
            // Send message
            Transport.send(message);
            System.out.println("Sent message successfully....");
        } catch (MessagingException mex) {
            mex.printStackTrace();
        }

    }
```

```
}
```

UserService.java

```
package service;

import Model.User;
import dao.UserDAO;


public class UserService {
	public static Integer saveUser(User user) {
		try {
			// If the user does not exist, it saves the user data using the 
UserDAO.saveUser() method.
			// If the user already exists, it returns 0 to indicate that 
saving was skipped.
			if (UserDAO.isExists(user.getEmail())) { //It checks if a user 
with the given email already exists in the database using the UserDAO.isExists() 
method.
				return 0;
			}else {
				return UserDAO.saveUser(user);
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
		return null;
	}
}
```

View Package

UserView.java

```
package views;

import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.List;
import java.util.Scanner;

import Model.Data;
import dao.DataDAO;

public class UserView {
	private String email;
	UserView(String email){
		this.email = email;
```

```
       }
       // The main method to display the user interface and handle user actions
       public void home() {
             do {
                    System.out.println("Wlcome "+ this.email);
                    System.out.println("Press 1 to show hidden files");
                    System.out.println("Press 2 to hide a new file");
                    System.out.println("Press 3 to unhide a file");
                    System.out.println("Press 0 to exit");
                    Scanner sc = new Scanner(System.in);
                    int ch = Integer.parseInt(sc.nextLine());
                    switch(ch) {
                    case 1 : {
                          try {
                                 List<Data> files =
DataDAO.getAllFiles(this.email);// Fetch all files for the current user
                                 System.out.println("ID - File Name");
                                 for (Data file : files) {  // Displaying each file
with its ID and name
                                       System.out.println(file.getId()+ " - "+
file.getFileName());
                                 }
                          } catch (SQLException e) {
                                 e.printStackTrace();
                          }
                    }
                    break;
                    case 2: { // Hide a file
                        System.out.println("Enter the file path:");
                        String path = sc.nextLine();
                        System.out.println("Path entered: " + path);
                        File f = new File(path);
                        if (!f.exists()) { // Check if file exists at the given path
                            System.out.println("File does not exist. Please check the
path.");

                            break;
                        }

                     // Create a Data object for the file to be hidden
                        Data file = new Data(0, f.getName(), path, this.email);
                        try {
                            DataDAO.hideFile(file); // Call the hideFile method to
hide the file
                            System.out.println("File hidden successfully.");
                        } catch (SQLException | IOException e) {
                            e.printStackTrace();
                        }
                    }
                    break;
                    case 3: { // Unhide a file
                          try {
                                 List<Data> files = DataDAO.getAllFiles(this.email);
                                 System.out.println("ID - File Name");
                                 for (Data file : files) {
                                        System.out.println(file.getId()+ " - "+
file.getFileName());
                                }

                        System.out.println("Enter the id of file to unhide");
                        int id = Integer.parseInt(sc.nextLine()); // Reading file
ID from user

                        // Check if the entered ID is valid
                        boolean isVaildID =false;
                        for (Data file : files) {
                                if (file.getId() == id) {
                                        isVaildID = true;
                                        break;
                                }
                        }
                                if (isVaildID) {
                                        DataDAO.unhide(id); //// Call the unhide
method to unhide the file
                                }else {
                                        System.out.println("Wrong ID.");
                                }

                        } catch (SQLException e) {
                                e.printStackTrace();
                        }catch (IOException e) {
                                // TODO: handle exception
                                e.printStackTrace();
                        }
                }
                break;
                case 0: {
                        System.exit(0);
                }
                }
        }while(true);
    }
}
```

Welcome.java

```
package views;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.SQLException;
import java.util.Scanner;

import Model.User;
import dao.UserDAO;
import service.GenerateOTP;
import service.SendOTPService;
```

```java
import service.UserService;

public class Welcome {
	private static final BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
	private static final Scanner sc = new Scanner(System.in);

	// This is my welcome screen.
	public void welcomeScreen() {
		System.out.println("Welcome to the app");
		System.out.println("Press 1 to login.");
		System.out.println("Press 2 to signup.");
		System.out.println("Press 0 to exit.");
		int choice = 0;
		try {
			choice = Integer.parseInt(br.readLine());
		switch (choice) {
		case 1:
			login();
			break;
		case 2:
			signUp();
			break;
		case 0:
			System.exit(0);
			break;
		default:
			System.out.println("Enter valid input.");
		}
		} catch (IOException | NumberFormatException e) {
			System.out.println("Invalid input. Please try again.");
			welcomeScreen();
		}
	}

	private void login() {
		System.out.println("Enter your email:");
		String email = sc.nextLine().trim();
		try {
			if (UserDAO.isExists(email)) {

				// Generate and send OTP
				String genOTP = GenerateOTP.getOTP();
				SendOTPService.sendOTP(email, genOTP);
				System.out.println("Enter the OTP:");
				String otp = sc.nextLine().trim();
				if (otp.equals(genOTP)) {
					new UserView(email).home();
				} else {
					System.out.println("Wrong OTP.");
				}
			} else {
				System.out.println("User not found.");
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	private void signUp() {
		System.out.println("Enter Name:");
		String name = sc.nextLine().trim();
		System.out.println("Enter email:");
		String email = sc.nextLine().trim();
		try {
			if (UserDAO.isExists(email)) {
				System.out.println("Email already exists.");
			} else {
				String genOTP = GenerateOTP.getOTP();
				SendOTPService.sendOTP(email, genOTP);
				System.out.println("Enter the OTP:");
				String otp = sc.nextLine().trim();
				if (otp.equals(genOTP)) {
					User user = new User(name, email);
					int response = UserService.saveUser(user);
					switch (response) {
					case 0:
						System.out.println("User already exists.");
						break;
					case 1:
						System.out.println("User registered 
successfully.");
						break;
					default:
						System.out.println("An unknown error 
occurred.");
					}
				} else {
					System.out.println("Wrong OTP. Registration 
failed.");
				}
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}
}
```

Main.java

```
	import views.Welcome;

	public class Main {
		public static void main(String[] args) {
			Welcome w = new Welcome();

			do {
				w.welcomeScreen();
```

```
                    } while (true);
            }
    }
```

Pom.xml

Dependencies which I am added.

MySQL Connector/J (JDBC Driver)

- **Purpose:**
    - **Provides connectivity between the application and a MySQL database.**
    - **Allows the application to execute SQL queries and interact with the database.**

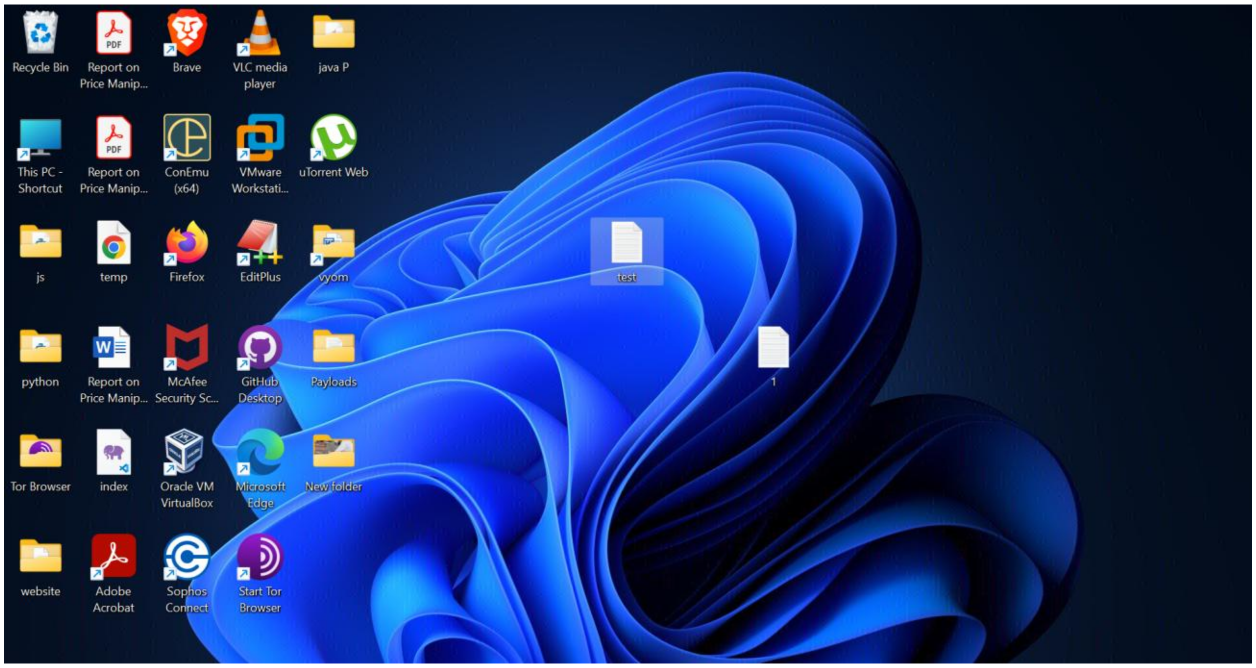
JavaMail API (Email Services)

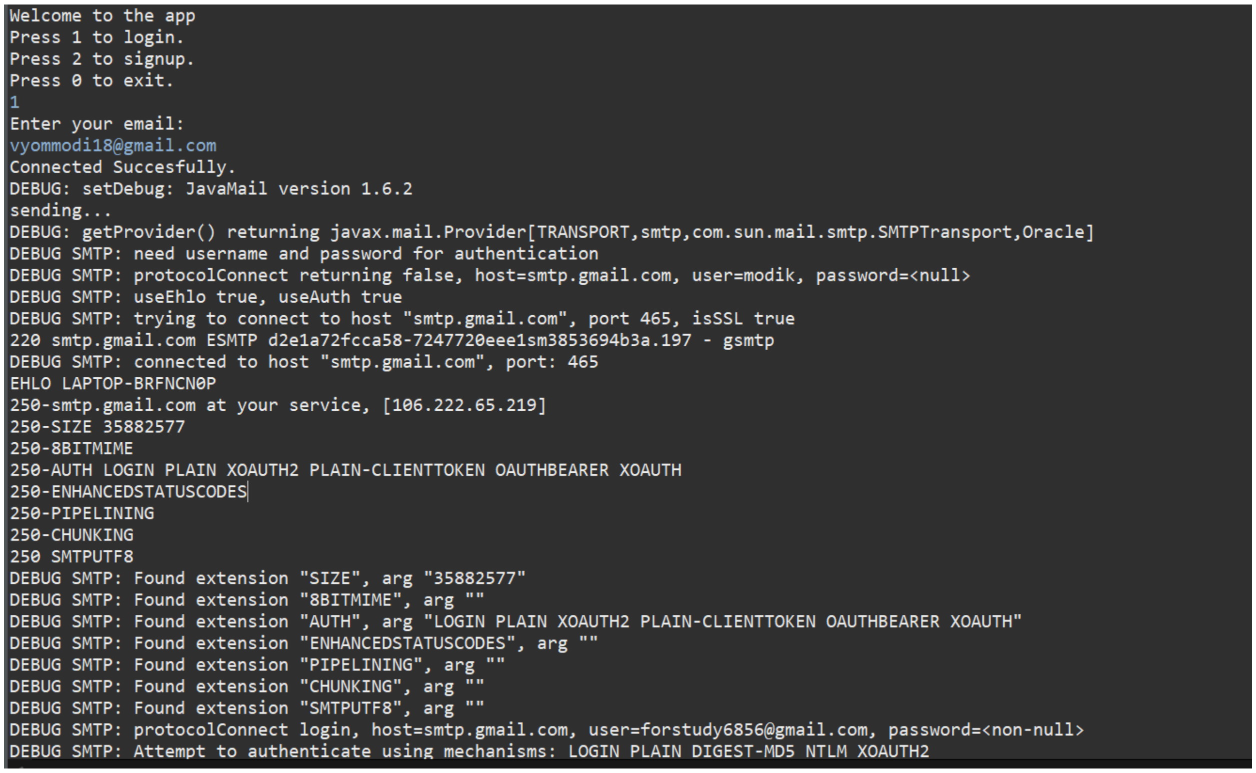
- **Purpose:**
    - **Enables the application to send emails using the JavaMail API.**
    - **Useful for implementing email-based features such as sending OTPs or notifications.**

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
      <modelVersion>4.0.0</modelVersion>

      <groupId>Email_Authentication</groupId>
      <artifactId>EmailAuthentication</artifactId>
      <version>0.0.1-SNAPSHOT</version>
      <packaging>jar</packaging>

      <name>EmailAuthentication</name>
      <url>http://maven.apache.org</url>

      <properties>
            <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
      </properties>

      <dependencies>
            <dependency>
                  <groupId>junit</groupId>
```

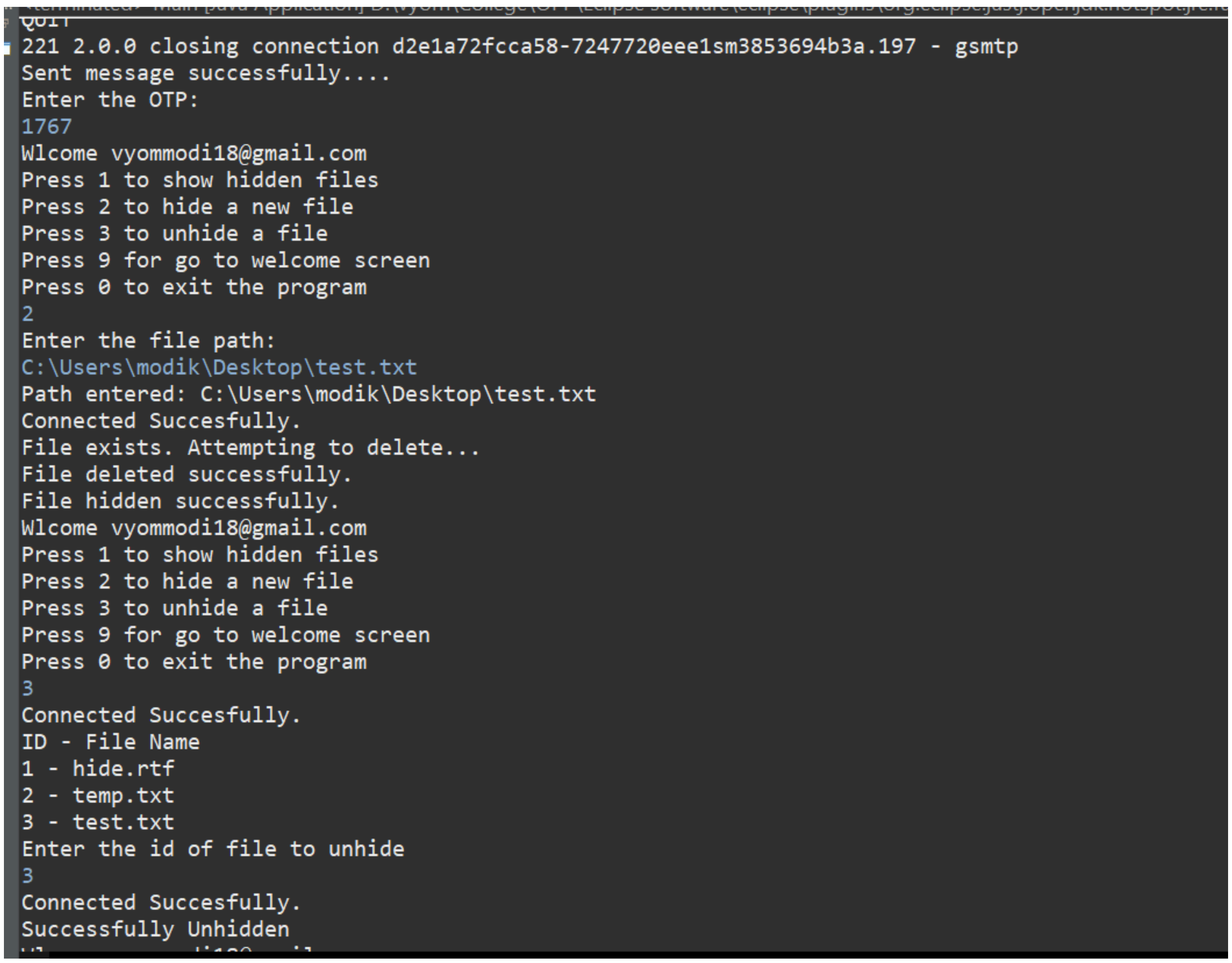
```
                <artifactId>junit</artifactId>
                <version>3.8.1</version>
                <scope>test</scope>
            </dependency>
            <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
            <dependency>
                <groupId>mysql</groupId>
                <artifactId>mysql-connector-java</artifactId>
                <version>8.0.33</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/javax.mail/javax.mail-api -->
            <dependency>
                <groupId>com.sun.mail</groupId>
                <artifactId>javax.mail</artifactId>
                <version>1.6.2</version>
            </dependency>

        </dependencies>
</project>
```

Outputs:-

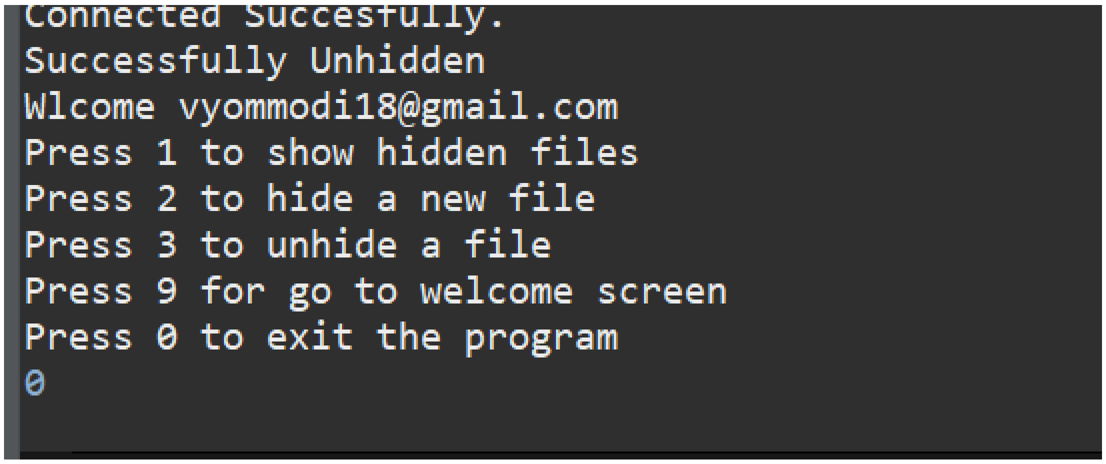




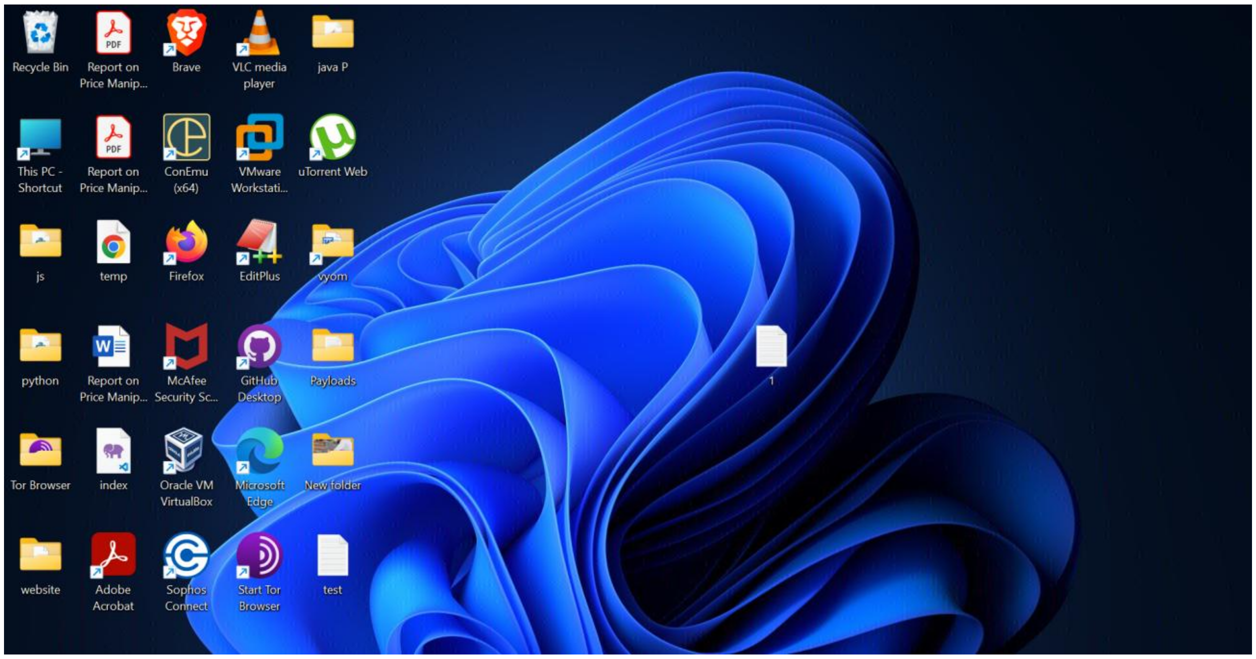
```
Welcome to the app
Press 1 to login.
Press 2 to signup.
Press 0 to exit.
1
Enter your email:
vyommodi18@gmail.com
Connected Succesfully.
DEBUG: setDebug: JavaMail version 1.6.2
sending...
DEBUG: getProvider() returning javax.mail.Provider[TRANSPORT,smtp,com.sun.mail.smtp.SMTPTransport,Oracle]
DEBUG SMTP: need username and password for authentication
DEBUG SMTP: protocolConnect returning false, host=smtp.gmail.com, user=modik, password=<null>
DEBUG SMTP: useEhlo true, useAuth true
DEBUG SMTP: trying to connect to host "smtp.gmail.com", port 465, isSSL true
220 smtp.gmail.com ESMTP d2e1a72fcca58-7247720eee1sm3853694b3a.197 - gsmtp
DEBUG SMTP: connected to host "smtp.gmail.com", port: 465
EHLO LAPTOP-BRFNCN0P
250-smtp.gmail.com at your service, [106.222.65.219]
250-SIZE 35882577
250-8BITMIME
250-AUTH LOGIN PLAIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUTH
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
DEBUG SMTP: Found extension "SIZE", arg "35882577"
DEBUG SMTP: Found extension "8BITMIME", arg ""
DEBUG SMTP: Found extension "AUTH", arg "LOGIN PLAIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUTH"
DEBUG SMTP: Found extension "ENHANCEDSTATUSCODES", arg ""
DEBUG SMTP: Found extension "PIPELINING", arg ""
DEBUG SMTP: Found extension "CHUNKING", arg ""
DEBUG SMTP: Found extension "SMTPUTF8", arg ""
DEBUG SMTP: protocolConnect login, host=smtp.gmail.com, user=forstudy6856@gmail.com, password=<non-null>
DEBUG SMTP: Attempt to authenticate using mechanisms: LOGIN PLAIN DIGEST-MD5 NTLM XOAUTH2
```



```
QUIT
221 2.0.0 closing connection d2e1a72fcca58-7247720eee1sm3853694b3a.197 - gsmtp
Sent message successfully....
Enter the OTP:
1767
Wlcome vyommodi18@gmail.com
Press 1 to show hidden files
Press 2 to hide a new file
Press 3 to unhide a file
Press 9 for go to welcome screen
Press 0 to exit the program
2
Enter the file path:
C:\Users\modik\Desktop\test.txt
Path entered: C:\Users\modik\Desktop\test.txt
Connected Succesfully.
File exists. Attempting to delete...
File deleted successfully.
File hidden successfully.
Wlcome vyommodi18@gmail.com
Press 1 to show hidden files
Press 2 to hide a new file
Press 3 to unhide a file
Press 9 for go to welcome screen
Press 0 to exit the program
3
Connected Succesfully.
ID - File Name
1 - hide.rtf
2 - temp.txt
3 - test.txt
Enter the id of file to unhide
3
Connected Succesfully.
Successfully Unhidden
```



```
Connected Succesfully.
Successfully Unhidden
Wlcome vyommodi18@gmail.com
Press 1 to show hidden files
Press 2 to hide a new file
Press 3 to unhide a file
Press 9 for go to welcome screen
Press 0 to exit the program
0
```


Recycle Bin
Report on Price Manip...
Brave
VLC media player
java P
This PC - Shortcut
Report on Price Manip...
ConEmu (x64)
VMware Workstati...
uTorrent Web
js
temp
Firefox
EditPlus
vyom
python
Report on Price Manip...
McAfee Security Sc...
GitHub Desktop
Payloads
1
Tor Browser
index
Oracle VM VirtualBox
Microsoft Edge
New folder
website
Adobe Acrobat
Sophos Connect
Start Tor Browser
test