Covid-19 Live Cases Tracker Application

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF COMPUTER APPLICATION

by

Mohd Amir

(Enrolment No - 190029014005248)

Under the Supervision of
Dr. Amit Kumar Gupta
KIET Group of Institutions, Ghaziabad

to the

Faculty of Computer Applications
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW
(Formerly Uttar Pradesh Technical University)

July, 2021

DECLARATION

I hereby declare that the work presented in this report entitled “Covid-19 Live Cases Tracker”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: Mohd Amir

Roll. No.: 1900290149063

Branch: Master of Computer Application

(Candidate Signature)

CERTIFICATE

Certified that **Mohd Amir** (**1900290149063**) has carried out the project work presented in this project report entitled "**Covid-19 Live Cases Tracker**" for the award of Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University, Lucknow under my supervision. The report embodies result of original work, and studies are carried out by the student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University.

Dr. Amit Kumar Gupta **External Examiner**
Assistant Professor
Dept. of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Ajay Kumar Shrivastava
Professor & Head
Department of Computer Application
KIET Group of Institutions, Ghaziabad

Date:

Covid-19 Live Cases Tracker

Mohd Amir

ABSTRACT

Covid-19 Live Data Tracker of Covid19India is web based application for case evaluation. Live Data Tracker of Covid19India not only replaces paperwork but also releases the workload of people. It fulfills the requirements of the people who are track live data of covid19india online. The people can track live data of covid19 by using internet without the need of going to any physical destination. They can view the result at the same time. Thus the purpose of the site is to provide a system that keeps social distancing “Stay Safe at Home”.

The project is going to fetch the live data from the API using Async/Await syntax. In our SFDC applications we often need to retrieve data from external APIs so it can be displayed in our web page. User interface of the app is created using Material-UI, It is an open-source project that features Lightning components that implement Google’s Material Design. Data Visualization is done using json file. Json file is one of the most popular language which is used for deserialized data.

Finally the project is deployed online through Salesforce, it provides free hosting to your website or application. The user can access the website online and can know about the live data of Covid19India.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, Dr. Amit Kumar Gupta for her guidance, help and encouragement throughout my research work.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Mohd Amir

1900290149063

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the year 2020 we are dealing with unfamiliar situation which is corona virus pandemic. Corona virus is a disease started from China and spread throughout the world. The corona virus is affecting 210 countries and territories around the world. As a developer we can track the live data of this pandemic across the India as well as in each State. This will help in providing awareness about its impact that we can face and also the necessary measures to be taken before it's out of control. The developed project is a web application which deals in tracking the live data of Covid19Indiapandemic across the India and gives the data visualization for better understanding.

This Web Application is developed using the latest technology Salesforce.in which use Apex, Visual Force and Lightning Components. Visual Force is a framework which is very helpful in developing the user interface of the web application. Through this framework we can develop the project in small reusable parts which is called components. To start the project we need the data of confirmed cases, recovered cases and the death cases to display to the user on the web page, for this we are going to fetch the data from API using the Async/Await syntax in Json. The API used for this project is https://sahil-covid19india-developer-edition.ap17.force.com/, after fetching the data we are going to display the covid-19 live data of confirmed cases, recovered cases and the death cases around the India and total number of cases in each affected State. For better understanding we have data visualization of the cases in the form of rows.Basically the project provides the awareness about the corona virus pandemic, we can get the information about the daily arising cases and precautions to be taken to stay safe with this pandemic.

After completing the project we have deployed the project online so that the user can access the application from anywhere and can stay up to date about the live data of corona virus pandemic.

IDENTIFICATION OF PROBLEM AND ISSUES

My first identification and problem was to collect the data, how we can get the live statistics of confirmed cases, the number of recovered cases and the number of death cases whole over the India, number of cases in different States, so the solution for this is to collect the data from API. To retrieve the data we need to make API request from the database so that we can display in our application. API is basically the intermediary between the developer and the database. To fetch the data from the API we have used the Lightning Component in our project.

My second approach is to display the data in our web page using Visual Force web framework, it will provide to develop the user interface of the application. To make the user interface attractive we have used material UI in our application, it is a library created by Google which provide simple and attractive look and feel to the user. Also the data visualization is done using the Json in our Salesforce project, the data visualization is made with the help of SFDC.

My third approach is to make the project live so that the user can access the application, so the solution is to use the SDFC which provide the free hosting to your project. Hosting your project on SFDC is easy and free for developer edition, it also support the continuous development through github, so that if any changes are made in the project they can immediately reflect on the live website.

SETTING UP THE PROJECT

The First of all, let me tell you what is Salesforce and what it actually does. So Salesforce is a customer relationship management solution that brings companies and customers together. It's one integrated CRM platform that gives all your departments — including marketing, sales, commerce, and service — a single, shared view of every customer. which you can build dynamic and faster applications. Also, it is really flexible. Therefore you can introduce it to an already existing code base without having to rewrite. More specifically I would say force.com is platforms which helps you to build user interfaces for the web. So let’s see how to set up the environment for our first Salesforce app.

COMPONENTS OF PROJECT

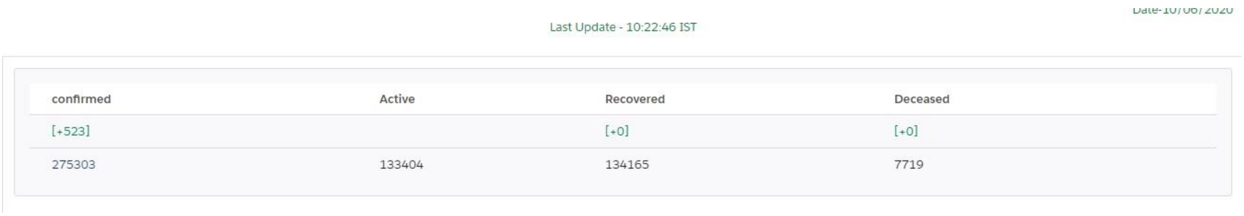
Components are the building blocks of any Salesforce app and a typical Salesforce app will have many of these. Simply put, a component is a Visual Force framework or function that optionally accepts inputs i.e. properties(props) and returns a Salesforce element that describes how a section of the UI (User Interface) should appear.

Navbar

Navigation bar is the component developed using Lightning framework, for this we need to copy the layout by using the Lighting Component Library. Navigation bar will have routing or we can say navigating the user to same page within the app. The project in Salesforce is a single page application that means the project does not reload while navigating to the same page. The navbar will have Home page.

Header

Header will display the API data in our application. Header is displayed using the Material-UI library which is basically a lightning component library. The header are displayed in our project to show the number of confirmed cases, the number of recovered cases and the number of death cases whole over the India. Green color numbers are for infected number of daily cases, recovered number of cases and the number of death cases. Rows also display when the data of API got updated.



Date-10/06/2020

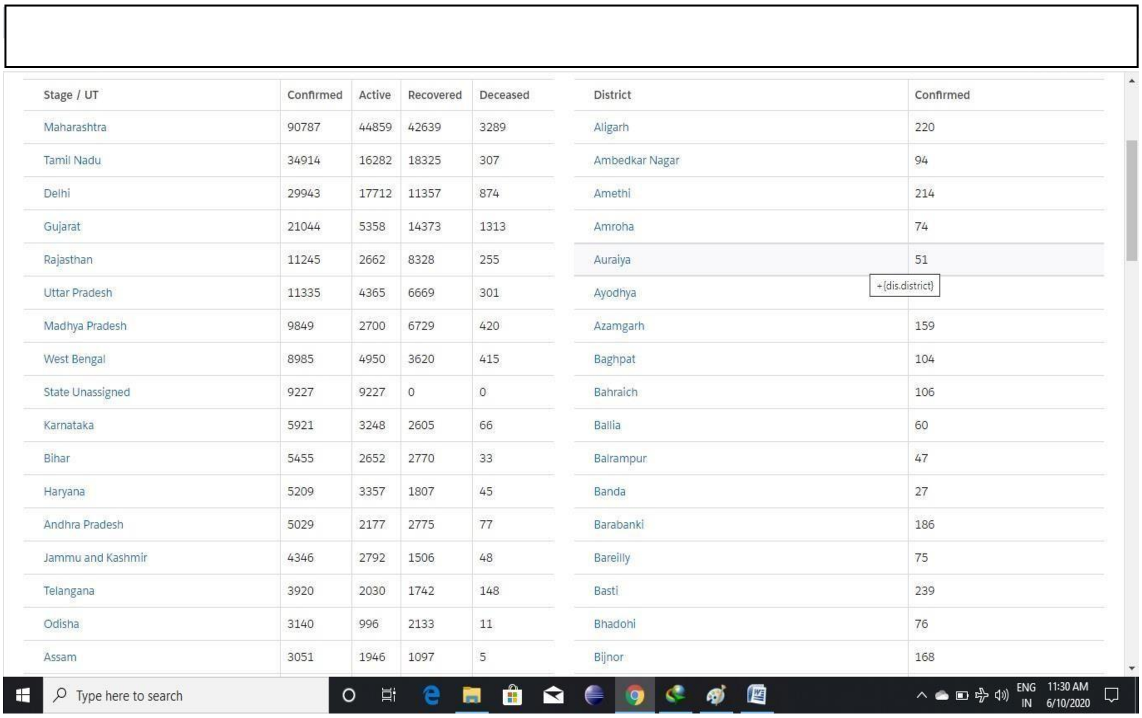
Last Update - 10:22:46 IST

confirmed	Active	Recovered	Deceased
[+523]		[+0]	[+0]
275303	133404	134165	7719

FIG.1.1 HEADER DATA

Body

Body component will display the data visualization of API for better understanding of the data. Body layout library is used for displaying the number rows lines which have to state wise data and as well as district wise data displayed in the project. The rows show the historic data across the state wise and district wise.



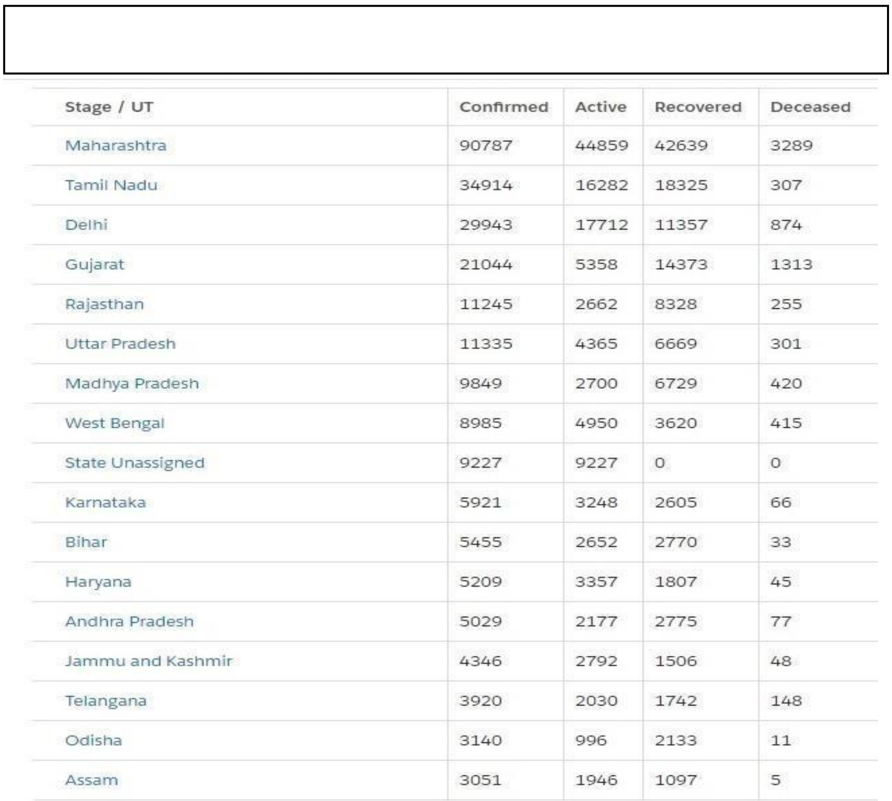
Stage / UT	Confirmed	Active	Recovered	Deceased
Maharashtra	90787	44859	42639	3289
Tamil Nadu	34914	16282	18325	307
Delhi	29943	17712	11357	874
Gujarat	21044	5358	14373	1313
Rajasthan	11245	2662	8328	255
Uttar Pradesh	11335	4365	6669	301
Madhya Pradesh	9849	2700	6729	420
West Bengal	8985	4950	3620	415
State Unassigned	9227	9227	0	0
Karnataka	5921	3248	2605	66
Bihar	5455	2652	2770	33
Haryana	5209	3357	1807	45
Andhra Pradesh	5029	2177	2775	77
Jammu and Kashmir	4346	2792	1506	48
Telangana	3920	2030	1742	148
Odisha	3140	996	2133	11
Assam	3051	1946	1097	5

District	Confirmed
Aligarh	220
Ambedkar Nagar	94
Amethi	214
Amroha	74
Auraiya	51
Ayodhya	+{dis.district}
Azamgarh	159
Baghpat	104
Bahraich	106
Ballia	60
Balrampur	47
Banda	27
Barabanki	186
Bareilly	75
Basti	239
Bhadohi	76
Bijnor	168

FIG.1.2 BODY DATA

State Picker

This component is used for displaying the data of different state around the India. In this component a list of states appear so that we can select the desire state which we are looking for district wise data.



Stage / UT	Confirmed	Active	Recovered	Deceased
Maharashtra	90787	44859	42639	3289
Tamil Nadu	34914	16282	18325	307
Delhi	29943	17712	11357	874
Gujarat	21044	5358	14373	1313
Rajasthan	11245	2662	8328	255
Uttar Pradesh	11335	4365	6669	301
Madhya Pradesh	9849	2700	6729	420
West Bengal	8985	4950	3620	415
State Unassigned	9227	9227	0	0
Karnataka	5921	3248	2605	66
Bihar	5455	2652	2770	33
Haryana	5209	3357	1807	45
Andhra Pradesh	5029	2177	2775	77
Jammu and Kashmir	4346	2792	1506	48
Telangana	3920	2030	1742	148
Odisha	3140	996	2133	11
Assam	3051	1946	1097	5

FIG.1.3 STATE PICKER

Footer

Footer of the application is simple and responsive in design which include cloud technology used in the project

@Powerd by SalesForce

FIG.1.4 FOOTER

Project Scope

- Covid-19 India cases are increasing every day and may reduce at certain period of time, project will help in tracking the live data across the India.
- Stay up to date about the Covid19India cases.
- Provide awareness about corona virus.
- We can track the real-time data in different state of the India.
- Data visualization will help in tracking the data.

Hardware and Software used in Project Table

Table 1.1 Hardware Used

Hardware	**Configuration**
Processor	Intel Pentium G1 54000 clocked at 2.6 GHz
Ram	8GB DDR4
Monitor	HP Backlit 19’’LED
Hard disk	Western Digital 1TB
Keyboard	HP Standard 104 keys

Table 1.2 Software Used

Software	**Configuration**
Operating System	Windows 10
Language	Salesforce
Code Editor	Visual Studio Code

Technologies Description

Table 1.3 Technologies used

Front-End	Visual Force, CSS, JavaScript
Back-End	Apex
Web Framework	Lightning Components
Deployment	Force.com

Frond-End:

The front-end is everything involved with what the user sees, including design and some languages like HTML and CSS. Front-end developers are responsible for a website's user-facing code and the architecture of its immersive user experiences. In order to execute those objectives, front-end developers must be adept at three main languages: HTML, CSS, and JavaScript programming. In addition to fluency in these languages, front-end developers need to be familiar with frameworks like Visual Force, Foundation, Backbone, Lightning Component Library, which ensure great-looking content no matter the device, and which package code into a more useful, time-saving form.[2]

Back-End:

So what makes the front end of a website possible? Where is all that data stored? This is where the back end comes in. The back end of a website consists of a server, an application, and a database. A back-end developer builds and maintains the technology that powers those components which, together, enable the user-facing side of the website to even exist in the first place.

About Apex, one of the most interesting developments recently gains popularity in the server side language. It is a framework for developing high-performance, concurrent programs that don't rely on the main stream multithreading approach but use asynchronous I/O with an event-driven programming model.[5]

In this project our real-time data is collected from API. In our Salesforce applications we often need to retrieve data from external APIs so it can be displayed in our web pages. An API (Application Programming Interface) is a software interface that allows computers to request, retrieve, and exchange data and information in a standardized way. An API specifies:[5]

- What information can be requested
- How to make the requests
- What results are returned in response to requests

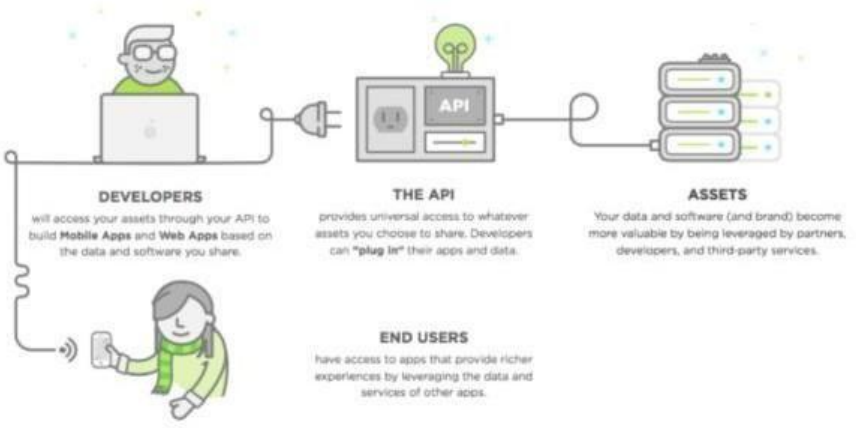

API
DEVELOPERS
will access your assets through your API to build **Mobile Apps** and **Web Apps** based on the data and software you share.
THE API
provides universal access to whatever assets you choose to share. Developers can **"plug in"** their apps and data.
ASSETS
Your data and software (and brand) become more valuable by being leveraged by partners, developers, and third-party services.
END USERS
have access to apps that provide richer experiences by leveraging the data and services of other apps.

FIG 1.5 API Request

Web Framework:

Lightning Aura Component is one of the popular web frameworks that has gained importance over other frameworks such as HTML, JavaScript , etc.. This is because of its implementation of Virtual DOM, whose primary objective is to enhance the overall performance of the application. However, there are certain things that one has to keep in mind before designing the applications. Failing to anticipate the problems that may occur component hierarchy will lead to performance degradation. Some of the commonly faced problems are component re-rendering, application lag due to background computations being run, lag due to processing large data sets in a single stretch, etc[1]

Lightning Component is a library for building user interfaces. Visual Force got that name for a reason. It’s a component-based JavaScript library that renders smartly and can seriously simplify your work. SFDC can be used as a base in the development of single-page or mobile applications. However, SFDC is only concerned with rendering data to the DOM, and so creating Salesforce applications usually requires the use of additional libraries for state management and routing. Lightning code is made of entities called components. Components can be rendered to a particular element in the DOM using the Aura Component DOM library. When rendering a component, one can pass in values that are known as "props".

Deployment:

While software engineering has so far mainly focused on software development, software deployment is now emerging as a new research field. Software (or application) deployment is a complex process, which covers all the activities that have to be carried out, from the end of the development itself, on the producer sites, to the actual installation and maintenance of the application on the consumers' computers. The main purpose of this part of the app is to provide information. We deploy this part of our code on a web server, so that the user can access the required information. Deployment involves packaging up your web application and putting it in a production environment that can run the app. Your web application must live somewhere other than your own desktop or laptop. A production environment is the canonical version of your current application and its associated data. In our project we have use Force.com for deploying our application. Deployment is done using Force.com, that offers hosting and server less backend services for web applications and static websites.[6]ApplicationURL:https://amir-covid19india-developer-edition.ap17.force.com/

CHAPTER 2

LITERATURE REVIEW

As of late, the significance of moderate access to dependable equipment, programming Assets and minimizing the support expenses and security concerns have energized vast foundation supervisors and organizations to relocate to distributed computing i.e. to cloud computing. The cloud services are running the world at a good pace. The paper presents the two services of cloud that is salesforce.com and Force.com and also the functionality of these services by introducing a ticket booking desktop application for metro rail users using Salesforce.com and Force.com. Salesforce.com is a suite of CRM (Customer Relationship Management) tools which are designed to help big and small businesses dramatically improving their customer service, retention rates, purchase analysis and much, much more. Salesforce provides both SaaS and PaaS cloud services. Force.com is the platform provided by Salesforce, is used for application development[1].

With the appearance of distributed computing, associations are hoping to move their Customer Relationship Management (CRM) applications from an On-Premise environment or we can say local servers to an On-Demand environment that is on cloud server. On-Premise environment is when association has the framework and programming inside their system. In On-Demand environment, outsider has the base and programming and charges the relationship in light of its participation show. Salesforce is the principle On-Demand CRM thing [2].

The APEX code generation tool for the Ada language is described. APEX provides the programmer with three different, yet consistent, graphical views of the Ada program under development. Once the graphical specification has been developed, Ada code can be generated from APEX. In addition, APEX can reverse engineer the graphical specification from existing Ada code in order to support software reuse. A brief description of the three APEX editors is given, and an illustration of each editor and the resulting code generated is provided [3].

Engineering a program in the context of performance aspect exhibits a crucial part, as it minimizes the time and space complexity. If the space complexity in a program is minimized, the application runs on minimum memory. Many JAVA developers write JAVA programs without consideration towards optimized execution. Developers of application should tune the application(s) earlier to use in production. Application code tuning often engenders massive performance enhancement. The proposed model is envisioned to aid the JAVA programmers to tune and enhance the JAVA based application(s). This paper elucidates miscellaneous techniques to escalate JAVA application program performance and can serve as an optimization tool for the JAVA application programmers. Our experimental denouements designate that performance i.e. Time and Space complexities are enhanced [4].

Motor task experiments play an essential role in exploring the brain mechanisms of movement control, and visual force-feedback is an important factor in these motor experiments. In this paper, the authors proposed a visual force-feedback system suitable for neuroscience experiment. With this system, the force output produced by participants can be detected and recorded in real time, while force output was visually displayed as a feedback cue to the participants simultaneously. Furthermore, this force feedback system is MRI compatible, and can be used both in FMRI and ERP experiments. The proposed system has been applied in hand-grip tasks and finger movement experiments, which were designed to explore the relationship between force output and brain activation mode in normal subject and stroke patient. The results demonstrated that various force levels were well detected and visual feedback signals enabled the accomplishment of experiments with both fixed and variable target force levels [5].

We present SOQL (social networks query language), a new language for querying and creating data in social networks. The language is designed to meet the growing need of social networks participants to efficiently manage the large, and quickly growing, amounts of data available to them, as well as automate processes of creating new data. This need is increasingly pressing as social networks gradually become an important working tool for business development and management. SOQL is a step in the direction of meeting the challenges of providing an expressive querying mechanism and automating processes in social networks. SOQL is an SQL-like language which enables the user to retrieve paths to other participants in the network, and use a retrieved path in order to attempt to create a connection with the participant at the end of the path. The language can specify complex conditions that a desired path should satisfy. The language also supports retrieving a group of participants which satisfy conditions as a group, and connecting its members to each other. SOQL uses the path and group as data types. This work presents the SOQL language and discusses implementation issues [6].

There are many technical procedures to analyze and evaluate raw source code of any software. However, most of them focus on productivity and efficiency of the code rather than the concealed historical value, especially with the legacy software and programs. On the other hand, preservation of digital and technological collection has increasingly been one of the major concerns in certain places like museum and science center. Such activity, therefore, includes the exploration of digital collection and extraction of the undiscovered or disappearing knowledge. This study displays an effort to experimentally use topic modeling approaches to extract interesting facts and knowledge within a special kind of digital-born collection, software's source code. By exercising multiple techniques with selected subjects, this study also estimates the possibility to create a primitive framework to investigate digital collection, such as codebase, for further use in museum context [7].

Currently, a large amount of data is amassed in electronic health records (EHRs). However, EHR systems are largely information silos, that is, uses of these systems are often confined to management of patient information and analytics specific to a clinician's practice. A growing trend in healthcare is combining multiple databases to support epidemiological research. The College Health Surveillance Network is the first national data warehouse containing EHR data from 31 different student health centers.

Each member university contributes to the data warehouse by uploading select EHR data including patient demographics, diagnoses, and procedures to a common server on a monthly basis. In this paper, we focus on the data quality dimensions from a subsample of the data comprised of over 5.7 million patient visits for approximately 980,000 patients with 4,465 unique diagnoses from 23 of those universities. We examine the data for measures of completeness, consistency, and availability for secondary use for epidemiological research. Additionally, clinical documentation practices and EHR vendor were evaluated as potential contributors to data quality. We found that overall about 70% of the data in the data warehouse is available for secondary use, and identified clinical documentation practices that are correlated to a reduction in data quality. This suggests that automated quality control and proactive clinical documentation support could reduce ad-hoc data cleaning needs resulting in greater data availability for secondary use [8].

Cloud computing has become the industry standard for rapid application deployment, scalable server support, mobile and distributed services, and it provides access to (theoretically) infinite resources. Unfortunately, researchers are still trying to converge towards cross-provider cloud computing frameworks to enable compatibility and seamless resource transition between cloud providers. Moreover, users are restricted to using the provider-specific pre-configured options of resources and services, irrespective of their current needs. At the same time, cloud services are provided as a direct service from the providers to the clients. This creates a segregated cloud market clientele, and non-negotiable pricing strategies for the cloud services. In this paper, we propose Jugo, a generic architecture for cloud composition and negotiated service delivery for cloud users. Jugo acts as a match-maker for service specifications from the users with the currently available assets from the cloud providers. The engagement of a middle-man as an opaque cloud service provider will create a better opportunity for cloud users to find cheaper deals, price-matching and flexible resource specifications, with increased revenue and higher resource utilization for the cloud service providers [9].

In this paper, we present a study exploring the use of API keywords within method summaries. We conducted a web-based study where we asked participants to rank Java method summaries based on five levels of detail, from low level to high level. We found that programmers widely use API in both high and low level summaries. Specifically, we found that 76.78% of higher level summaries contain Java API keywords. Additionally, we found that 93.75% of lower level summaries also contain them. This also shows that, in general, as the detail level decreases, the number of API keywords within the summary increases. It is our hope that this line of research will spark a discussion about API usage outside of source code. It is possible that method summaries are not the only form of documentation that API usage plays an important role. We believe these may be important results that could lead to an improvement for API usability design [10].

In a live programming environment, the state of the running program is available during the editing process. An ideal live programming system should be able to harness the live program to offer improved abilities for code creation and manipulation. We introduce Circa, a language and platform designed to address this need. We argue in favor of a dataflow-based model of computation, and we show how this format enables useful methods of code inspection and manipulation. We present a framework based on the back propagation algorithm that allows the user to manipulate their program by expressing a desire against the program's result. We discuss how these code editing abilities can combine to produce a highly effective environment [11].

CCN is one of the future Internet architecture. However, the lack of real traffic becomes an obstacle to advanced CCN researches. HTTP, as an content-oriented application- layer protocol working over current Internet, is similar to CCN in many aspects. In this demo, we try to convert HTTP traffic into CCN traffic with HTTP-CCN gateway. Although HTTP is not equivalent to CCN, we can design carefully to make the conversion correct in most situations. The gateway will introduce real traffic into CCN testbed to support CCN researches. Our demonstration will show how to use this gateway [12].

Lua API does not contain functions needed for executing HTTP/HTTPS requests. This paper deals with the implementation of support of HTTP/HTTPS protocol for Lua interface. Aim of this work is the extension of Lua API for support of synchronous and asynchronous HTTP requests. This is achieved using C++ programming language and cURL open source library [13].

The growing number of Web pages on the Internet introduces a need to combine and integrate information from HTML tables of different Web pages that contain similar information into a single Web page, especially information from the same domain of interest. This paper presents an approach of HTML table integration by combining several existing methods that are proved to solve different issues in the integration processes. The integration of HTML table consists of three phases: (1) extraction of the structure of the tables; (2) integration of the tables' schema; (3) integration of the data values. To solve the conflicts in semantics and naming in the tables schema, domain-ontology is used. To improve quality of integration of data values in the tables, the vector space model is used to check the duplications of data values. As the integration result, a single HTML table is obtained. The approach is implemented on an engine built using Python. Results of the experiment shows that the engine can successfully integrate two HTML tables into single table [14].

Interpretation has been a promising approach for static analysis of Java Script programs. Static analysis is used for security auditing, debugging, optimization and error checking. Java Script is dynamically typed, uses prototype-based inheritance and first class functions. It supports reflective calls, access to object fields and allows object fields to be dynamically added and deleted. These dynamic features make Java Script flexible to use. At the same time, they make Java Script applications more susceptible to programming errors. The challenge that comes with the analysis of such programs is the design of abstract domains that will precisely track properties of interest without affecting performance. This paper presents our work on improving analysis precision of Java Script programs. We used an extended domain of intervals to track ranges of numeric values of variables. This is the first time interval domain has been applied to the analysis of the full Java Script language. We implemented the new abstract domain

Within a Java Script abstract interpreter. Our experiments show that the new abstract domain enables the abstract interpreter to infer more precise type information for most of the benchmark programs and strikes a good balance between analysis precision and cost. While the analysis of some benchmarks take more time as expected, some other benchmarks actually take less time [15].

CHAPTER 3

FEASIBILITY STUDY

Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggest done?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: Ø Is there sufficient support for the management from the users? Will the system be used and work properly if it is being developed and implemented? Will there be any resistance from the user that will undermine the possible application benefits? This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.

The benefits in the form of reduced costs or fewer costly errors. Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

CHAPTER 4

DESIGN

The design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language. The software design process can be divided into the following three levels of phases of design

1. Interface Design

2. Architectural Design

3. Detailed Design

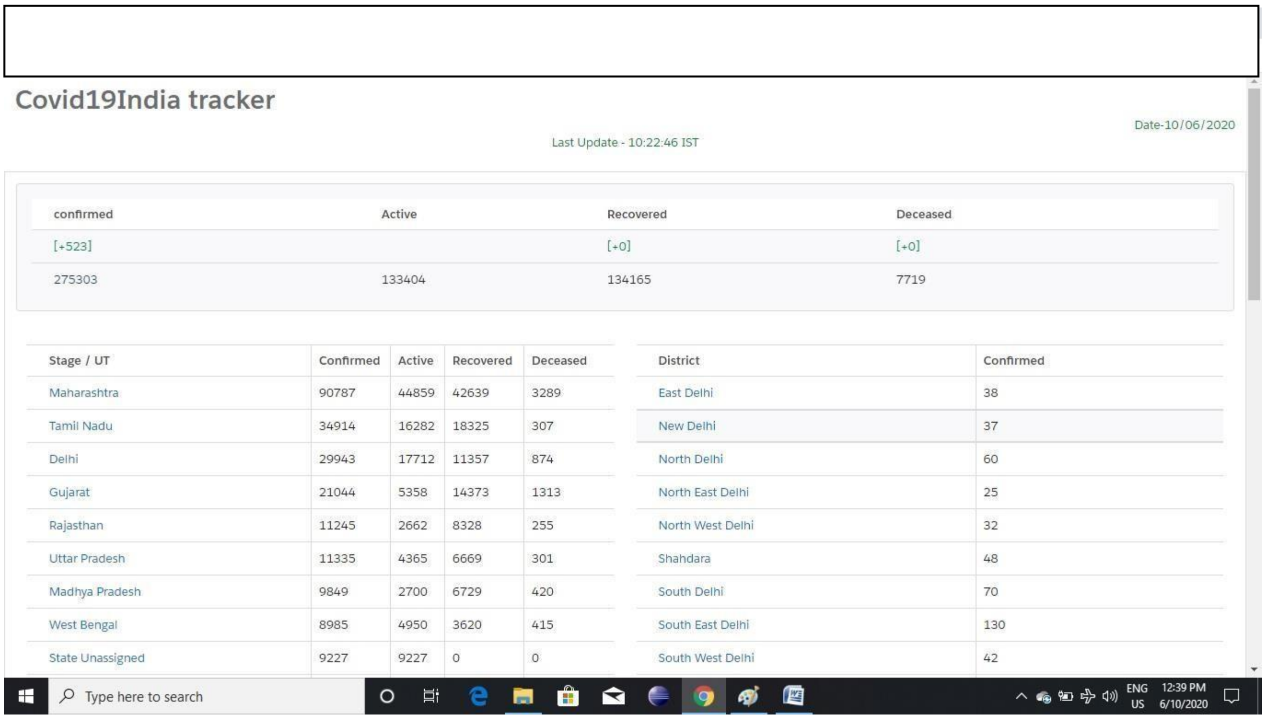
Interface Design: Interface design is the specification of the interaction between a system and its environment. this phase proceeds at a high level of abstraction with respect to the inner workings of the system that is during interface design, the internal of the systems are completely ignored and the system is treated as a black box. Attention is focused on the dialogue between the target system and the users, devices, and other systems with which it interacts. The design problem statement produced during the problem analysis step should identify the people, other systems, and devices which are collectively called agents.

Architectural Design: Architectural design is the specification of the major components of a system, the responsibilities, properties, interfaces and the relationships and interactions between them. In architectural design comma the overall structure of the system is chosen, but the internal details of major components are ignored.

Detailed Design: Detailed design is the specification of the internal elements of all major system components, their properties, relationships, processing and often their algorithms and data structures. e from the users that can undermine the possible application benefits.

4.1 SCREENSHOTS

FIG 4.1 shows the web home page of the application in which we can see the application is running on the server live and the real-time data is available this is the finished product.



Covid19India tracker

Date-10/06/2020

Last Update - 10:22:46 IST

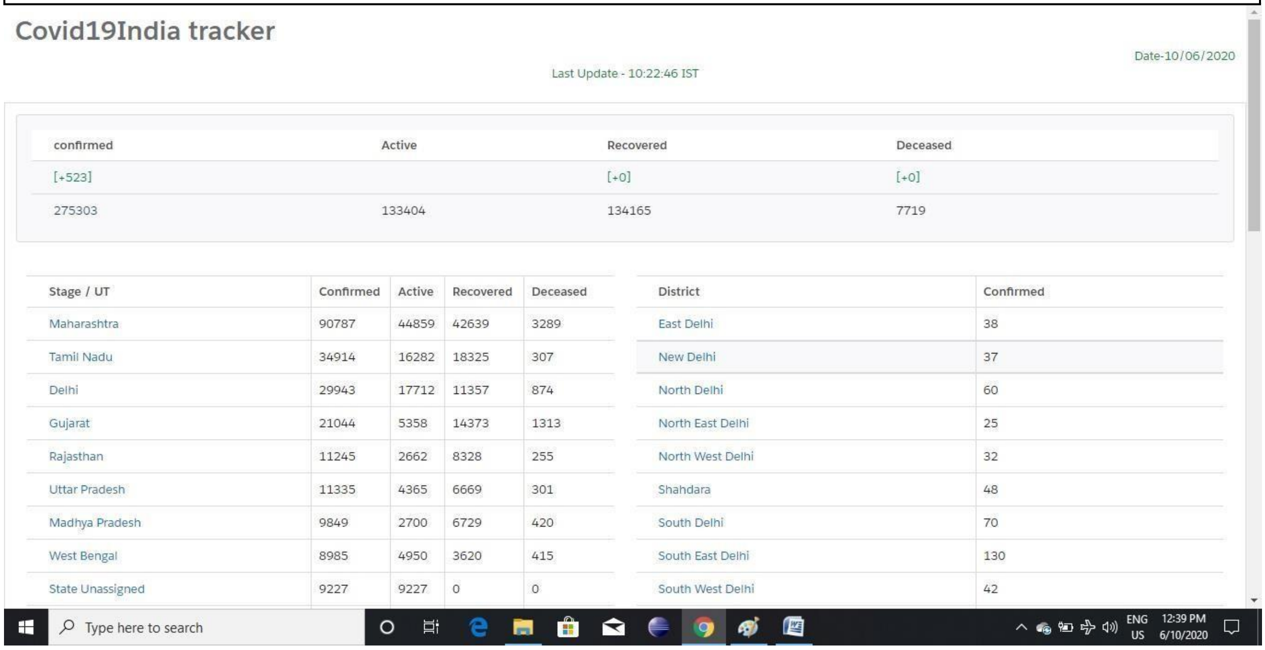
confirmed	Active	Recovered	Deceased
[+523]		[+0]	[+0]
275303	133404	134165	7719

Stage / UT	Confirmed	Active	Recovered	Deceased
Maharashtra	90787	44859	42639	3289
Tamil Nadu	34914	16282	18325	307
Delhi	29943	17712	11357	874
Gujarat	21044	5358	14373	1313
Rajasthan	11245	2662	8328	255
Uttar Pradesh	11335	4365	6669	301
Madhya Pradesh	9849	2700	6729	420
West Bengal	8985	4950	3620	415
State Unassigned	9227	9227	0	0

District	Confirmed
East Delhi	38
New Delhi	37
North Delhi	60
North East Delhi	25
North West Delhi	32
Shahdara	48
South Delhi	70
South East Delhi	130
South West Delhi	42

FIG4.1 Home Page

FIG 4.2 shows the statistics of cases in country of India with the help of header.



Covid19India tracker

Date-10/06/2020

Last Update - 10:22:46 IST

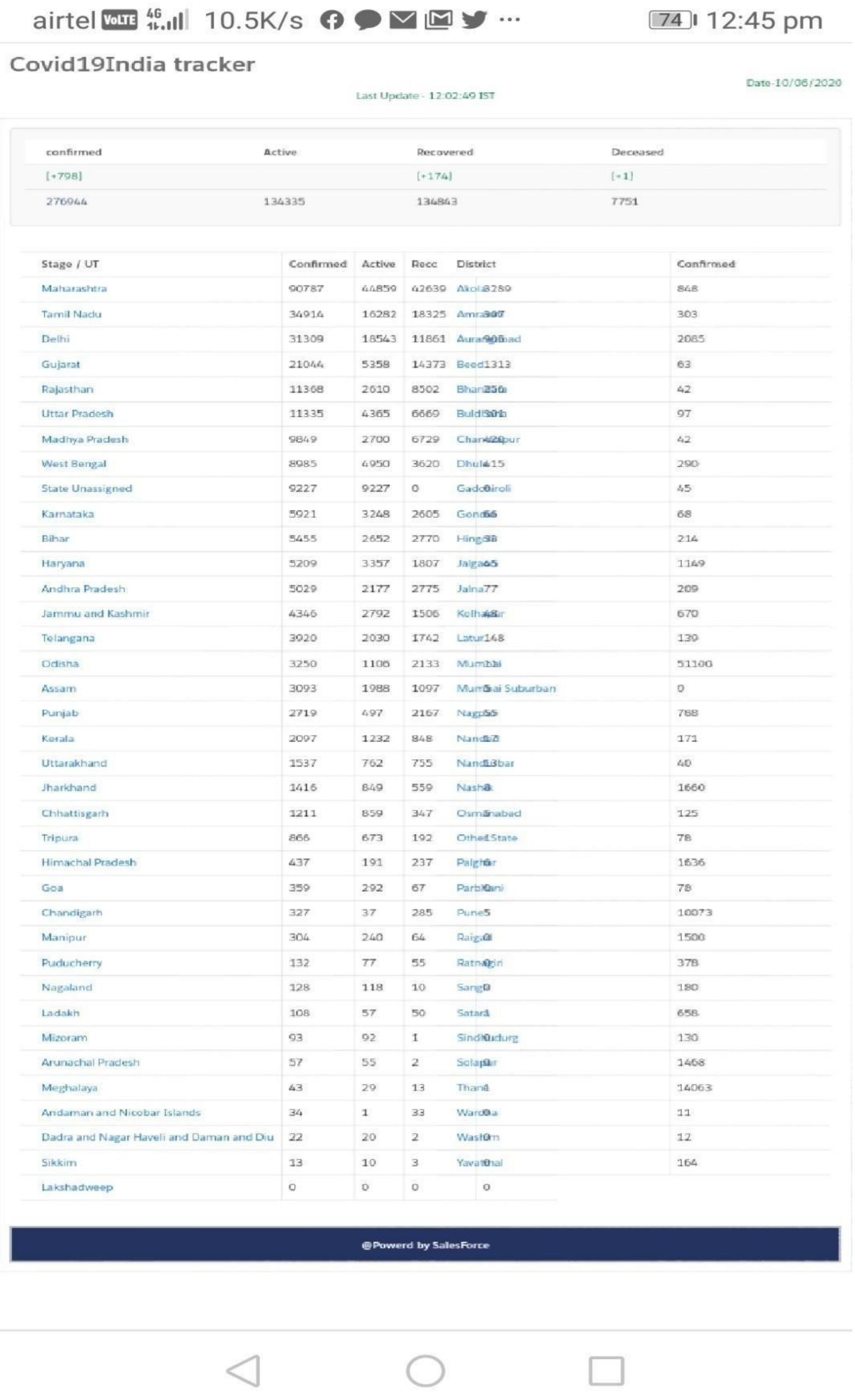
confirmed	Active	Recovered	Deceased
[+523]		[+0]	[+0]
275303	133404	134165	7719

Stage / UT	Confirmed	Active	Recovered	Deceased
Maharashtra	90787	44859	42639	3289
Tamil Nadu	34914	16282	18325	307
Delhi	29943	17712	11357	874
Gujarat	21044	5358	14373	1313
Rajasthan	11245	2662	8328	255
Uttar Pradesh	11335	4365	6669	301
Madhya Pradesh	9849	2700	6729	420
West Bengal	8985	4950	3620	415
State Unassigned	9227	9227	0	0

District	Confirmed
East Delhi	38
New Delhi	37
North Delhi	60
North East Delhi	25
North West Delhi	32
Shahdara	48
South Delhi	70
South East Delhi	130
South West Delhi	42

FIG.4.2 Country Data

FIG4.3 shows the screenshots of mobile platform as the application is fully responsive. This shows how the application look and feel changes according to the mobile screen.



airtel VoLTE 4G 10.5K/s ··· 74 12:45 pm

Covid19India tracker

Date-10/06/2020

Last Update - 12:02:49 IST

confirmed	Active	Recovered	Deceased
[+798]		[+174]	[+1]
276944	134335	134843	7751

Stage / UT	Confirmed	Active	Recc	District	Confirmed
Maharashtra	90787	44859	42639	Akol3289	848
Tamil Nadu	34914	16282	18325	Amra307	303
Delhi	31309	18543	11861	Aura905bad	2085
Gujarat	21044	5358	14373	Beed1313	63
Rajasthan	11368	2610	8502	Bhan256a	42
Uttar Pradesh	11335	4365	6669	Buld301a	97
Madhya Pradesh	9849	2700	6729	Chan420pur	42
West Bengal	8985	4950	3620	Dhul415	290
State Unassigned	9227	9227	0	Gad0iroli	45
Karnataka	5921	3248	2605	Gond66	68
Bihar	5455	2652	2770	Hing38	214
Haryana	5209	3357	1807	Jalga65	1149
Andhra Pradesh	5029	2177	2775	Jalna77	209
Jammu and Kashmir	4346	2792	1506	Kolha48r	670
Telangana	3920	2030	1742	Latur148	130
Odisha	3250	1106	2133	Mumb11	51100
Assam	3093	1988	1097	Mum5ai Suburban	0
Punjab	2719	497	2167	Nagp55	788
Kerala	2097	1232	848	Nand17	171
Uttarakhand	1537	762	755	Nand13bar	40
Jharkhand	1416	849	559	Nash8	1660
Chhattisgarh	1211	859	347	Osm5nabad	125
Tripura	866	673	192	Othe1State	78
Himachal Pradesh	437	191	237	Palgh6r	1636
Goa	359	292	67	Parb0ani	78
Chandigarh	327	37	285	Pune5	10073
Manipur	304	240	64	Raiga0	1500
Puducherry	132	77	55	Ratna0iri	378
Nagaland	128	118	10	Sang0	180
Ladakh	108	57	50	Satar1	658
Mizoram	93	92	1	Sindh0durg	130
Arunachal Pradesh	57	55	2	Solap0r	1468
Meghalaya	43	29	13	Than1	14063
Andaman and Nicobar Islands	34	1	33	Ward0a	11
Dadra and Nagar Haveli and Daman and Diu	22	20	2	Wash0m	12
Sikkim	13	10	3	Yava0hal	164
Lakshadweep	0	0	0	0	

@Powerd by SalesForce

FIG 4.3Mobile Screenshot

CHAPTER 5

CODING

The coding phase of the software lifecycle is concerned with the development of code that will implement the design. This code is written is a formal language called a programming language. Programming languages have evolved over time from sequences of ones and zeros directly interpretable by a computer, through symbolic machine code, assembly languages, and finally to higher-level languages that are more understandable

Fetching API

```
public class CovidController {

   @AuraEnabled
   public static List<satewisedata>CovidData(){
      //State wise data request
      Http http=new Http();
HttpRequest req=new HttpRequest();
req.setEndpoint('https://api.covid19india.org/data.json');
req.setMethod('GET');
req.setTimeout(60000);
HttpResponse res=http.send(req);
      String jsonStateData=res.getBody();
wrpStateWiseStateList =(wrpStateWise)JSON.deserialize(jsonStateData, wrpStateWise.class);

      Http http1=new Http();
HttpRequest req1=new HttpRequest();
      req1.setEndpoint('https://api.covid19india.org/v2/state_district_wise.json');
      req1.setMethod('GET');
      req1.setTimeout(60000);
```

```
HttpResponse res1=http1.send(req1);
        String jsonDistricData=res1.getBody();
Sates[] DisList =(Sates[])JSON.deserialize(jsonDistricData, Sates[].class);

for(satewisedataws: StateList.statewise){

for(Sates s:DisList){

if(ws.state == s.state){
ws.disData=new List<districdata>();
ws.disData.addAll(s.districtData);
                }
            }
        }



        return StateList.statewise;
    }

    //for state wise data process

    public class wrpStateWise{

        @AuraEnabled public satewisedata[]statewise {get;set;}
    }
    public class satewisedata{

        @AuraEnabled public String active {get;set;}
        @AuraEnabled public String confirmed {get;set;}
        @AuraEnabled public String deaths {get;set;}
        @AuraEnabled public String deltaconfirmed {get;set;}
        @AuraEnabled public String deltadeaths {get;set;}
        @AuraEnabled public String deltarecovered {get;set;}
        @AuraEnabled public String lastupdatedtime {get;set;}
        @AuraEnabled public String recovered {get;set;}
        @AuraEnabled public String state {get;set;}
        @AuraEnabled public String statecode {get;set;}
        @AuraEnabled public districdata[]disData {get;set;}

    }
    //for distric data process

    public class Sates{

        String state;
districdata[]districtData;
    }
    public class districdata{

        @AuraEnabled public String district;
        @AuraEnabled public Integer confirmed;
        @AuraEnabled public String lastupdatedtime;
        @AuraEnabled public deltadata delta;
    }
    public class deltadata{

        @AuraEnabled public Integer confirmed;
    }

}
```

Home Component

```
<aura:component controller="CovidController"
implements="forceCommunity:availableForAllPageTypes" access="global" >
<aura:attribute name="CovidList" type="List" access="public" />
<aura:attribute name="SelectedDisList" type="List" access="public" />
<aura:attribute name="Date" type="string" />
<aura:attribute name="Time" type="string" />

<aura:handler name="init" value="{!this}" action="{!c.doinit}" />
<div class="slds-p-around_small">
<div class="slds-text-title "><p class="slds-text-heading_large"><b>Covid19India
tracker</b></p></div>
<div class="slds-text-align_rightslds-text-color_success">Date-{!v.Date}</div>
<center class="slds-text-color_success"><p> Last Update -{!v.Time} IST      </p></center>
</div>
<div class="c-container">
<lightning:layoutmultipleRows="true">
<lightning:layoutItem padding="around-small" size="12">
<div class="header-column">
<div class="slds-table slds-table_cell-buffer slds-no-row-hover slds-table_borderedslds-
table_fixed-layout slds-box slds-theme_shade" role="grid">
<table >
<thead>
<tr class="slds-line-height_reset">
<th class="" scope="col">
<div class="slds-truncate" title="confirmed">confirmed</div>
</th>
<th class="" scope="col">
<div class="slds-truncate" title="Active">Active</div>
</th>
<th class="" scope="col">
<div class="slds-truncate" title="Recovered">Recovered</div>
```

```
</th>
<th class="" scope="col">
<div class="slds-truncate" title="Deceased">Deceased</div>
</th>
</tr>
</thead>
<tbody>
<aura:iteration start="0" end="1" items="{!v.CovidList}" var="co">
<tr class="slds-hint-parent">
<th scope="row">
<div class="slds-truncate slds-text-color_success"
title="confirmed">[+{!co.deltaconfirmed}]</div>
</th>
<td role="gridcell">
<div class="slds-truncate" title="Active"></div>
</td>
<td role="gridcell">
<div class="slds-truncate slds-text-color_success"
title="Recovered">[+{!co.deltarecovered}]</div>
</td>
<td role="gridcell">
<div class="slds-truncate slds-text-color_success"
title="Deceased">[+{!co.deltadeaths}]</div>
</td>
</tr>
<tr class="slds-hint-parent">
<th scope="row">
<div class="slds-truncate" title="Confimed">{!co.confirmed}</div>
</th>
<td role="gridcell">
<div class="slds-truncate" title="Active">{!co.active}</div>
</td>
```

```
<td role="gridcell">
<div class="slds-truncate" title="Recovered">{!co.recovered}</div>
</td>
<td role="gridcell">
<div class="slds-truncate" title="Deceased">{!co.deaths}</div>
</td>
</tr>
</aura:iteration>
</tbody>
</table>
</div>
</div>

</lightning:layoutItem>

<lightning:layoutItem padding="around-small" size="12">
<lightning:layout>

<lightning:layoutItem padding="around-small" size="6">
<div class="header-column">
<table class="slds-table slds-table_cell-buffer slds-table_borderedslds-table_col-bordered">

<thead>
<tr class="slds-line-height_reset">
<th class="" scope="col">
<div class="slds-truncate" title="Stage / UT">Stage / UT</div>
</th>
<th class="" scope="col">
<div class="slds
-truncate" title="Confirmed">Confirmed</div>
</th>
```

```
<th class="" scope="col">

<div class="slds-truncate" title="Active">Active</div>

</th>

<th class="" scope="col">

<div class="slds-truncate" title="Recovered">Recovered</div>

</th>

<th class="" scope="col">

<div class="slds-truncate" title="Deceased">Deceased</div>

</th>

</tr>

</thead>

<tbody>


<aura:iteration start="1" items="{!v.CovidList}" var="covid" indexVar="index">

<tr class="slds-hint-parent" onclick="{! c.showRow }" data-row-index="{!index}">



<th data-label="Stage / UT" scope="row">

<div class="slds-truncate" title="+{covid.state}">

<a href="javascript:void(0);" tabindex="-1">{!covid.state}</a>

</div>

</th>

<td data-label="Confirmed">

<div class="slds-truncate" title="{!covid.confirmed}">{!covid.confirmed}</div>

</td>

<td data-label="Close Date">

<div class="slds-truncate" title="{! covid.active}">{!covid.active}</div>

</td>

<td data-label="Prospecting">

<div class="slds-truncate" title="{!covid.recovered}">{!covid.recovered}</div>

</td>
```

```
<td data-label="Confidence">

<div class="slds-truncate" title="{!covid.deaths}">{!covid.deaths}</div>

</td>


</tr>

</aura:iteration>

</tbody>


</table>

</div>

</lightning:layoutItem>

<lightning:layoutItem padding="around-small" size="6">

<div class="header-column">

<table class="slds-table slds-table_cell-buffer slds-table_borderedslds-table_col-bordered">


<thead>

<tr class="slds-line-height_reset">

<th class="" scope="col">

<div class="slds-truncate" title="District">District</div>

</th>

<th class="" scope="col">

<div class="slds-truncate" title="Confirmed">Confirmed</div>

</th>


</tr>

</thead>

<tbody>


<aura:iteration start="1" items="{!v.SelectedDisList}" var="dis" >

<tr class="slds-hint-parent" >

<th data-label="District" scope="row">
```

```
<div class="slds-truncate" title="+{dis.district}">
<a href="javascript:void(0);" tabindex="-1">{!dis.district}</a>
</div>
</th>
<td data-label="Confirmed">
<div class="slds-truncate" title="Confirmed">{!dis.confirmed}</div>
</td>

</tr>
</aura:iteration>
</tbody>
</table>
</div>
</lightning:layoutItem>
</lightning:layout>
</lightning:layoutItem>

<lightning:layoutItem flexibility="auto" padding="around-small" size="12">
<div class="page-footer page-section" align="center">
<h4><b>@Powerd by SalesForce</b></h4>
</div>
</lightning:layoutItem>

</lightning:layout>
</div>



</aura:component>
```

Covid19 API

```
{
  "State Unassigned": {
    "districtData": {
    "Unassigned": {
        "notes": "",
        "active": 8959,
        "confirmed": 8959,
        "deceased": 0,
        "recovered": 0,
        "delta": {
        "confirmed": 516,
          "deceased": 0,
          "recovered": 0
        }
      }
    },
    "statecode": "UN"
  },
  "Andaman and Nicobar Islands": {
    "districtData": {
      "Nicobars": {
        "notes": "",
        "active": 0,
        "confirmed": 0,
        "deceased": 0,
        "recovered": 0,
        "delta": {
```

```
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"North and Middle Andaman": {
  "notes": "",
  "active": 0,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 1,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"South Andaman": {
  "notes": "",
  "active": 0,
  "confirmed": 32,
  "deceased": 0,
  "recovered": 32,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
    "Unknown": {
      "notes": "",
      "active": 1,
      "confirmed": 1,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "AN"
},
"Andhra Pradesh": {
  "districtData": {
  "Foreign Evacuees": {
      "notes": "",
      "active": 114,
      "confirmed": 132,
      "deceased": 0,
      "recovered": 18,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
"Anantapur": {
 "notes": "",
 "active": 128,
 "confirmed": 306,
 "deceased": 5,
 "recovered": 173,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"Chittoor":   {
 "notes":    "",
 "active": 86,
 "confirmed": 303,
 "deceased": 4,
 "recovered": 213,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"East Godavari": {
 "notes": "",
 "active": 165,
 "confirmed": 281,
 "deceased": 2,
 "recovered": 114,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"Guntur": {
 "notes": "",
 "active": 164,
 "confirmed": 566,
 "deceased": 9,
 "recovered": 393,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"Krishna": {
 "notes": "",
 "active": 193,
 "confirmed": 570,
 "deceased": 24,
 "recovered": 353,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
  }
},
"Kurnool": {
  "notes": "",
  "active": 145,
  "confirmed": 776,
  "deceased": 28,
  "recovered": 603,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Other State": {
  "notes": "",
  "active": 553,
  "confirmed": 907,
  "deceased": 0,
  "recovered": 354,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Prakasam": {
  "notes": "",
  "active": 17,
  "confirmed": 86,
  "deceased": 0,
  "recovered": 69,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"S.P.S. Nellore": {
  "notes": "",
  "active": 160,
  "confirmed": 352,
  "deceased": 4,
  "recovered": 188,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Srikakulam": {
  "notes": "",
  "active": 22,
  "confirmed": 33,
  "deceased": 0,
  "recovered": 11,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Visakhapatnam": {
  "notes": "",
  "active": 59,
  "confirmed": 143,
  "deceased": 1,
  "recovered": 83,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Vizianagaram": {
  "notes": "",
  "active": 32,
  "confirmed": 45,
  "deceased": 0,
  "recovered": 13,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"West Godavari": {
  "notes": "",
  "active": 149,
  "confirmed": 209,
  "deceased": 0,
  "recovered": 60,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Y.S.R. Kadapa": {
  "notes": "",
  "active": 59,
  "confirmed": 173,
  "deceased": 0,
  "recovered": 114,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Unknown": {
  "notes": "",
  "active": 131,
  "confirmed": 147,
  "deceased": 0,
  "recovered": 16,
      "delta": {
        "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "AP"
},
"Arunachal Pradesh": {
  "districtData": {
  "Anjaw": {
      "notes": "",
      "active": 0,
      "confirmed": 0,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
    "Changlang": {
      "notes": "",
      "active": 38,
      "confirmed": 38,
      "deceased": 0,
      "recovered": 0,
      "delta": {
        "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
    "East Kameng": {
      "notes": "",
      "active": 0,
      "confirmed": 0,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
    "East Siang": {
      "notes": "",
      "active": 2,
      "confirmed": 2,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
```

```
},
"Kamle": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"KraDaadi": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"KurungKumey": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Lepa Rada": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Lohit": {
  "notes": "",
  "active": 1,
  "confirmed": 2,
  "deceased": 0,
  "recovered": 1,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
   }
},
"Longding": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
   }
},
"Lower Dibang Valley": {
  "notes": "",
  "active": 1,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
   }
},
"Lower Siang": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Lower Subansiri": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Namsai":  {
  "notes": "",
  "active": 2,
  "confirmed": 2,
  "deceased": 0,
  "recovered":  0,
  "delta": {
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"PakkeKessang": {
  "notes": "",
  "active": 0,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 1,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Papum Pare": {
  "notes": "",
  "active": 7,
  "confirmed": 7,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Shi Yomi": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Siang": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Tawang": {
  "notes": "",
  "active": 1,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Tirap": {
  "notes": "",
  "active": 1,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Upper Dibang Valley": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Upper Siang": {
  "notes": "",
  "active": 1,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Upper Subansiri": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"West Kameng": {
  "notes": "",
  "active": 0,
      "confirmed": 0,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
    "West Siang": {
      "notes": "",
      "active": 1,
      "confirmed": 1,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "AR"
},
"Assam": {
  "districtData": {
  "Baksa": {
      "notes": "",
      "active": 17,
 "confirmed": 17,
 "deceased": 0,
 "recovered": 0,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"Barpeta":   {
 "notes":   "",
 "active": 50,
 "confirmed": 105,
 "deceased": 0,
 "recovered": 55,
 "delta": {
 "confirmed": 0,
  "deceased": 0,
  "recovered": 0
 }
},
"Biswanath": {
 "notes": "",
 "active": 45,
 "confirmed": 52,
 "deceased": 0,
 "recovered": 7,
 "delta": {
 "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bongaigaon": {
  "notes": "",
  "active": 12,
  "confirmed": 19,
  "deceased": 0,
  "recovered": 7,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Cachar": {
  "notes": "",
  "active": 8,
  "confirmed": 90,
  "deceased": 0,
  "recovered": 82,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Charaideo": {
  "notes": "",
  "active": 15,
  "confirmed": 15,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Chirang":   {
  "notes":   "",
  "active": 38,
  "confirmed": 38,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Darrang":   {
  "notes":   "",
  "active": 89,
  "confirmed": 89,
  "deceased": 0,
  "recovered": 0,
  "delta": {
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Dhemaji":  {
  "notes":   "",
  "active": 65,
  "confirmed": 71,
  "deceased": 0,
  "recovered": 6,
  "delta": {
  "confirmed": 1,
    "deceased": 0,
    "recovered": 0
  }
},
"Dhubri": {
  "notes": "",
  "active": 301,
  "confirmed": 306,
  "deceased": 0,
  "recovered": 5,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
```

```
},
"Dibrugarh": {
  "notes": "",
  "active": 50,
  "confirmed": 55,
  "deceased": 0,
  "recovered": 5,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Dima Hasao": {
  "notes": "",
  "active": 81,
  "confirmed": 88,
  "deceased": 0,
  "recovered": 7,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Goalpara": {
  "notes": "",
  "active": 38,
  "confirmed": 57,
  "deceased": 0,
  "recovered": 4,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Golaghat": {
  "notes": "",
  "active": 138,
  "confirmed": 225,
  "deceased": 0,
  "recovered": 87,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Hailakandi": {
  "notes": "",
  "active": 46,
  "confirmed": 52,
  "deceased": 1,
  "recovered": 5,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Hojai": {
  "notes": "",
  "active": 266,
  "confirmed": 266,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Jorhat": {
  "notes": "",
  "active": -8,
  "confirmed": 46,
  "deceased": 0,
  "recovered": 54,
  "delta": {
  "confirmed": 2,
    "deceased": 0,
    "recovered": 0
  }
},
"Kamrup": {
  "notes": "",
  "active": 128,
  "confirmed": 149,
  "deceased": 0,
  "recovered": 21,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Kamrup Metropolitan": {
  "notes": "",
  "active": 105,
  "confirmed": 423,
  "deceased": 3,
  "recovered": 315,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"KarbiAnglong": {
  "notes": "",
  "active": 7,
  "confirmed": 14,
  "deceased": 0,
  "recovered": 7,
  "delta": {
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Karimganj": {
  "notes": "",
  "active": 67,
  "confirmed": 83,
  "deceased": 0,
  "recovered": 16,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Kokrajhar": {
  "notes": "",
  "active": 63,
  "confirmed": 73,
  "deceased": 0,
  "recovered": 10,
  "delta": {
  "confirmed": 16,
    "deceased": 0,
    "recovered": 0
  }
},
"Lakhimpur": {
  "notes": "",
  "active": 69,
  "confirmed": 75,
  "deceased": 0,
  "recovered": 6,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"Majuli": {
  "notes": "",
  "active": 7,
  "confirmed": 7,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Morigaon": {
  "notes": "",
  "active": 30,
  "confirmed": 49,
  "deceased": 0,
  "recovered": 19,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Nagaon": {
  "notes": "",
  "active": 173,
  "confirmed": 198,
  "deceased": 0,
  "recovered": 25,
  "delta": {
  "confirmed": 19,
    "deceased": 0,
    "recovered": 0
  }
},
"Nalbari":    {
  "notes":    "",
  "active": 16,
  "confirmed": 25,
  "deceased": 0,
  "recovered": 9,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Other State": {
  "notes": "",
  "active": 0,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 1,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Sivasagar": {
  "notes": "",
  "active": 27,
  "confirmed": 27,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Sonitpur": {
  "notes":  "",
  "active": 6,
  "confirmed": 41,
  "deceased": 0,
  "recovered": 35,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"South SalmaraMankachar": {
  "notes": "",
  "active": 6,
  "confirmed": 7,
  "deceased": 0,
  "recovered": 1,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Tinsukia": {
  "notes": "",
  "active": 96,
  "confirmed": 102,
  "deceased": 0,
  "recovered": 6,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Udalguri": {
  "notes": "",
  "active": 83,
  "confirmed": 85,
  "deceased": 0,
  "recovered": 2,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"West KarbiAnglong": {
  "notes": "",
  "active": 17,
  "confirmed": 17,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Unknown": {
      "notes": "",
      "active": -175,
      "confirmed": 126,
      "deceased": 1,
      "recovered": 300,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "AS"
},
"Bihar": {
  "districtData": {
  "Araria": {
      "notes": "",
      "active": 43,
      "confirmed": 83,
      "deceased": 1,
      "recovered": 39,
      "delta": {
      "confirmed": 4,
        "deceased": 0,
        "recovered": 0
      }
    },
    "Arwal": {
```

```
  "notes": "",
  "active": 23,
  "confirmed": 63,
  "deceased": 0,
  "recovered": 40,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"Aurangabad": {
  "notes": "",
  "active": 37,
  "confirmed": 91,
  "deceased": 1,
  "recovered": 53,
  "delta": {
  "confirmed": 3,
    "deceased": 0,
    "recovered": 0
  }
},
"Banka": {
  "notes": "",
  "active": 70,
  "confirmed": 150,
  "deceased": 0,
  "recovered": 80,
  "delta": {
    "confirmed": 2,
    "deceased": 0,
    "recovered": 0
  }
},
"Begusarai": {
  "notes": "",
  "active": 95,
  "confirmed": 282,
  "deceased": 3,
  "recovered": 184,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bhagalpur": {
  "notes": "",
  "active": 138,
  "confirmed": 280,
  "deceased": 1,
  "recovered": 141,
  "delta": {
  "confirmed": 11,
    "deceased": 0,
    "recovered": 0
  }
```

```
},
"Bhojpur":   {
  "notes":    "",
  "active": 52,
  "confirmed": 128,
  "deceased": 2,
  "recovered": 74,
  "delta": {
  "confirmed": 18,
    "deceased": 0,
    "recovered": 0
  }
},
"Buxar": {
  "notes": "",
  "active": 40,
  "confirmed": 144,
  "deceased": 0,
  "recovered": 104,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Darbhanga": {
  "notes": "",
  "active": 73,
  "confirmed": 135,
  "deceased": 1,
  "recovered": 61,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"East Champaran": {
  "notes": "",
  "active": 64,
  "confirmed": 131,
  "deceased": 1,
  "recovered": 66,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Gaya": {
  "notes": "",
  "active": 88,
  "confirmed": 125,
  "deceased": 0,
  "recovered": 37,
  "delta": {
  "confirmed": 5,
    "deceased": 0,
    "recovered": 0
  }
},
"Gopalganj": {
  "notes": "",
  "active": 47,
  "confirmed": 148,
  "deceased": 0,
  "recovered": 101,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Jamui": {
  "notes": "",
  "active": 24,
  "confirmed": 48,
  "deceased": 1,
  "recovered": 23,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"Jehanabad": {
  "notes": "",
  "active": 56,
  "confirmed": 180,
  "deceased": 1,
  "recovered": 123,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Kaimur":    {
  "notes":    "",
  "active": 57,
  "confirmed": 117,
  "deceased": 0,
  "recovered": 60,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"Katihar": {
  "notes": "",
  "active": 113,
  "confirmed": 192,
  "deceased": 0,
  "recovered": 79,
  "delta": {
    "confirmed": 1,
    "deceased": 0,
    "recovered": 0
  }
},
"Khagaria": {
  "notes": "",
  "active": 166,
  "confirmed": 280,
  "deceased": 3,
  "recovered": 111,
  "delta": {
  "confirmed": 1,
    "deceased": 0,
    "recovered": 0
  }
},
"Kishanganj": {
  "notes": "",
  "active": 95,
  "confirmed": 113,
  "deceased": 0,
  "recovered": 18,
  "delta": {
  "confirmed": 9,
    "deceased": 0,
    "recovered": 0
  }
},
"Lakhisarai": {
  "notes": "",
  "active": 36,
  "confirmed": 66,
  "deceased": 0,
  "recovered": 30,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Madhepura": {
  "notes": "",
  "active": 85,
  "confirmed": 124,
  "deceased": 1,
  "recovered": 38,
  "delta": {
  "confirmed": 8,
    "deceased": 0,
    "recovered": 0
  }
},
"Madhubani": {
  "notes": "",
  "active": 154,
  "confirmed": 240,
  "deceased": 0,
 "recovered": 86,
 "delta": {
 "confirmed": 1,
   "deceased": 0,
   "recovered": 0
 }
},
"Munger":   {
 "notes":   "",
 "active": 84,
 "confirmed": 224,
 "deceased": 1,
 "recovered": 139,
 "delta": {
 "confirmed": 8,
   "deceased": 0,
   "recovered": 0
 }
},
"Muzaffarpur": {
 "notes": "",
 "active": 91,
 "confirmed": 128,
 "deceased": 1,
 "recovered": 36,
 "delta": {
 "confirmed": 2,
   "deceased": 0,
   "recovered": 0
```

```
  }
},
"Nalanda":   {
  "notes":    "",
  "active": 33,
  "confirmed": 138,
  "deceased": 1,
  "recovered": 104,
  "delta": {
  "confirmed": 3,
    "deceased": 0,
    "recovered": 0
  }
},
"Nawada":   {
  "notes":    "",
  "active": 98,
  "confirmed": 170,
  "deceased": 1,
  "recovered": 71,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"Patna": {
  "notes": "",
  "active": 96,
  "confirmed": 294,
  "deceased": 2,
  "recovered": 196,
  "delta": {
  "confirmed": 2,
    "deceased": 0,
    "recovered": 0
  }
},
"Purnia": {
  "notes": "",
  "active": 93,
  "confirmed": 149,
  "deceased": 0,
  "recovered": 56,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Rohtas": {
  "notes": "",
  "active": 70,
  "confirmed": 261,
  "deceased": 1,
  "recovered": 190,
  "delta": {
  "confirmed": 6,
    "deceased": 0,
    "recovered": 0
  }
},
"Saharsa":   {
  "notes":   "",
  "active": 50,
  "confirmed": 95,
  "deceased": 0,
  "recovered": 45,
  "delta": {
  "confirmed": 2,
    "deceased": 0,
    "recovered": 0
  }
},
"Samastipur": {
  "notes": "",
  "active": 75,
  "confirmed": 138,
  "deceased": 1,
  "recovered": 62,
  "delta": {
  "confirmed": 5,
    "deceased": 0,
    "recovered": 0
  }
},
"Saran": {
  "notes": "",
  "active": 71,
  "confirmed": 117,
  "deceased": 1,
  "recovered": 45,
  "delta": {
  "confirmed": 6,
    "deceased": 0,
    "recovered": 0
  }
},
"Sheikhpura": {
  "notes": "",
  "active": 73,
  "confirmed": 117,
  "deceased": 0,
  "recovered": 44,
  "delta": {
  "confirmed": 2,
    "deceased": 0,
    "recovered": 0
  }
},
"Sheohar":   {
  "notes":    "",
  "active": 17,
  "confirmed": 27,
  "deceased": 1,
  "recovered": 9,
  "delta": {
    "confirmed": 3,
    "deceased": 0,
    "recovered": 0
  }
},
"Sitamarhi": {
  "notes": "",
  "active": 49,
  "confirmed": 84,
  "deceased": 2,
  "recovered": 33,
  "delta": {
  "confirmed": 6,
    "deceased": 0,
    "recovered": 0
  }
},
"Siwan": {
  "notes": "",
  "active": 99,
  "confirmed": 161,
  "deceased": 3,
  "recovered": 59,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
```

```
},
"Supaul": {
  "notes": "",
  "active": 121,
  "confirmed": 168,
  "deceased": 0,
  "recovered": 47,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Vaishali":   {
  "notes":    "",
  "active": 53,
  "confirmed": 104,
  "deceased": 2,
  "recovered": 49,
  "delta": {
  "confirmed": 4,
    "deceased": 0,
    "recovered": 0
  }
},
"West Champaran": {
  "notes": "",
  "active": 51,
  "confirmed": 88,
      "deceased": 0,
      "recovered": 37,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "BR"
},
"Chandigarh": {
  "districtData": {
  "Chandigarh": {
      "notes": "",
      "active": 37,
      "confirmed": 327,
      "deceased": 5,
      "recovered": 285,
      "delta": {
      "confirmed": 1,
        "deceased": 0,
        "recovered": 0
      }
    }
  },
  "statecode": "CH"
},
"Chhattisgarh": {
```

```
"districtData": {
  "Balod": {
  "notes": "",
  "active": 20,
    "confirmed": 41,
    "deceased": 0,
    "recovered": 21,
    "delta": {
    "confirmed": 0,
      "deceased": 0,
      "recovered": 0
    }
  },
  "Baloda Bazar": {
    "notes": "",
    "active": 106,
    "confirmed": 120,
    "deceased": 0,
    "recovered": 14,
    "delta": {
    "confirmed": 0,
      "deceased": 0,
      "recovered": 0
    }
  },
  "Balrampur": {
    "notes": "",
    "active": 14,
    "confirmed": 22,
  "deceased": 0,
  "recovered": 8,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bametara": {
  "notes": "",
  "active": 11,
  "confirmed": 25,
  "deceased": 0,
  "recovered": 14,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bastar": {
  "notes": "",
  "active": 2,
  "confirmed": 3,
  "deceased": 1,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bijapur":  {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Bilaspur":   {
  "notes":   "",
  "active": 71,
  "confirmed": 105,
  "deceased": 1,
  "recovered": 33,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Dakshin Bastar Dantewada": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Dhamtari": {
  "notes": "",
  "active": 4,
  "confirmed": 6,
  "deceased": 0,
  "recovered": 2,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Durg": {
  "notes": "",
  "active": 28,
  "confirmed": 41,
  "deceased": 2,
  "recovered": 11,
  "delta": {
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Gariaband": {
  "notes": "",
  "active": 6,
  "confirmed": 10,
  "deceased": 0,
  "recovered": 4,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"JanjgirChampa": {
  "notes": "",
  "active": 53,
  "confirmed": 69,
  "deceased": 0,
  "recovered": 16,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Jashpur":   {
  "notes":   "",
  "active": 71,
  "confirmed": 79,
  "deceased": 0,
  "recovered": 8,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Kabeerdham": {
  "notes": "",
  "active": 67,
  "confirmed": 81,
  "deceased": 0,
  "recovered": 14,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Kondagaon": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Korba": {
  "notes": "",
  "active": 87,
  "confirmed": 125,
  "deceased": 0,
  "recovered": 38,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Koriya": {
  "notes": "",
  "active": 38,
  "confirmed": 45,
  "deceased": 0,
  "recovered": 7,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Mahasamund": {
  "notes": "",
  "active": 48,
  "confirmed": 57,
  "deceased": 0,
  "recovered": 9,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Mungeli":  {
  "notes":   "",
  "active": 50,
  "confirmed": 106,
  "deceased": 0,
  "recovered": 56,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Narayanpur": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Raigarh":   {
  "notes":   "",
  "active": 39,
  "confirmed": 53,
  "deceased": 0,
  "recovered": 14,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Raipur": {
  "notes": "",
  "active": 85,
  "confirmed": 97,
  "deceased": 1,
  "recovered": 11,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Rajnandgaon": {
  "notes": "",
  "active": 23,
  "confirmed": 59,
  "deceased": 0,
  "recovered": 36,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Sukma": {
  "notes": "",
  "active": 0,
  "confirmed": 0,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Surajpur": {
 "notes": "",
 "active": 5,
 "confirmed": 13,
 "deceased": 0,
 "recovered": 8,
 "delta": {
 "confirmed": 0,
   "deceased": 0,
   "recovered": 0
 }
},
"Surguja":   {
 "notes":   "",
 "active": 17,
 "confirmed": 24,
 "deceased": 0,
 "recovered": 7,
 "delta": {
 "confirmed": 0,
   "deceased": 0,
   "recovered": 0
 }
},
"Uttar BastarKanker": {
 "notes": "",
 "active": 13,
 "confirmed": 26,
 "deceased": 0,
 "recovered": 13,
  "delta": {
    "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"GaurelaPendraMarwahi": {
  "notes": "",
  "active": 0,
  "confirmed": 3,
  "deceased": 0,
  "recovered": 3,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"Unknown": {
  "notes": "",
  "active": 1,
  "confirmed": 1,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
```

```
    }
  },
  "statecode": "CT"
},
"Delhi": {
  "districtData": {
  "Central Delhi": {
      "notes": "",
      "active": 184,
      "confirmed": 184,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
    },
    "East Delhi": {
      "notes": "",
      "active": 38,
      "confirmed": 38,
      "deceased": 0,
      "recovered": 0,
      "delta": {
      "confirmed": 0,
        "deceased": 0,
        "recovered": 0
      }
```

```
},
"New Delhi": {
  "notes": "",
  "active": 37,
  "confirmed": 37,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"North Delhi": {
  "notes": "",
  "active": 60,
  "confirmed": 60,
  "deceased": 0,
  "recovered": 0,
  "delta": {
  "confirmed": 0,
    "deceased": 0,
    "recovered": 0
  }
},
"North East Delhi": {
  "notes": "",
  "active": 25,
  "confirmed": 25,
    "deceased": 0,
    "recovered": 0,
    "delta": {
    "confirmed": 0,
     "deceased": 0,
     "recovered": 0
    }
   },
   "North West Delhi": {
    "notes": "",
    "active": 31,
    "confirmed": 32,
    "deceased": 1,
    "recovered": 0,
    "delta": {
    "confirmed": 0,
     "deceased": 0,
     "recovered": 0
}
}
```

CHAPTER 6

TESTING

Testing is defined as a process of evaluation that either the specific system meets its originally specified requirements or not. The main purpose of the test plan is to discuss the testing details of the use cases of Covid-19 Live tracker. The software project test plan also describes the objective, scope and approach of the software testing effort for this project. The test plan for this project also indicates the personnel responsible for each task and also specifies the risks associated with the test plan.[4]

Unit Testing.

The Unit Testing is a test that tests each single module of the software to check for errors. This is mainly done to discover errors in the code of this Project. The main goal of the unit testing would be to isolate each part of the program and to check the correctness of the code. In the case of this project, all the web forms and the php , JavaScript code will be tested. There are many benefits for this unit testing:

- The unit testing facilitates change in the code.
- It allows testing to be done in a bottom up fashion.

At the same time, unit testing has some disadvantages such as, It might not identify each and every error in the system.

Integration Testing.

In Integration Testing, the individual software modules are combined and tested as a whole unit. The integration testing generally follows unit testing where each module is tested as a separate unit. The main purpose of the integration testing is to test the functional and performance requirements on the major items of the project. All the modules of the project developed individually would be combined together and tested as a whole system in the integration testing.

You can test React components similar to testing other JavaScript code. There are a few ways to test React components. Broadly, they divide into two categories:

1. Rendering component trees in a simplified test environment and asserting on their output.
2. Running a complete app in a realistic browser environment (also known as “end-to-end” tests).

Test runners like Jest, mocha, ava let you write test suites as regular JavaScript, and run them as part of your development process. Additionally, test suites are run as part of continuous integration.

Jest is widely compatible with React projects, supporting features like mocked modules and timers, and jsdom support. If you use Create React App, Jest is already included out of the box with useful defaults.

Libraries like mocha work well in real browser environments, and could help for tests that explicitly need it.

End-to-end tests are used for testing longer flows across multiple pages, and require a different setup.

Test Cases

Test Case 1 – Opening Application

- **Fail Criteria**: Unable to open application due to some exception.
- **Pass Criteria**: Successfully opening of application and Landing Page isshown.

Test Case 2 – API Connection

- **Fail Criteria**: Incorrect connection string.

- **Pass Criteria**: Successfully connected.

Test Case 3 – Selecting State

- **Fail Criteria**: Unable to select the country due to server error.
- **Pass Criteria**: Country Data is displayed successfully

Test Case 4 – Routing

- **Fail Criteria**: Unable to navigate to different pages
- **Pass Criteria**: Successfully navigate to different pages

REFERENCES

The following references have been used by me, during all the phases of the project:

1. R. Buyya, C. Vecchiola and S. T. Selvi, *Mastering Cloud Computing: Foundations and Applications Programming*, 2013.
2. Stefan Kempter, *Access Management*, pp. 1-4, 2018.
3. *A White paper on "On-Premise CRM to Salesforce Migration — Benefits Challenges and Best Practices"*.
4. Veronica S. Moertini, Niko Ibrahim and Verliyantina, e-CRM Development Method for e-Commerce System Owned by Small Medium Enterprises, IEEE.
5. An approach for improving the concept of Cyclomatic Complexity for Object-Oriented Programming.
6. S. M. Fernandes and C. Coutinho, *Key Performance Indicators for improving a CRM Implementation*, pp. 1401-1406, 2017.
7. Complex Systems and Computational Neuroscience Group, February 2018.
8. Sable Research Group David Eng. Java intermediate language (jil) 1.0, 2002
9. E Balaguru Swamy, "Programming with JAVA A Primer" in , Tata McGraw Hill Publications, 2010.
10. M.E. Smith, L.K. McEvoy, A. Gevins, Neurophysiological indices of strategy development, and skill acquisition. Cognit. Brain Res. Vol. 7, pp. 389-404, 1999.
11. SalesForce.com. Soql injection, 2012.
12. The Common Object Request Broker: Architecture and Specification, MA, Framingham:Object Management Group, Inc, 1995.
13. R. M. Adler, "Emerging standards for component software", *Computer*, vol. 28, no. 3, pp. 68-77, March 1995.
14. Yeon-Seok Kim and Kyong-Ho Lee, "Extracting logical structures from HTML tables", *Computer Standards and Interfaces (Elsevier)*, vol. 30, no. 5, pp. 296-308, August 2007.
15. A. F. R. Rahman, H. Alam and R. Hartono, "Content Extraction from HTML Documents", *1st International Workshop on Web Document Analysis (WDA2001)*, 2001..