A Course Based Project Report on

Email evaluator

Submitted to the

Department of Information Technology

in partial fulfillment of the requirements for the completion of course

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LABORATORY

(22PC2IT201)

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

I. ABHILESH	**22071A1225**
J. RAKESH	**22071A1226**
J. ABHINAV SAI	**22071A1227**
J. SHIVARAM	**22071A1228**

Under the guidance of

Dr. Gunupudi Rajesh Kumar

(Course Instructor)

Associate Professor, Department of IT, VNRVJIET

DEPARTMENT OF INFORMATION TECHNOLOGY

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DECEMBER 2023

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report entitled **"EMAIL EVALUATOR"** is a bonafide work done under our supervision and is being submitted by **Mr. I. Abhilesh (22071A1225), Mr. J. Rakesh (22071A1226), Mr. J. Abhinav Sai (22071A1227), Mr. J. Shivaram (22071A1228)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Information Technology, of the VNRVJIET, Hyderabad during the academic year 2023-2024.

Dr. Gunupudi Rajesh Kumar
Associate Professor
Department of IT

Dr D Srinvasa Rao
Associate Professor & HOD
Department of IT

Course based Projects Reviewer

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade,
Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We declare that the course based project work entitled "**Email evaluator**" submitted in the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Information Technology** is a bonafide record of our own work carried out under the supervision of **Dr. Gunupudi Rajesh Kumar, Associate Professor, Department of IT, VNRVJIET.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

I. Abhilesh	**J. Rakesh**	**J. Abhinav Sai**	**J. Shiva ram**
(22071A1225)	**(**22071A1226)	(22071A1227)	(22071A1228)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor Dr. SRINIVASA RAO DAMMAVALAM, Associate Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, **Dr. Gunupudi Rajesh Kumar**, Associate Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

Mr. I. Abhilesh (22071A1225)

Mr. J. Rakesh (22071A1226)

Mr. J. Abhinav sai (22071A1227)

Mr. J. Shiva ram (22071A1228)

ABSTRACT

The Email Validator Application is a Java-based software tool designed to offer a user-friendly environment for email address validation. Leveraging Java's Swing library, the application presents a graphical user interface (GUI) that simplifies the process of validating email addresses. Users can input an email address into the provided text field and trigger the validation process by clicking a dedicated button. The application employs a robust email validation algorithm based on a regular expression pattern, ensuring a reliable assessment of whether the entered email adheres to common format standards. Immediate feedback is provided through a dynamic result label on the GUI, indicating whether the email address is valid or fails to meet the expected format. With an emphasis on ease of use, the intuitive design makes the Email Validator accessible to users with varying technical expertise. The application also incorporates error handling mechanisms to gracefully manage invalid inputs or technical issues, providing clear error messages for improved user understanding. Overall, the Email Validator Application is a valuable utility for developers, quality assurance professionals, and end-users seeking a convenient tool to enhance data quality and user experience in email-related processes.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

1.1 PROBLEM DEFINITION

The program aims to create a simple Email Validator application using Java Swing. It provides a graphical user interface (GUI) where users can input their details like email to verify it..

1.2 OBJECTIVE

- The primary objective of the Email Validator application is to provide a user-friendly and efficient solution for validating email addresses.

1.3 OVERVIEW

1. Components:

- Text fields for user input (email).
- "Verify Email" button to execute the program.
- GUI layout is structured using JPanel, JTable, JScrollPane, and JFrame.

2. Functionality:

- Users input their details into designated text field.
- Upon clicking the "Verify Email" button, the email is verified.

3. Main Method:

- The main method initializes the GUI by creating an instance of the EmailVerifierApp class.

CHAPTER-2

SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class EmailVerifierApp extends JFrame {
    private JTextField emailTextField;
    private JLabel resultLabel;

    public EmailVerifierApp() {
        super("Email Verifier");

        // Create components
        emailTextField = new JTextField(20);
        JButton verifyButton = new JButton("Verify Email");
        resultLabel = new JLabel("");

        // Set layout
        setLayout(new FlowLayout());

        // Add components to the frame
```

```
        add(new JLabel("Enter Email: "));
        add(emailTextField);
        add(verifyButton);
        add(resultLabel);

        // Add action listener to the verify button
        verifyButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                verifyEmail();
            }
        });

        // Set frame properties
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 150);
        setLocationRelativeTo(null); // Center the frame
        setVisible(true);
    }

    private void verifyEmail() {
        String email = emailTextField.getText();
        if (isValidEmail(email)) {
            resultLabel.setForeground(Color.GREEN);
            resultLabel.setText("Email is valid!");
        } else {
            resultLabel.setForeground(Color.RED);
            resultLabel.setText("Invalid email format!");
        }
```

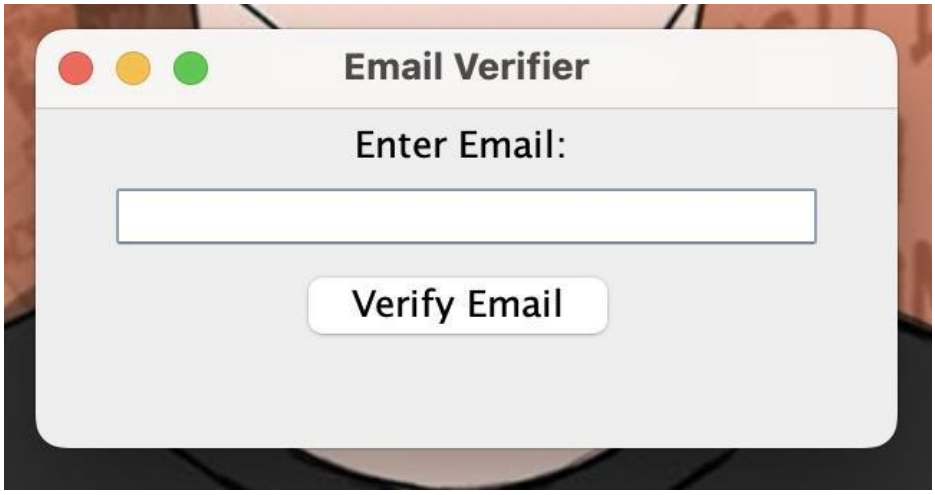
```
    }

    private boolean isValidEmail(String email) {
        String emailRegex = "^[a-zA-Z0-9_+&*-]+(?:\\.[a-zA-Z0-9_+&*-
]+)*@(?:[a-zA-Z0-9-]+\\.)+[a-zA-Z]{2,7}$";
        Pattern pattern = Pattern.compile(emailRegex);
        Matcher matcher = pattern.matcher(email);
        return matcher.matches();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new EmailVerifierApp();
            }
        });
    }
}
```

CHAPTER-3

TEST CASES/ OUTPUT

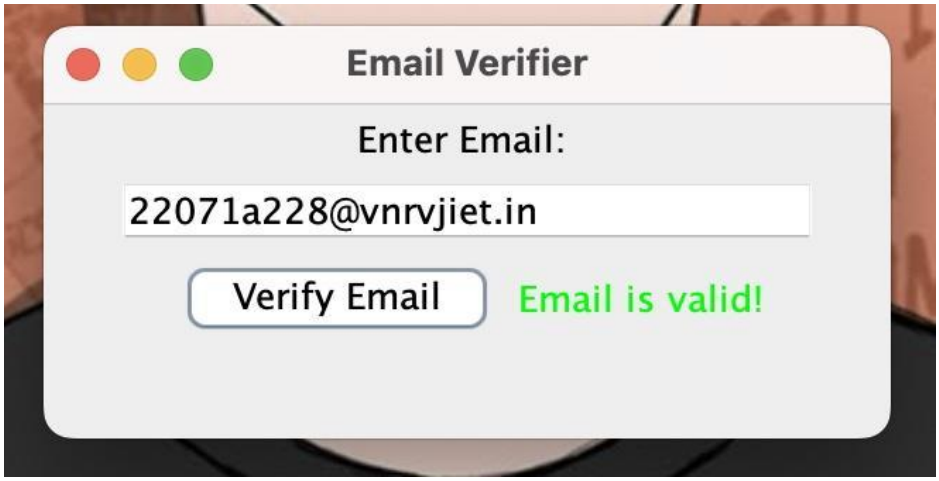


GUI of Email Verification Application

The application consists of a Jlabel, JtextFeild and Jbutton.

Entering a email for test case

An email has been entered into the JtextField to verify whether it is a valid email or not. This is done by clicking the verify email button.

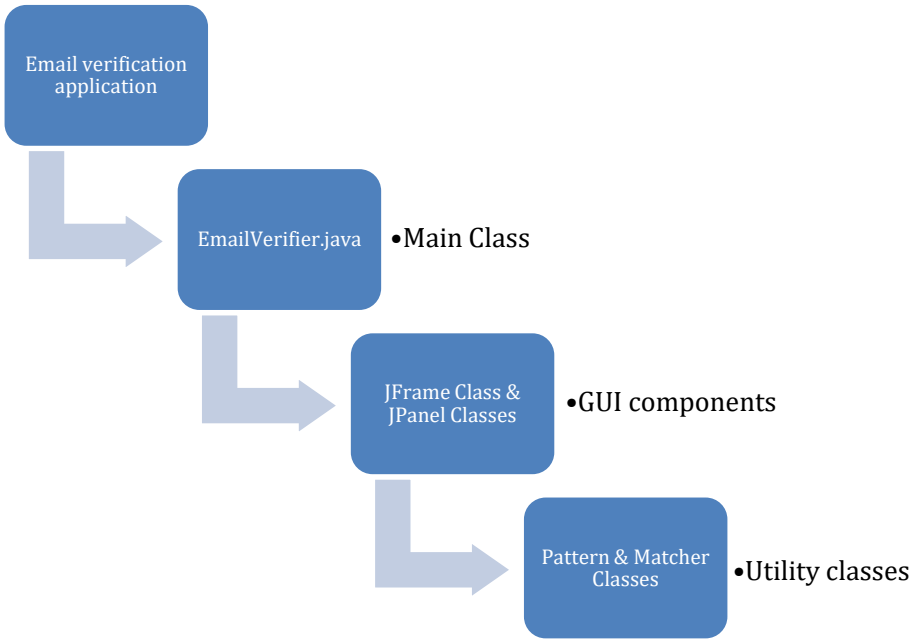


Output of the Application

Upon entering the email and clicking the button, we get the output of the application displayed in the frame.

CHAPTER-4

PROJECT ARCHITECTURE



Flow chart explaining the interaction of classes with each other in the application.

The application consists of the main class named "EmailVerifier.java" which consists of the main method. It is the entry point into out application. Then we have used the JFrame and JPanel classes with the help of Swing to create a graphical user interface. Then we have used the Pattern and Matcher classes from the utilities package to evaluate the email input by the user.

CHAPTER-5

CONCLUSION

In conclusion, the Email Validator Application serves as a valuable tool to streamline and enhance the process of email address validation. With a user-friendly graphical user interface, the application enables users to effortlessly input email addresses and promptly receive validation feedback. The incorporation of a robust email validation algorithm, based on a well-defined regular expression pattern, ensures the accurate assessment of email addresses against common format standards.

The application's emphasis on instantaneous feedback through a dynamic result label contributes to a seamless user experience. Clear error-handling mechanisms further enhance the application's reliability, providing users with informative messages in the event of invalid inputs or technical issues.

Versatility lies at the core of the Email Validator, accommodating a diverse user base that includes developers, quality assurance professionals, and end-users. Its role in contributing to improved data quality, particularly in scenarios where valid and properly formatted email addresses are essential, underscores its significance in various applications, such as web forms, registration processes, and communication systems.

By focusing on user-friendliness, reliability, and versatility, the Email Validator Application addresses the need for an accessible and efficient tool for email address validation. It stands as a testament to the importance of ensuring data accuracy and user satisfaction in email-related processes, ultimately contributing to a smoother and more efficient workflow for users across different domains.

REFERENCES

[1]. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition, Pearson Education

[2]. Thinking in Java, Bruce Eckel, Pearson Education

[3]. Understanding Object-Oriented Programming with Java, T. Budd, Pearson Education

[4]. https://www.w3schools.com/java/java_oop.asp

[5]. https://www.tutorialspoint.com/java/index.htm