**Project-Report**

**COVID TRACKER APP USING JAVA**



**Submitted to: Sir Shahroz Bakht**

**By:**

**Muhammad Tabish Rashid (SP20-BSCS-0031)**

**Basim Ahmad Siddiqui (SP20-BSCS-0026)**

**Zaeem Ahmed (SP20-BSCS-0034)**

**Date: 13 Jan 2020**

**Department of Computer Science**

##### Muhammad Ali Jinnah University

22-E Block 6, PECHS, Karachi-Pakistan

**Table of Contents:**

1. Introduction

2. Project Scope

3. Problem Statement

4. User Manual

4a. Prerequisites

4b. How to use

5. Source Code

6. User Interface

**1. Introduction:**

We have built up a Covid Tracker Application utilizing the ideas of OOP in Java with the prospect that this would assist individuals with getting the insights concerning the quantity of influenced individuals from Covid in the whole world. As Covid is spreading like fire step by step. We believed that it would be an extraordinary thought to make an application which gives us the insights regarding the number influenced nations around the globe.

* This Application is liberated from cost.
* No actual space utilization.
* This Application also check if you are affected by corona virus or not.
* An exquisite UI.

**2. Project Scope:**

Various applications are created for various purposes. Consequently, it is a need to characterize the extent of Covid Tracker to make it more intelligible and help comprehend its particular objectives and use's point of view.

* User can check no. of influenced individuals around the globe anyplace, whenever.
* User can likewise run a test on himself to check his manifestations.

**3. Problem Statement:**

Coronavirus is spreading at an uncommon rate all around the world. Medical care frameworks are encountering a huge measure of pressing factor and wellbeing experts work on debilitating movements to manage the interest. The infection doesn't recognize boundaries, races, or religion! It is, accordingly, our obligation to be joined together and face this profoundly infectious sickness.

The world has effectively a few comparable pandemics to COVID-19 anyway there is an outstanding contrast this time: Mass coordinated effort, straightforwardness and open conversation were not accessible at that point! As tech tenderfoots, it has now gotten conceivable to intentionally add to battling the infection, which is exactly what this project is about.

**4. User Manual:**

Welcome to the Covid Tracker User Manual.

**4a. Prerequisites:**

* **Software:** Intellij Idea
* **Java Version:** 15
* **Platform:** Computer or Laptop

**4b. How to use:**

* First install latest version of Intellij idea with JDK 15
* After installing open Covid Tracker project on Intellij and run it
* Then you can answer the questions that Covid Tracker ask you to check your symptoms.
* Then go to your browser and search **http://localhost:8080/**
* Then you can see the UI
* If you want to search a country just press **ctrl + F** and type country name.

**5. Source Code:**

**Main Class:**

package io.TBZ.CovidTracker;  
  
import io.TBZ.CovidTracker.Form.form;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.scheduling.annotation.EnableScheduling;  
  
@SpringBootApplication  
@EnableScheduling  
public class CovidTrackerApplication  
{  
 public static void main(String[] args)  
 {  
 SpringApplication.*run*(CovidTrackerApplication.class, args);  
 form f1 = new form();  
 f1.assessment();  
 }  
}

**Service Class:**

package io.TBZ.CovidTracker.Services;  
  
import io.TBZ.CovidTracker.models.LocationStats;  
import org.apache.commons.csv.CSVFormat;  
import org.apache.commons.csv.CSVRecord;  
import org.springframework.scheduling.annotation.Scheduled;  
import org.springframework.stereotype.Service;  
import javax.annotation.PostConstruct;  
import java.io.IOException;  
import java.io.StringReader;  
import java.net.URI;  
import java.net.http.HttpClient;  
import java.net.http.HttpRequest;  
import java.net.http.HttpResponse;  
import java.util.ArrayList;  
import java.util.List;  
  
  
@Service  
public class CoronaVirusDataService {  
 private static String *Confirmed\_Cases\_URL* = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_confirmed\_global.csv";  
 private static String *Confirmed\_Deaths\_URL* = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_deaths\_global.csv";  
 private static String *Confirmed\_Recovered\_URL* = "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_recovered\_global.csv";  
  
 private List<LocationStats> allStats\_CC = new ArrayList<>();// created an array list for locations.  
 private List<LocationStats> allStats\_CD = new ArrayList<>();  
 private List<LocationStats> allStats\_CR = new ArrayList<>();  
  
  
 public List<LocationStats> getAllStats\_CC() {  
 return allStats\_CC;  
 }  
  
 public List<LocationStats> getAllStats\_CD() {  
 return allStats\_CD;  
 }  
  
 public List<LocationStats> getAllStats\_CR() {  
 return allStats\_CR;  
 }  
  
 public static String getConfirmed\_Cases\_URL() {  
 return *Confirmed\_Cases\_URL*;  
 }  
  
 public static String getConfirmed\_Deaths\_URL() {  
 return *Confirmed\_Deaths\_URL*;  
 }  
  
 public static String getConfirmed\_Recovered\_URL() {  
 return *Confirmed\_Recovered\_URL*;  
 }  
  
 @PostConstruct //telling program to execute this method when you construct this service.  
 @Scheduled(cron = "\* \* 5 \* \* \*") // scheduling program to run on fifth hour of everyday.  
 public void fetchVirusData() throws IOException, InterruptedException // Added Exceptions so that if client send fails the program won't crash  
 {  
 List<LocationStats> newStats\_CC= new ArrayList<>(); // i made this list for concurrency because I dont want people to get error while we are updating this.  
 List<LocationStats> newStats\_CD= new ArrayList<>();  
 List<LocationStats> newStats\_CR= new ArrayList<>();  
 // For Confirmed Cases  
 HttpClient client\_CC = HttpClient.*newHttpClient*(); // Http call  
 HttpRequest request\_CC = HttpRequest.*newBuilder*() // Creating a Http Request  
 .uri(URI.*create*(*Confirmed\_Cases\_URL*)) // Creating URI  
 .build();  
 HttpResponse<String> httpResponse\_CC = client\_CC.send(request\_CC, HttpResponse.BodyHandlers.*ofString*()); // sending Request and returning body as a string  
 StringReader csvBodyReader\_CC = new StringReader(httpResponse\_CC.body());  
  
 // For Confirmed Deaths  
 HttpClient client\_CD = HttpClient.*newHttpClient*(); // Http call  
 HttpRequest request\_CD = HttpRequest.*newBuilder*() // Creating a Http Request  
 .uri(URI.*create*(*Confirmed\_Deaths\_URL*)) // Creating URI  
 .build();  
 HttpResponse<String> httpResponse\_CD = client\_CD.send(request\_CD, HttpResponse.BodyHandlers.*ofString*()); // sending Request and returning body as a string  
 StringReader csvBodyReader\_CD = new StringReader(httpResponse\_CD.body());  
  
 // For Confirmed Recovered  
 HttpClient client\_CR = HttpClient.*newHttpClient*(); // Http call  
 HttpRequest request\_CR = HttpRequest.*newBuilder*() // Creating a Http Request  
 .uri(URI.*create*(*Confirmed\_Recovered\_URL*)) // Creating URI  
 .build();  
 HttpResponse<String> httpResponse\_CR = client\_CR.send(request\_CR, HttpResponse.BodyHandlers.*ofString*()); // sending Request and returning body as a string  
 StringReader csvBodyReader\_CR = new StringReader(httpResponse\_CR.body());  
  
 // Parsing..  
  
 Iterable<CSVRecord> records\_CC = CSVFormat.*DEFAULT*.withFirstRecordAsHeader().parse(csvBodyReader\_CC);  
 Iterable<CSVRecord> records\_CD = CSVFormat.*DEFAULT*.withFirstRecordAsHeader().parse(csvBodyReader\_CD);  
 Iterable<CSVRecord> records\_CR = CSVFormat.*DEFAULT*.withFirstRecordAsHeader().parse(csvBodyReader\_CR);  
  
 for(CSVRecord record : records\_CC)  
 {  
 LocationStats locationStat\_CC = new LocationStats();  
 locationStat\_CC.setState(record.get("Province/State"));  
 locationStat\_CC.setCountry(record.get("Country/Region"));  
 int latestCases = Integer.*parseInt*(record.get(record.size() - 1));  
  
 int prevDayCases = Integer.*parseInt*(record.get(record.size() - 2));  
  
 locationStat\_CC.setConfirmedCases(latestCases);  
 locationStat\_CC.setDiffFromPrevDay(latestCases - prevDayCases);  
// System.out.println(locationStat\_CC);  
 newStats\_CC.add(locationStat\_CC);  
 }  
  
 for(CSVRecord record : records\_CD)  
 {  
 LocationStats locationStat\_CD = new LocationStats();  
 locationStat\_CD.setState(record.get("Province/State"));  
 locationStat\_CD.setCountry(record.get("Country/Region"));  
 locationStat\_CD.setConfirmedDeaths(Integer.*parseInt*(record.get(record.size() - 1)));  
// System.out.println(locationStat\_CD);  
 newStats\_CD.add(locationStat\_CD);  
 }  
  
 for(CSVRecord record : records\_CR)  
 {  
 LocationStats locationStat\_CR = new LocationStats();  
 locationStat\_CR.setState(record.get("Province/State"));  
 locationStat\_CR.setCountry(record.get("Country/Region"));  
 locationStat\_CR.setConfirmedRecovered(Integer.*parseInt*(record.get(record.size() - 1)));  
// System.out.println(locationStat\_CR);  
 newStats\_CR.add(locationStat\_CR);  
 }  
  
 this.allStats\_CC = newStats\_CC;  
 this.allStats\_CD = newStats\_CD;  
 this.allStats\_CR = newStats\_CR;  
  
 }  
}

**Home Controller Class:**

package io.TBZ.CovidTracker.controllers;  
  
  
import io.TBZ.CovidTracker.Services.CoronaVirusDataService;  
import io.TBZ.CovidTracker.models.LocationStats;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
  
import java.util.List;  
  
@Controller  
public class HomeController extends CoronaVirusDataService

{  
 @Autowired  
 CoronaVirusDataService coronaVirusDataService;  
 @GetMapping("/")  
 public String home(Model model){  
  
  
 List<LocationStats> allStats\_cc = coronaVirusDataService.getAllStats\_CC();  
 int totalReportedCases = allStats\_cc.stream().mapToInt(stat -> stat.getConfirmedCases()).sum();  
 int totalNewCases = allStats\_cc.stream().mapToInt(stat -> stat.getDiffFromPrevDay()).sum();  
  
  
 model.addAttribute("locationStats" , allStats\_cc);  
 model.addAttribute("totalReportedCases" , totalReportedCases);  
 model.addAttribute("totalNewCases" , totalNewCases);  
// model.addAttribute("locationStats" , coronaVirusDataService.getAllStats\_CD());  
// model.addAttribute("locationStats" , coronaVirusDataService.getAllStats\_CR());  
// model.addAttribute("testName" , "TEST");  
 return "home";  
 }  
  
  
}

**Home.html:**

<!DOCTYPE html>  
  
<html xmlns:th="http://www.thymeleaf.org">  
  
<head>  
 <title>Coronavirus Tracker Application</title>  
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">  
  
  
  
</head>  
  
  
<body class="text-dark bg-light" style="font-family:cursive;">  
  
  
  
<div style="border:2px solid black;background-color:#101010;" class="container mt-5">  
<!--background-size:'cover' -->  
<!-- -->  
<!-- background-repeat:no-repeat;-->  
  
  
 <div style="background-color:#ffea00;" class="container-fluid mt-3">  
 <div class="row">  
 <div class="col-md-3">  
 <img style="border:2px solid black" src="https://lh3.googleusercontent.com/plQnUf0aq-vGnkr0Geh\_QA6Wi\_QIh9p14bne6SBK27D314E82NvZQfuAXvEZ\_GbiaQ" width="150px" height="150px" />  
  
 </div>  
 <div class="col-md-9 mt-5">  
  
 <h2 style="font-size:30px">CORONA VIRUS TRACKER APPLICATION</h2>  
<!-- font-weight:300px-->  
<!-- font-family:Helvetica-->  
  
 </div>  
  
  
 </div>  
  
 <p class="" style="background-color:#ffea00;font-size:20px" align="right">Developed By: TBZ</p>  
 </div>  
  
  
  
<div>  
  
  
<marquee style="background-color:#ffea00" class="text-dark">This Application lists the current number of cases reported across the World</marquee>  
<!--<p th:text="${testName}"></p> -->  
  
</div>  
  
<div style="border:2px solid black;" class="jumbotron">  
  
 <h1 class="display-4" th:text="${totalReportedCases}"></h1>  
 <p class="lea d">Total Cases Reported as of Today.</p>  
 <hr class="my-4">  
 <p>  
 <span>New Cases reported since Previous Day.</span>  
 <span th:text="${totalNewCases}"></span>  
 </p>  
</div>  
  
<div style="border:2px solid black;background-color:#ffea00" class="container">  
 <table class="table">  
 <tr>  
 <th>State</th>  
 <th>Country</th>  
 <th>Total Confirmed Cases</th>  
 <th>Changes since Last Day</th>  
<!-- <th>Total Confirmed Deaths</th>-->  
<!-- <th>Total Recovered Cases</th>-->  
 </tr>  
 <tr th:each="locationStat : ${locationStats}">  
 <td th:text="${locationStat.state}"></td>  
 <td th:text="${locationStat.country}"></td>  
 <td th:text="${locationStat.ConfirmedCases}"></td>  
 <td th:text="${locationStat.diffFromPrevDay}"></td>  
<!-- <td th:text="${locationStat.ConfirmedDeaths}"></td>-->  
<!-- <td th:text="${locationStat.ConfirmedRecovered}"></td>-->  
 </tr>  
 </table>  
</div>  
</div>  
  
  
</body>  
  
</html>

**Form Class:**

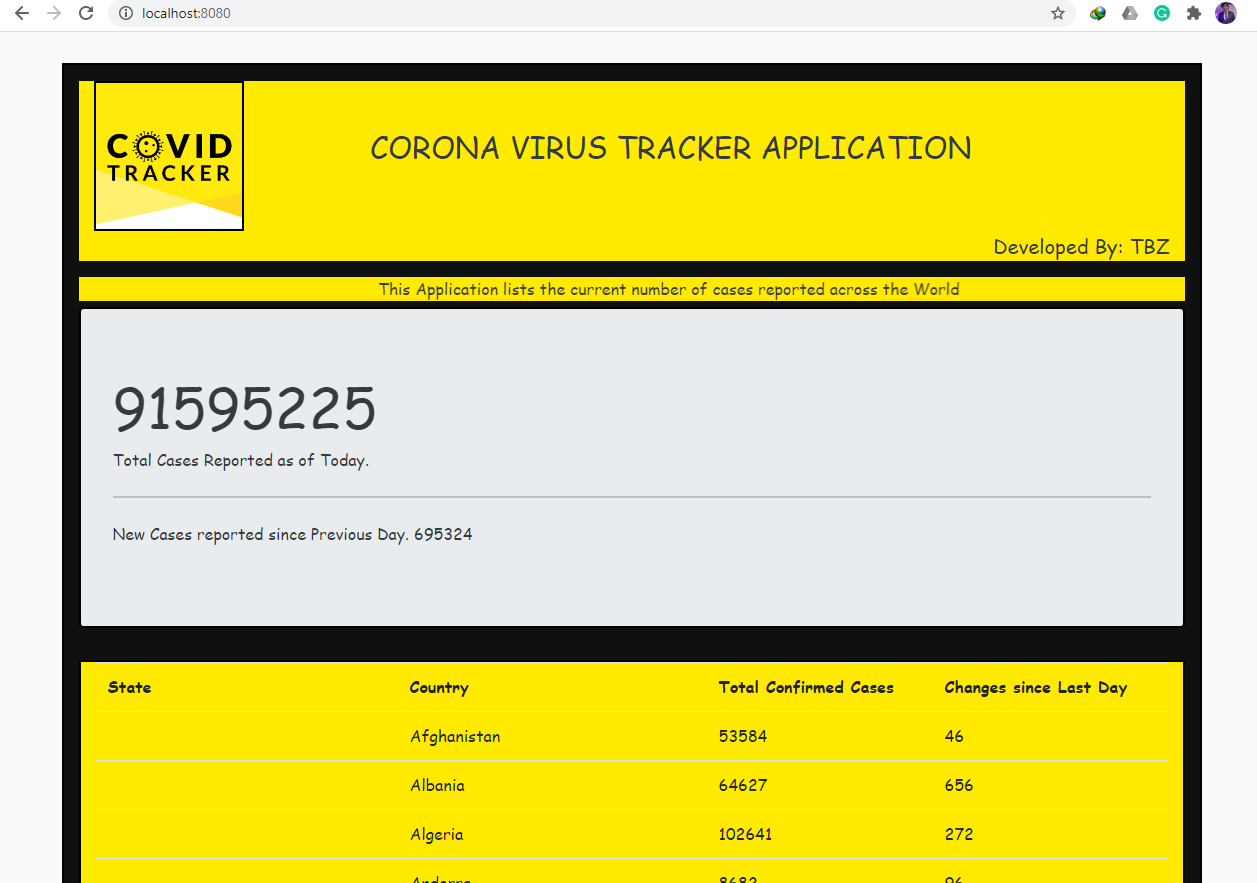
package io.TBZ.CovidTracker.Form;  
  
import java.util.Scanner;

abstract class Test  
{  
 int ycount=0, ncount=0;  
 long age = 0;  
 String gender = null;  
 String name1 = null;  
 char ch = 0;  
 public abstract void assessment();  
}

public class form extends Test

{  
 public void assessment(){  
 int ycount=0,ncount=0;  
 long age;  
 String gender;  
 String name1;  
 char ch;  
 Scanner sc = new Scanner(System.*in*);  
  
  
 System.*out*.println("--------------------------------------------------------------");  
 System.*out*.println("\n");  
 System.*out*.println("\n CORONA TESTING FORM FOR USER");  
  
 System.*out*.println("=============================================================");  
 System.*out*.println("Enter your age: ");  
 age = sc.nextLong();  
 System.*out*.println("What is your Name? ");  
 name1 = sc.next();  
 System.*out*.println("What is your Gender? (male/female/other): ");  
 gender = sc.next();  
  
 System.*out*.println("Do you have Severe Dry cough, Fever or Tiredness? (y/n) ");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
 System.*out*.println("Do you have Severe and constant pain or pressure in the chest? (y/n) ");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
 System.*out*.println("Extreme difficulty breathing (such as gasping for air, being unable to talk without catching your breath, severe wheezing, nostrils flaring) (y/n): ");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
 System.*out*.println("Do you feel Unconscious or is it sometimes very difficult to wake up (y/n):");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
 System.*out*.println("Any Signs of low blood pressure (too weak to stand, dizziness, lightheaded, feeling cold, pale, clammy skin) (y/n): ");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
 System.*out*.println("Dehydration (dry lips and mouth, not urinating much, sunken eyes) (y/n): ");  
 ch = sc.next().charAt(0);  
  
 if(ch == 'y' || ch == 'Y')  
 {  
 ycount+=1;  
 }  
  
 else if(ch == 'n' || ch == 'N')  
 {  
 ncount+=1;  
 }  
  
 else  
 {  
 System.*out*.println("You did not enter correct option.");  
 }  
  
  
 if(ycount >= 4){  
  
 System.*out*.println("=============================================================");  
 System.*out*.println("Name: " + name1);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Gender: " + gender);  
 System.*out*.println("\nFinal Report: ");  
 System.*out*.println("Based on your symptoms, you have to isolate yourself, \nyou may need urgent medical care. Please go to the nearest emergency department.");  
 System.*out*.println("=============================================================");  
  
 }  
  
 else if(ycount == 3 && ncount == 3){  
  
 System.*out*.println("=============================================================");  
 System.*out*.println("Name: " + name1);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Gender: " + gender);  
 System.*out*.println("\nFinal Report: ");  
 System.*out*.println("As per your assessment, it is recommended that you have a Corona virus test, \nBut its upto your choice whether you want to have a test or not, \nbut WE Recommend you to have one.");  
 System.*out*.println("=============================================================");  
  
 }  
  
 else {  
  
 System.*out*.println("=============================================================");  
 System.*out*.println("Name: " + name1);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Gender: " + gender);  
 System.*out*.println("\nFinal Report: ");  
 System.*out*.println("No COVID-19 testing needed at this time\nBased on the answers given, you do not need to get tested.\nLearn more about COVID-19 and steps you can take to protect yourself and others");  
 System.*out*.println("=============================================================");  
  
 }  
 }  
}

**6. User Interface:**

****

