



INNOVATION. AUTOMATION. ANALYTICS

**PROJECT ON**

**REGEX Matcher App Deployment**

# About me

## **Background:**

I am Surya Atrish pursuing B.Tech in Computer Science and Engineering with a strong interest in Python and Data Analytics.

## **Motivation for Data Science:**

Since the start of my undergraduate life, I was deeply interested in Python, its various libraries and frameworks and all the various kinds of problems it can solve. So, I started learning about it, started doing projects. At the same time, I also completed an internship on Data Analytics in Python, SQL Tableau and Power BI. Through research and introspection, I discovered that Data Science resonates deeply with me, offering a perfect fit for my aspirations and skills.

## **Work Experience:**

I am Currently interning at Innomatics Research Labs, transitioning from a computer science and engineering background to data science.

**LinkedIn:** <https://www.linkedin.com/in/surya-atrish/>

**GitHub:** <https://github.com/suryaatrish>

# Objective:

The aim of this project is to develop a Regex Matcher web application that enables users to input a test string and a regular expression (regex) pattern. Upon submission, the application will identify and display all matches found in the test string based on the provided regex pattern. Additionally, the application will offer functionality for validating email addresses entered by users. By providing a user-friendly interface for regex matching and email validation, the web application seeks to streamline the process of pattern matching and enhance user productivity.

# Key Features:

The Regex Matcher app will empower users to effortlessly test regex patterns against sample text strings, facilitating efficient data extraction and manipulation tasks. Furthermore, the integrated email validation feature will offer users a convenient method to verify the correctness of email addresses, ensuring data integrity and accuracy in various applications. With a focus on simplicity and functionality, the Regex Matcher app aims to serve as a valuable tool for developers, data analysts, and individuals requiring precise pattern matching capabilities.

# Codebase:

## Python:

```
from flask import Flask, render_template, request, jsonify
import re

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/results', methods=['POST'])
def results():
    test_string = request.json['test_string']
    regex = request.json['regex']
    matches = re.findall(regex, test_string)
    return jsonify({'matches': matches})

@app.route('/validate_email', methods=['POST'])
def validate_email():
    email = request.json['email']
    if re.match(r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$', email):
        return 'Valid Email'
    else:
        return 'Invalid Email'

if __name__ == '__main__':
    app.run(debug=True, host="0.0.0.0")
```

## HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Regex Matcher</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
  <h1>Regex Matcher</h1>

  <!-- Regex Input Form -->
  <section id="regexSection" class="...">
    <h2>Regular Expression Matcher</h2>
    <form id="regexForm" class="...">
      <label for="test_string">Test String:</label>
      <input type="text" id="test_string" name="test_string" required><br><br>
      <label for="regex">Regex:</label>
      <input type="text" id="regex" name="regex" required><br>
      <small id="regexResult" style="color: red;"></small><br><br>
      <input type="submit" value="Submit">
    </form>
    <small id="matchesResult"></small>
  </section>

  <!-- Email Validation -->
  <section id="emailValidationSection" class="...">
    <h2>Email Validation</h2>
    <form id="emailValidationForm" class="...">
      <label for="email">Enter Email:</label>
      <input type="email" id="email" name="email" required>
      <input type="submit" value="Validate">
      <div id="emailValidationResult"></div>
    </form>
  </section>
```

# Codebase:

```
<!-- Text Sections -->
<section class="ooo">
  <h2>Text Strings</h2>
  <ul id="textStringsList">
    <li>Text strings are sequences of characters, such as words or phrases, that are used as input data for various operations or analyses.</li>
    <li>For example, "apple, banana, orange" and "watermelon, kiwi, melon" are text strings containing names of fruits separated by commas.</li>
    <li>Text strings can contain alphanumeric characters, punctuation, whitespace, and other symbols.</li>
  </ul>
</section>

<section class="ooo">
  <h2>Regular Expressions</h2>
  <ul id="regexList">
    <li>Regular expressions (regex) are patterns used to match character combinations in strings.</li>
    <li>They are commonly used for searching, extracting, and validating specific patterns of text.</li>
    <li>For example, "\b[a-zA-Z]+\b" matches words containing only alphabetic characters.</li>
  </ul>
</section>

<section class="ooo">
  <h2>Emails</h2>
  <p>Emails are electronic messages sent over a computer network, typically the internet, from one person to another.</p>
  <h3>Valid</h3>
  <ul id="validEmailsList">
    <li>Valid email addresses adhere to specific syntax rules and can be successfully delivered to recipients.</li>
    <li>For example, "user@example.com" and "test@example.com" are valid email addresses.</li>
  </ul>
  <h3>Invalid</h3>
  <ul id="invalidEmailsList">
    <li>Invalid email addresses do not conform to the syntax rules and may fail to be delivered or rejected by email servers.</li>
    <li>For example, "user@example" and "test@example." are invalid email addresses.</li>
  </ul>
</section>
```

# Codebase:

```
<!-- JavaScript -->
<script>
  document.getElementById("regexForm").addEventListener("submit", function(event) {
    event.preventDefault();
    var testString = document.getElementById("test_string").value;
    var regexPattern = document.getElementById("regex").value;
    var regexResult = document.getElementById("regexResult");
    if (testString.match(new RegExp(regexPattern))) {
      regexResult.innerText = "Valid Regex";
      regexResult.style.color = "green";
      // Fetch and display the matches
      fetch('/results', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json'
        },
        body: JSON.stringify({ test_string: testString, regex: regexPattern })
      })
      .then(response => {
        if (!response.ok) {
          throw new Error('Failed to fetch matches');
        }
        return response.json();
      })
      .then(data => {
        var matchesResult = document.getElementById("matchesResult");
        matchesResult.innerText = "Matches Found: " + data.matches.join(", ");
      })
      .catch(error => {
        console.error('Error:', error);
        alert('An error occurred while fetching matches: ' + error.message);
      });
    } else {
      regexResult.innerText = "Invalid Regex";
      regexResult.style.color = "red";
    }
  });
};
```

```
document.getElementById("emailValidationForm").addEventListener("submit", function(event) {
  event.preventDefault();
  var email = document.getElementById("email").value;
  // Send the email to the server for validation
  fetch('/validate_email', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({email: email})
  })
  .then(response => {
    if (!response.ok) {
      throw new Error('Failed to validate email');
    }
    return response.text();
  })
  .then(result => {
    if (result === 'Valid Email') {
      document.getElementById("emailValidationResult").innerHTML = "<span style='color:green;'>Valid Email</span>";
    } else {
      document.getElementById("emailValidationResult").innerHTML = "<span style='color:red;'>Invalid Email</span>";
    }
  })
  .catch(error => {
    console.error('Error:', error);
    alert('An error occurred while validating email: ' + error.message);
  });
});
</script>
</body>
</html>
```

# Codebase:

CSS:

```
/* General Styles */
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  margin: 0;
  padding: 0;
  color: #333;
}

.container {
  width: 80%;
  margin: 20px auto;
}

h1, h2 {
  text-align: center;
}

/* Form Styles */
.form-container {
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 20px;
}

.form-container label {
  display: block;
  margin-bottom: 10px;
}

.form-container input[type="text"],
.form-container input[type="email"],
.form-container input[type="submit"],
.form-container button {
  width: 100%;
  padding: 10px;
  margin: 5px 0 10px 0;
  border: 1px solid #ccc;

```

```
  border-radius: 5px;
  box-sizing: border-box;
}

.form-container input[type="submit"],
.form-container button {
  cursor: pointer;
  background-color: #007bff;
  color: #fff;
  border: none;
}

.form-container input[type="submit"]:hover,
.form-container button:hover {
  background-color: #0056b3;
}

.validate-button {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  border-radius: 8px;
}

/* Results Styles */
.results-container {
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  padding: 20px;
  margin-top: 20px;

```

```

}

.results-container h2 {
  margin-bottom: 10px;
}

.results-container ul {
  list-style: none;
  padding: 0;
}

.results-container ul li {
  margin-bottom: 5px;
}

/* Text Section Styles */
.text-section {
  background-color: #f9f9f9;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  margin-top: 20px;
}

.text-section h2 {
  margin-bottom: 10px;
}

.text-section ul {
  list-style: none;
  padding: 0;
}

.text-section ul li {
  margin-bottom: 5px;
}

```

# Final Output and Directory Structure:

←

→

↻

🏠

⚠ Not secure 43.204.221.56:5000

☆

🔊

🔒

📁

🖨

👤

⋮

Regex Matcher

Regular Expression Matcher

Test String:

apple, banana, orange, 2023-01-01, user@example.com, 123-456-7890 watermelon, kiwi, melon, 2023-02-15, test@example.com, 987-654-3210 strawberry, blueberry, grape, 2023-03-30, info@e:

Regex:

\b\w+@\w+\.\w+\b

Valid Regex

Submit

Matches Found: user@example.com, test@example.com, info@example.com, support@example.com, sales@example.com

Email Validation

Enter Email:

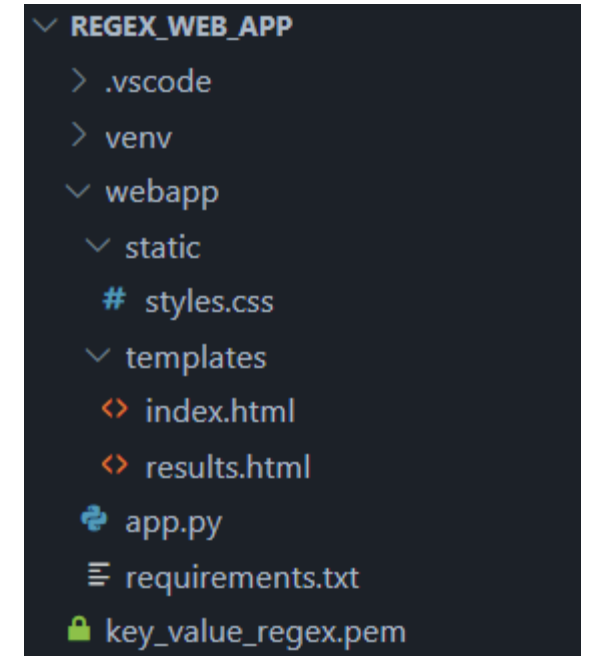
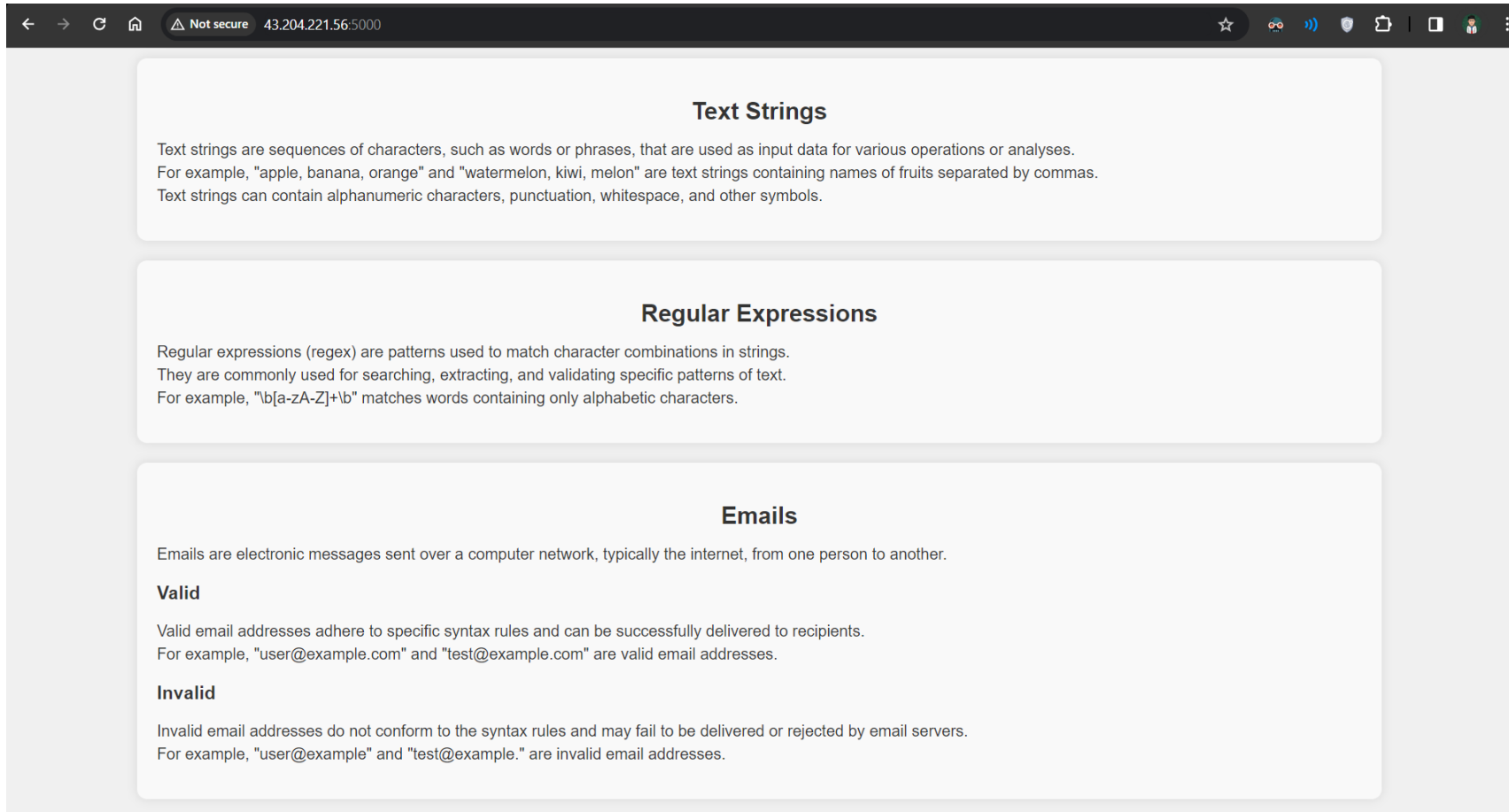
atrishsurya460@gmail.com

Validate

Valid Email



# Final Output and Directory Structure:



# Conclusion:

In conclusion, the Regex Matcher web application provides users with a convenient platform for pattern matching and email validation tasks. By allowing users to input a test string and a regular expression (regex) pattern, the application efficiently identifies and displays all matches found in the test string based on the provided regex pattern. Additionally, the integrated email validation feature offers users the capability to verify the correctness of email addresses, ensuring data integrity and reliability.

Through its user-friendly interface and robust functionality, the Regex Matcher app serves as a valuable tool for developers, data analysts, and individuals requiring precise pattern matching capabilities. By streamlining the process of regex matching and email validation, the application enhances user productivity and facilitates efficient data extraction and manipulation tasks. Overall, the Regex Matcher web application empowers users to effectively tackle pattern matching challenges and achieve accurate results in various applications and scenarios.

THANK  
YOU

