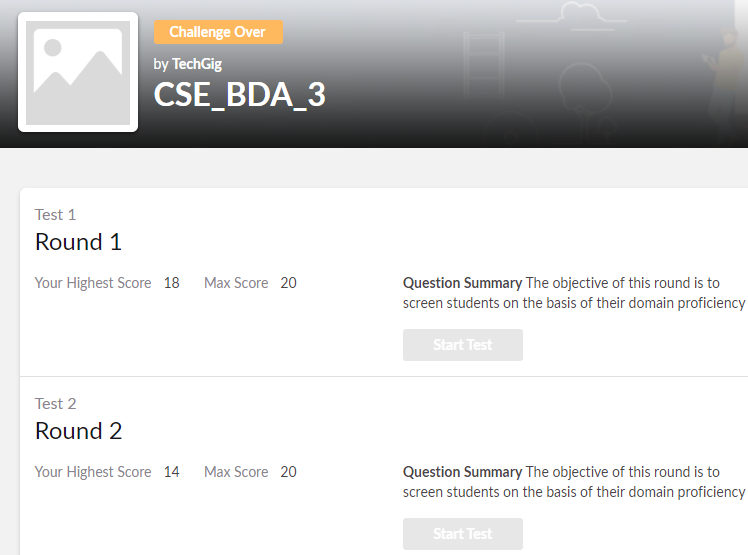
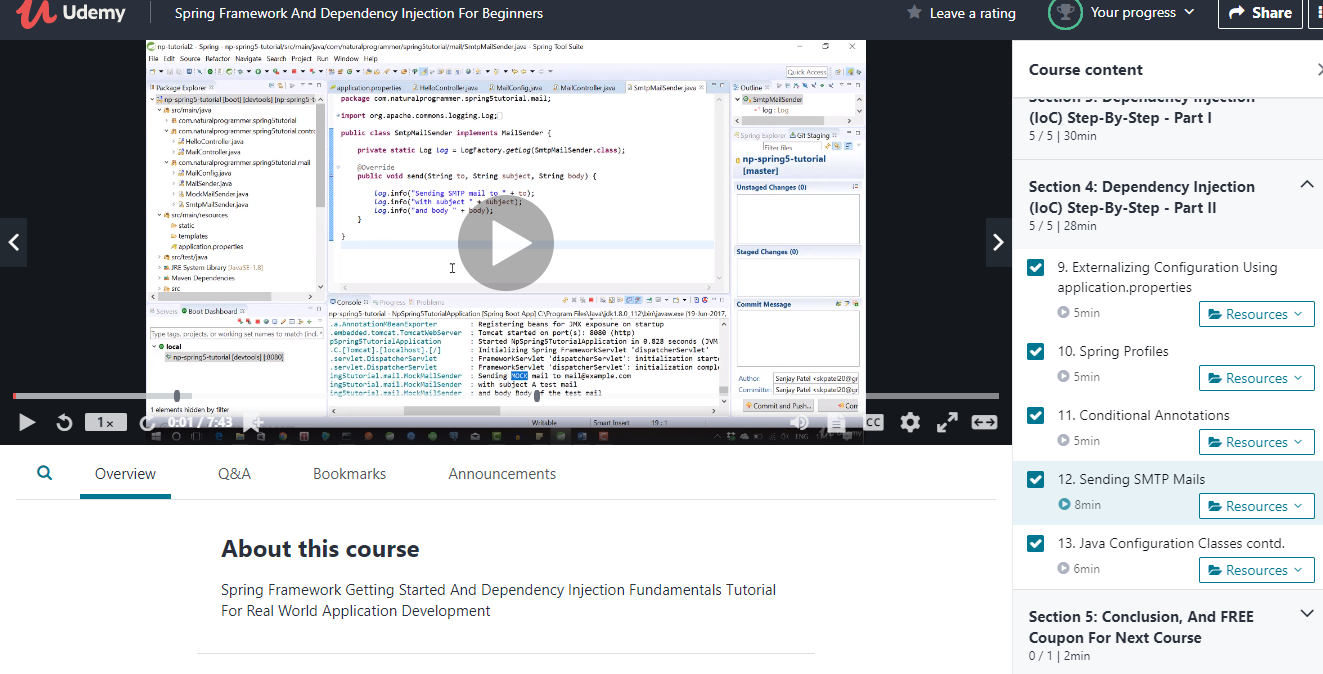
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **26/05/2020** | | | | **Name:** | **Ravi K R** | |
| **Sem & Sec** | **8th- B** | | | | **USN:** | **4AL16CS076** | |
| **Online Test Summary** | | | | | | | |
| **Subject** | | **BDA** | | | | | |
| **Max. Marks** | | **40** | | **Score** | | **32** | |
| **Certification Course Summary** | | | | | | | |
| **Course** | **Spring Framework Getting Started And Dependency Injection Fundamentals Tutorial For Real World Application Development** | | | | | | |
| **platform** | | | **Udemy** | **Duration** | | | **2 hours** |
| **Coding Challenges** | | | | | | | |
| **Problem Statement:**  **1) Write** the java code to validate IP adress with the help of Regular Expressions or Regex. | | | | | | | |
| **Status: Executed** | | | | | | | |
| **Uploaded the report in Github** | | | | **Yes** | | | |
| **If yes Repository name** | | | | <https://github.com/alvas-education-foundation/Ravi_kr> | | | |
| **Uploaded the report in slack** | | | | **Yes** | | | |

Online Test Details:



Certification:



Coding Challenges Details:

import java.util.regex.\*;

class IPAddressValidation {

    // Function to validate the IPs address.

    public static boolean isValidIPAddress(String ip)

    {

        // Regex for digit from 0 to 255.

        String zeroTo255

            = "(\\d{1,2}|(0|1)\\"

              + "d{2}|2[0-4]\\d|25[0-5])";

        // Regex for a digit from 0 to 255 and

        // followed by a dot, repeat 4 times.

        // this is the regex to validate an IP address.

        String regex

            = zeroTo255 + "\\."

              + zeroTo255 + "\\."

              + zeroTo255 + "\\."

              + zeroTo255;

        // Compile the ReGex

        Pattern p = Pattern.compile(regex);

        // If the IP address is empty

        // return false

        if (ip == null) {

            return false;

        }

        // Pattern class contains matcher() method

        // to find matching between given IP address

        // and regular expression.

        Matcher m = p.matcher(ip);

        // Return if the IP address

        // matched the ReGex

        return m.matches();

    }

    // Driver code

    public static void main(String args[])

    {

        // Checking for True case.

        // Test Case: 1

        System.out.println("Test Case 1:");

        String str1 = "000.12.12.034";

        System.out.println("Input: " + str1);

        System.out.println(

            "Output: "

            + isValidIPAddress(str1));

        // Test Case: 2

        System.out.println("\nTest Case 2:");

        String str2 = "121.234.12.12";

        System.out.println("Input: " + str2);

        System.out.println(

            "Output: "

            + isValidIPAddress(str2));

        // Checking for False case.

        // Test Case: 3

        System.out.println("\nTest Case 3:");

        String str3 = "000.12.234.23.23";

        System.out.println("Input: " + str3);

        System.out.println(

            "Output: "

            + isValidIPAddress(str3));

        // Test Case: 4

        System.out.println("\nTest Case 4:");

        String str4 = "I.Am.not.an.ip";

        System.out.println("Input: " + str4);

        System.out.println(

            "Output: "

            + isValidIPAddress(str4));

    }

}