

PROJECT REPORT
ON
DIGITAL TRACKING OF CHILDREN OF MIGRANT POPULATION TO
FOLLOW UP ON THEIR HEALTH

A Dissertation submitted in partial fulfilment of the
Requirements for the award of the degree of
BACHELOR OF TECHNOLOGY

in
COMPUTER SCIENCE AND ENGINEERING

Submitted By

Aditya Kuna – 17B81A0509

Arun Kusuma – 17B81A0530

Mohammed Faizuddin – 17B81A0560

Under the esteemed guidance of

Ms.A Venu Madhavi

Associate Professor-CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CVR COLLEGE OF ENGINEERING

(UGC Autonomous Institution)

Accredited by NBA & NAAC 'A' Grade

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University, Hyderabad) Vastunagar,
Mangalpally (v), Ibrahimpatanam (M), R.R.Dist-501 510.Telangana



Cherabuddi Education Society's
CVR COLLEGE OF ENGINEER

(An Autonomous Institution)

ACCREDITED BY NBA, NAAC 'A' Grade

(Approved by AICTE & Government of Telangana and Affiliated to JNTU Hyderabad) Vastunagar, Mangalpalli
(V), Ibrahimpatnam (M), R.R.District. Web: <http://www.cvr.ac.in>, email: info@cvr.ac.in Ph : 08414 – 252222,
252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled “DIGITAL TRACKING OF CHILDREN OF MIGRANT POPULATION TO FOLLOW UP ON THEIR HEALTH” being submitted by Aditya Kuna(17B81A0560), Arun Kusuma(17B81A0530) and Mohammed Faizuddin(17B81A0560) in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering to Jawaharlal Nehru Technological University (JNTUH), Hyderabad, is a bonafide work carried out by them under my guidance and supervision. The results provided in this report have not been submitted to any other university or institution for the award of any degree.

Signature of the project guide

Ms.A Venu Madhavi

Associate Professor

Dept of CSE

CVR College of Engineering

Signature of the HOD

Dr. K Venkateshwar Rao

HOD – CSE

CVR College of Engineering

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is great pleasure to convey our profound sense of gratitude to our **principal Dr. Nayanathara K. S, Vice Principal Prof. L.C. Siva Reddy, Dr. K. Venkateshwar Rao, Head of the CSE Department(HOD)**, CVR college of Engineering for having been kind enough for arranging the necessary facilities for executing the project in the college.

We would like to express our sincere gratitude to our guide, **Ms A Venu Madhavi Ma'am, Associate professor ,CSE Dept., CVR College of Engineering**, whose guidance and valuable suggestions have been indispensable to bring about the successful completion of our project.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

ABSTRACT

DIGITAL TRACKING OF CHILDREN OF MIGRANT POPULATION TO FOLLOW UP ON THEIR HEALTH is a project that is developed to support and help the immigrants using computer science knowledge. We all know how important it is to get vaccinated. Vaccination protects us from serious illness and complications of vaccine-preventable diseases which can include amputation of an arm or leg, paralysis of limbs, hearing loss, convulsions, brain damage, and death. Immunizations are one of the success stories of modern medicine. India is one of those tropical countries where many diseases occur due to lack of vaccinations. Most people who come from overseas to India might not know which vaccinations are needed and how many they already took. Our aim is to solve this problem by developing a web app which shows vaccines needed and keeps track of the user vaccinations. The user can see the vaccinations he took and gets a reminder when its time to take a vaccination. Vaccinations can save million lives if taken.

Table of Contents

TABLE OF FIGURES.....	vi
1.INTRODUCTION	1
1.0.1 What is a vaccine?	1
1.0.2 Why are vaccines important?	1
1.1 MOTIVATION.....	2
1.2 PROBLEM STATEMENT	2
1.3. PROJECT REPORT ORGANIZATIONS:	2
2.PROPOSED SOLUTION.....	3
2.1 The Characteristics of the Problem	3
2.2 DESIGN CHALLENGES:	3
2.3 PROPOSED SOLUTION:	4
3.REQUIREMENTS AND SPECIFICATIONS.....	5
3.1 SOFTWARE REQUIREMENTS:	5
3.1.1 Functional Requirements:	5
3.1.2 Non-Functional Requirements:.....	5
3.2 SYSTEM SPECIFICATIONS:.....	6
3.2.1 SOFTWARE SPECIFICATIONS:	6
3.2.2 HARDWARE SPECIFICATIONS:.....	6
4 ANALYSIS AND DESIGN	7
4.1 USE CASE DIAGRAM:	7
4.2 CLASS DIAGRAM:.....	8
4.4 SEQUENCE DIAGRAM:	10
4.5 SYSTEM ARCHITECTURE:.....	11
4.6 TECHNOLOGY DESCRIPTION:	12
4.6.1 FIREBASE	12
4.6.2 VISUAL STUDIO CODE	14
5. IMPLEMENTATION AND TESTING	16
5.1 Implementation.....	16
5.1.1 Registration of User:	16
5.1.2 Login:	21
5.1.3 Validation of Username:	24
5.1.4 Validation of Password:	25
5.1.5 Firebase Reference:.....	25

5.2 TESTING.....	29
5.2.1 Front page:.....	29
5.2.2 Login page:.....	30
5.2.3 Profile page(Recommendations):.....	31
5.2.3 Profile page(history):.....	31
6. CONCLUSION & FUTURE SCOPE	32
6.1 Conclusion:	32
6.2 Future Scope:.....	32
REFERENCES.....	32

TABLE OF FIGURES

Figure 4.0.a use case diagram	7
Figure 4.2.a class diagram	8
Figure 4.3.0.a Activity diagram.....	9
Figure 4.4.0.a Sequence diagram	10
Figure 4.5.0.a Architecture Diagram.....	11
Figure 5.1.1(a) register.....	16
Figure 5.1.1(b) register	17
Figure 5.1.1(c) register	18
Figure 5.1.1(d) register	19
Figure 5.1.1(e) register.....	20
Figure 5.1.2(a) Login	21
Figure 5.1.2(b) Login	22
Figure 5.1.2(c) Login.....	23
Figure 5.1.2(d) Login	24
Figure 5.1.5(a) firebase reference	26
Figure 5.1.5(b) firebase reference	27
Figure 5.1.5(c) firebase reference	28
Figure 5.2.1(a) register UI	29
Figure 5.2.2(a) Login UI.....	30
Figure 5.2.3(a) Home UI.....	31
Figure 5.2.3(a) Home UI.....	31

1.INTRODUCTION

Vaccines are one of our most effective health interventions, but are often misunderstood.

1.0.1 What is a vaccine?

We're protected from infectious disease by our immune system, which destroys disease-causing germs – also known as pathogens – when they invade the body. If our immune system isn't quick or strong enough to prevent pathogens taking hold, then we get ill. We use vaccines to stop this from happening. A vaccine provides a controlled exposure to a pathogen, training and strengthening the immune system so it can fight that disease quickly and effectively in future. By imitating an infection, the vaccine protects us against the real thing.

1.0.2 Why are vaccines important?

- They protect us from dangerous diseases. In some regions or populations, dangerous diseases are constantly present (endemic). Examples include hepatitis B, cholera and polio. So long as these diseases are around, we need vaccines to bolster our immune systems and protect us from harm.
- They protect children and the elderly. Our immune systems are strongest in adulthood, meaning that young children and the elderly are particularly susceptible to dangerous infections. By strengthening our immune systems early and late on in life, vaccines bypass this risk.
- They can help us control epidemics. In a world of denser cities, increased international travel, migration and ecological change, the ability of emerging infectious diseases (such as Ebola) to spread and cause devastation is increasing. Vaccines can be a key tool in managing this threat – but only if we have them ready for diseases when they appear.
- They are our most effective health intervention. Vaccines prevent an estimated 2–3 million deaths worldwide every year. But, a further 1.5 million lives could be saved annually with better global vaccine coverage. —

1.1 MOTIVATION

As we understand the threat of not being vaccinated finding a way to keep track of vaccinations, giving information about what vaccines to take for immigrants, and reminding them when to take it is an important contribution in the field of healthcare. We took this opportunity to contribute our work for healthcare.

Vaccines prevent an estimated 2–3 million deaths worldwide every year. But, a further 1.5 million lives could be saved annually with better global vaccine coverage.

When Immigrants have information about what vaccines to be taken and what vaccines to be given to their child when visiting India, they can protect themselves and their loved ones from potentially life-threatening disease-causing viruses.

This could also ease the stress on doctors and hospitals.

1.2 PROBLEM STATEMENT

Title: Digital tracking of children of migrant population to follow-up on their health status and immunization services received/to be given.

Description: Harnessing the power of digital technologies is essential for achieving the Sustainable Development Goals, including universal health coverage. To reduce the inequities in health in population with the migration either because of occupation, shelter or some other reasons, migration among cities, states, rural to urban, it is essential to track their health record so that health services can reach up to such people in the form of continuum of care. immunization of children is an important field where we can provide full immunization coverage to such community who are often left behind.

1.3. PROJECT REPORT ORGANIZATIONS:

This book contains six chapters. The first chapter contains motivation and problem statement of the project. The second chapter includes the characteristics of the problem, design challenges and proposed solution. The third chapter includes requirements and specifications. The fourth chapter includes which contains UML diagrams and Technology Description . The fifth chapter includes Implementation which contains the technologies used for developing the application and code snippets. The fifth chapter also contains testcases and screenshots of the applications. The sixth chapter looks into the future enhancements and conclusion of the project.

2.PROPOSED SOLUTION

2.1 The Characteristics of the Problem

Lack of information about vaccinations leads to death of around 1.2 million deaths globally. People who migrate to India(especially children) are vulnerable to various types of life-threatening diseases.

2.2 DESIGN CHALLENGES:

Since immunization programs were launched in low- and middle-income countries worldwide in the mid-1970s, most countries have been using the same standard package of six vaccines—measles, tetanus, diphtheria, whooping cough (pertussis), tuberculosis, and polio—in their national immunization schedule. In the last decade, however, as the public health impact of vaccines has become increasingly clear, interest in and funding for new vaccine development and delivery has surged. Throughout the next decade, low- and middle-income countries will have opportunities to introduce many new lifesaving vaccines into their standard immunization programs—in some cases doubling the number of vaccines offered in their programs within ten years. In spite of the success of immunization programs, one in five children still does not have access to basic vaccines.

The success of currently available vaccines is based on their ability to induce antibodies that block or neutralize infectious agents or their products. Antibody-mediated protection is not sufficient in all infectious diseases. The major contagious killers of today for which we do not yet have vaccines are restrained to some extent not only by antibodies but also by cell-mediated immunity. There is currently a strong clinical development pipeline for novel vaccines against infectious diseases. These candidates have entered clinical trials based on evidence for protective efficacy in animal models, or, in the case of malaria and some other infections, in a human challenge model. However, their development has proceeded without knowledge of the correlates of vaccine-induced protection and biomarkers or a

biological signature (a custom-made compilation of different biomarkers) that can predict reduced disease incidence. Ideally, clinical trials generate sufficient data to allow definition of a biosignature that can be used to predict vaccine efficacy and safety of novel candidates

2.3 PROPOSED SOLUTION:

The application will be mainly used by Indians and migrants to keep track of their vaccination status and get information about what vaccines they need to take. The app features a reminder which reminds the person if the number of days to take a vaccination becomes zero. We ask the user his personal information(name, age, email, mobile number, gender) when he is signing up for the service. When the user logs in his information is displayed along with what vaccines to take and his history of vaccines. When the user has successfully gets vaccinated with the vaccine mentioned in his recommendations he can click the 'add' button which adds the vaccine he took to the his history and removes the vaccine from his recommendation tab. This helps the user to keep a track of what vaccines he took and what vaccines he has to take and how many days are left for a particular vaccine. When a user clicks on the vaccine under the recommendation tab, A new window opens up informing him about the details of the vaccine, why is it taken?, it's side effects and other various things. This helps the user in knowing vital information about the vaccine so he can make the right decision regarding the vaccination. All the user data along with his history is stored in a realtime database known as firebase. Firebase is provided by Google and is one of the best and secure noSql database out there. So the user does not need to worry about his sensitive data being open to public.

3.REQUIREMENTS AND SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS:

Operating System: Windows 7 or above / Mac-OS or above / Ubuntu

Front end: HTML, CSS, Bootstrap, JQuery

Back-end: JavaScript

Database- Firebase

3.1.1 Functional Requirements:

These requirements define the capabilities and functions that the implemented system must have in order to achieve its intended purpose. It includes a set of inputs, behaviour and outputs in line with the objectives of the study. They include:

1. Login and Logout: To gain access to the web application users must first signup/create account with their respective details and later login using username and password. Users need to logout in order to exit the system.
2. Nationality: The user must provide his nationality so that the application can recommend him vaccines

3.1.2 Non-Functional Requirements:

These requirements that specify the criteria used to judge the operation of the system. They were constructed in agreement with functional requirements that define specific behaviour and functions. They include:

1. Usability: the system interface should be easy to use.
2. Reliability and availability: the system should be reliable and always available to perform tasks requested by the user.
3. Scalability: the system should be able to adopt additional functionalities. Additional data should be easy to incorporate.
4. Integrity: the system being data oriented, it should ensure that the data analysed and stored is not altered or corrupted.

3.2 SYSTEM SPECIFICATIONS:

Windows 7 or above/ Mac-OS or above/ Ubuntu

3.2.1 SOFTWARE SPECIFICATIONS:

Visual Studio Code

Browser(preferably chrome)

Firebase

Sublime Text Editor

3.2.2 HARDWARE SPECIFICATIONS:

A PC with a good internet connection

2gb or above RAM

4 ANALYSIS AND DESIGN

4.1 USE CASE DIAGRAM:

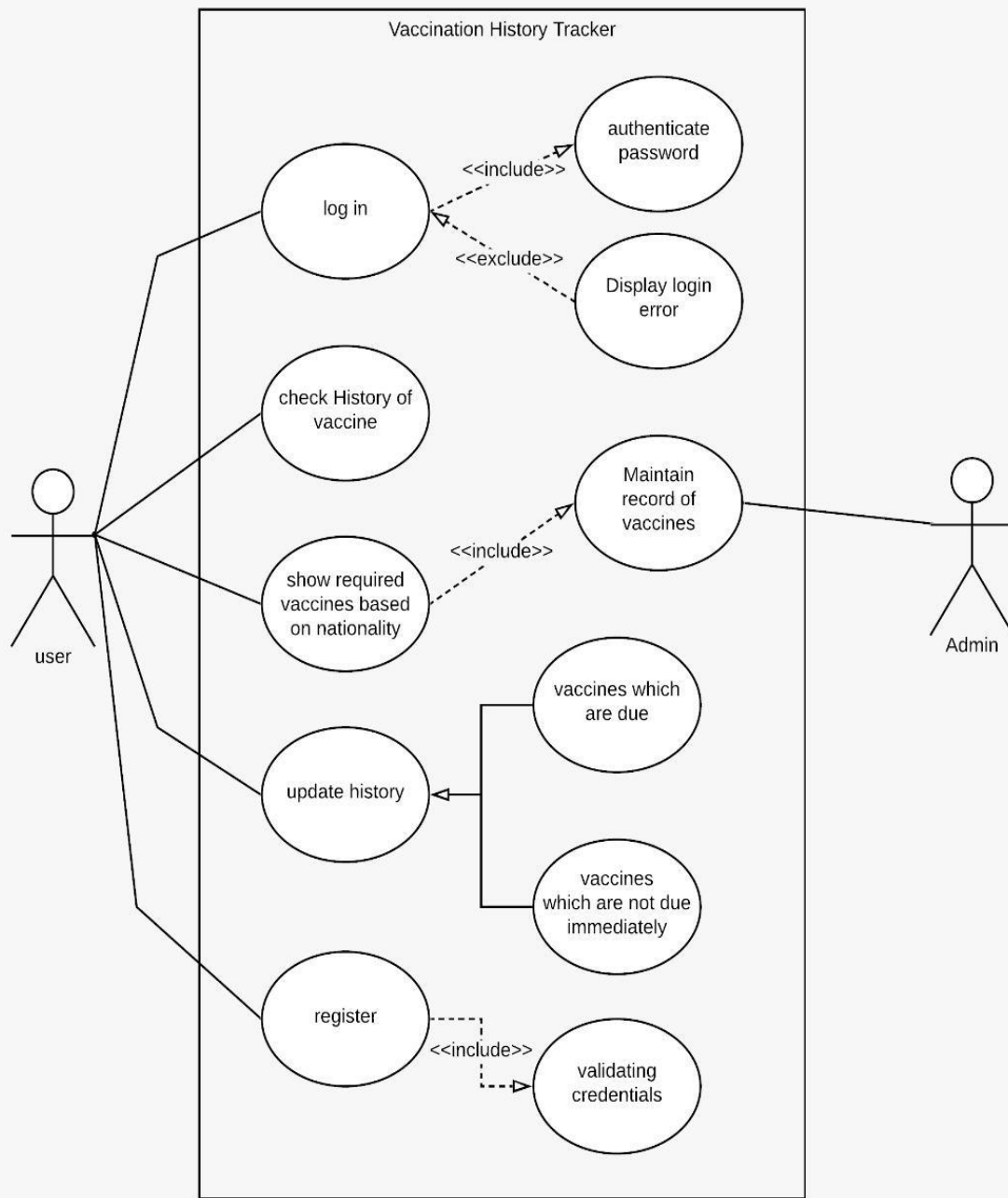


Figure 4.0.a use case diagram

4.2 CLASS DIAGRAM:

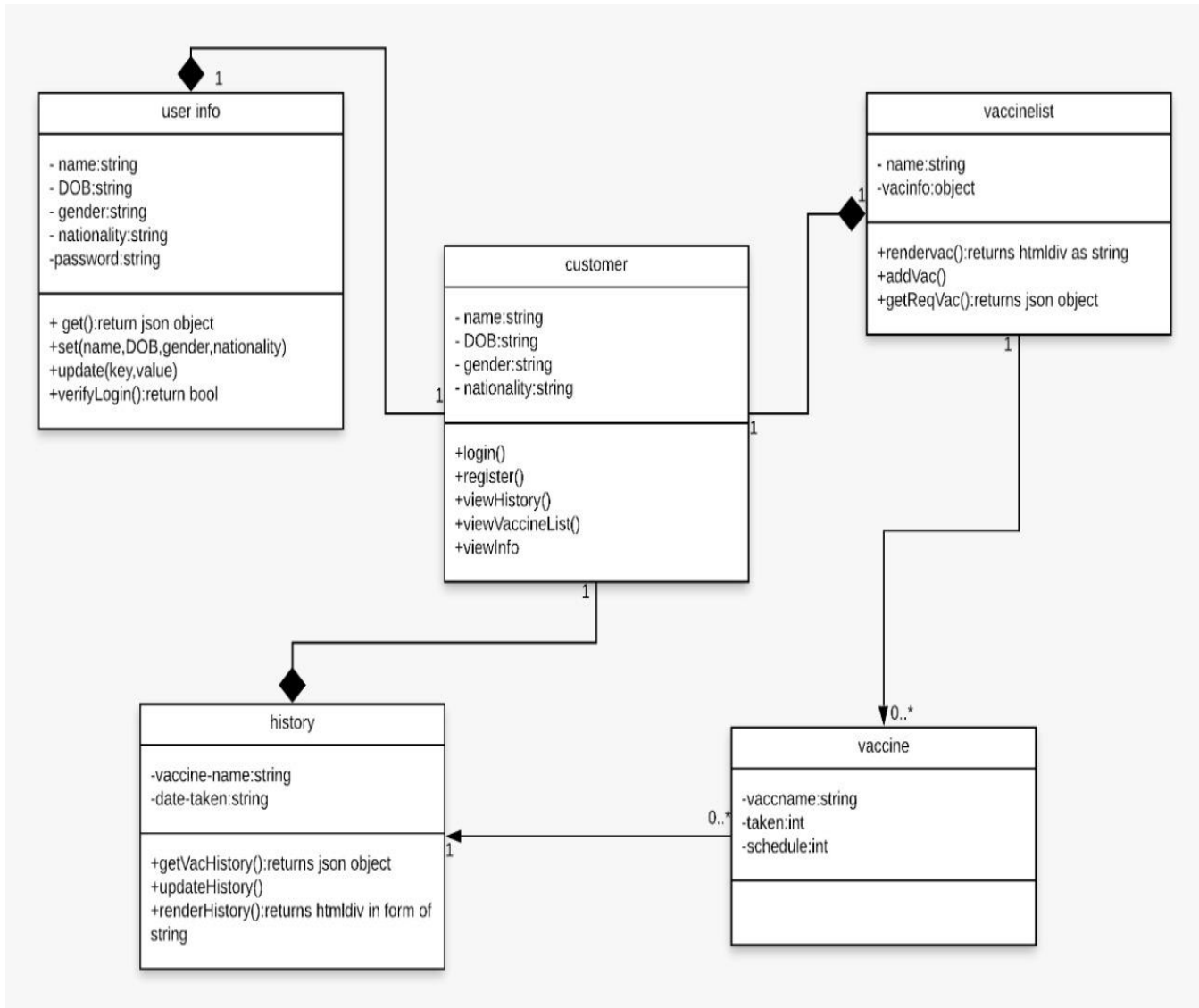


Figure 4.2.a class diagram

4.3 ACTIVITY DIAGRAM:

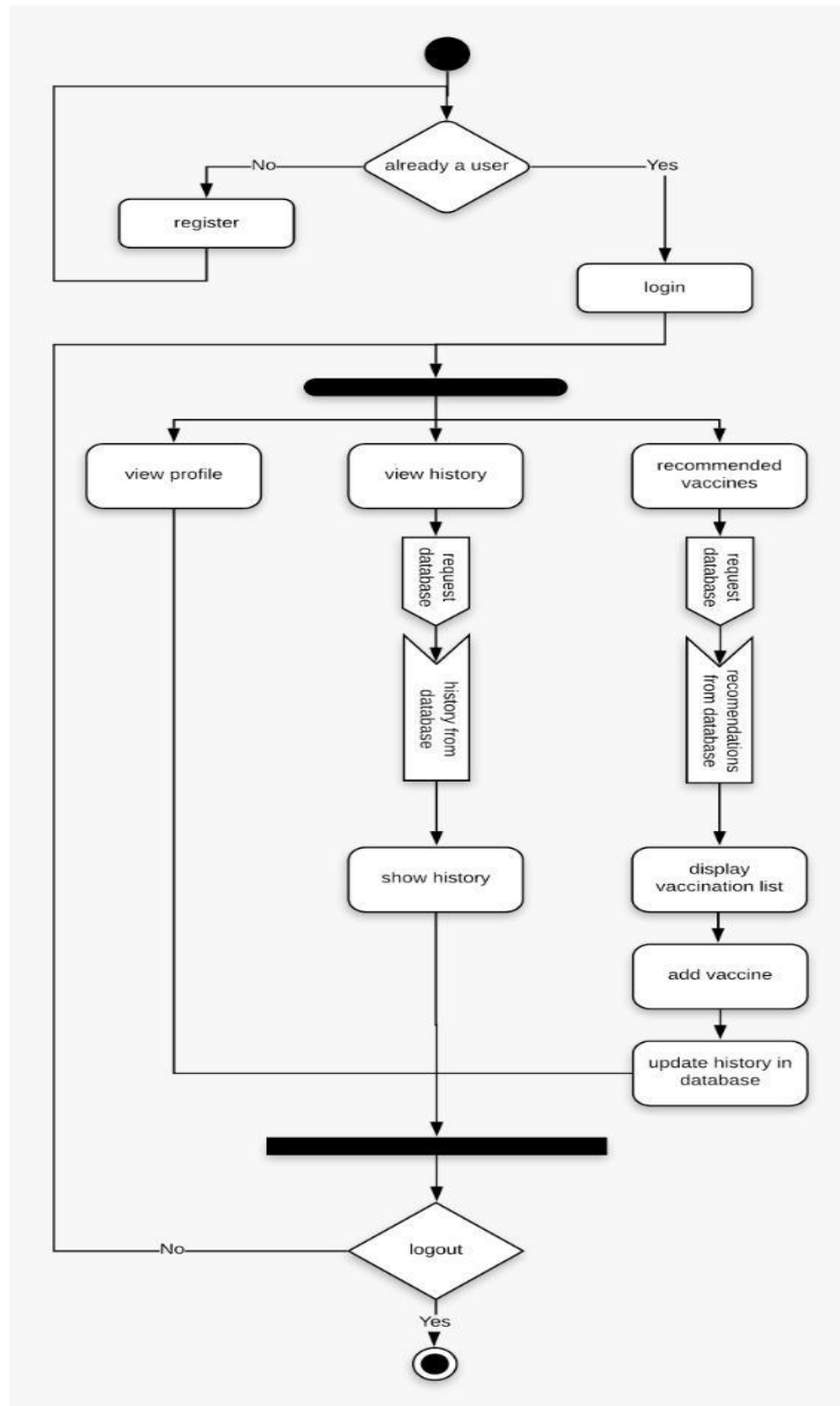


Figure 4.3.0.a Activity diagram

4.4 SEQUENCE DIAGRAM:

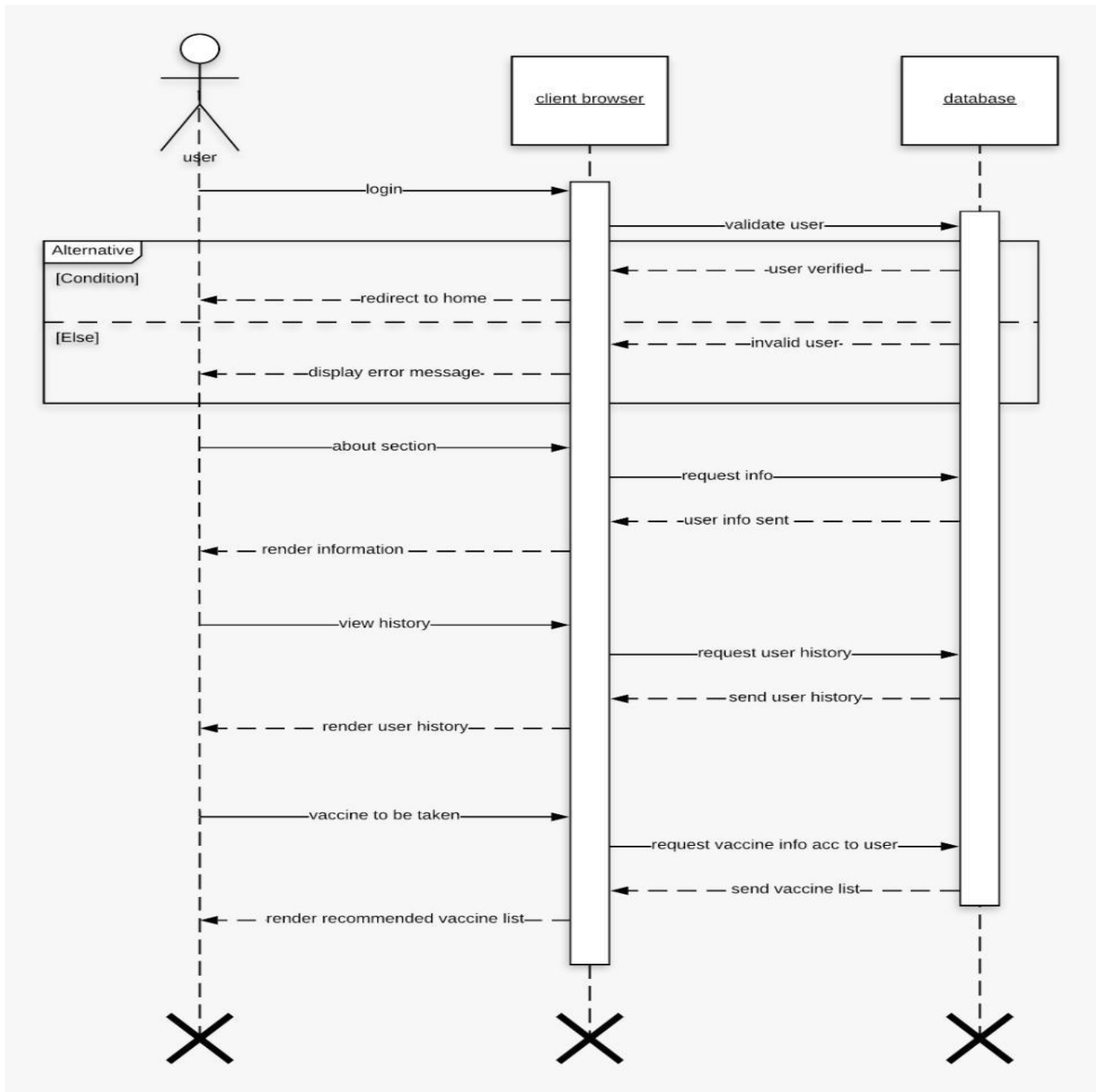


Figure 4.4.0.a Sequence diagram

4.5 SYSTEM ARCHITECTURE:

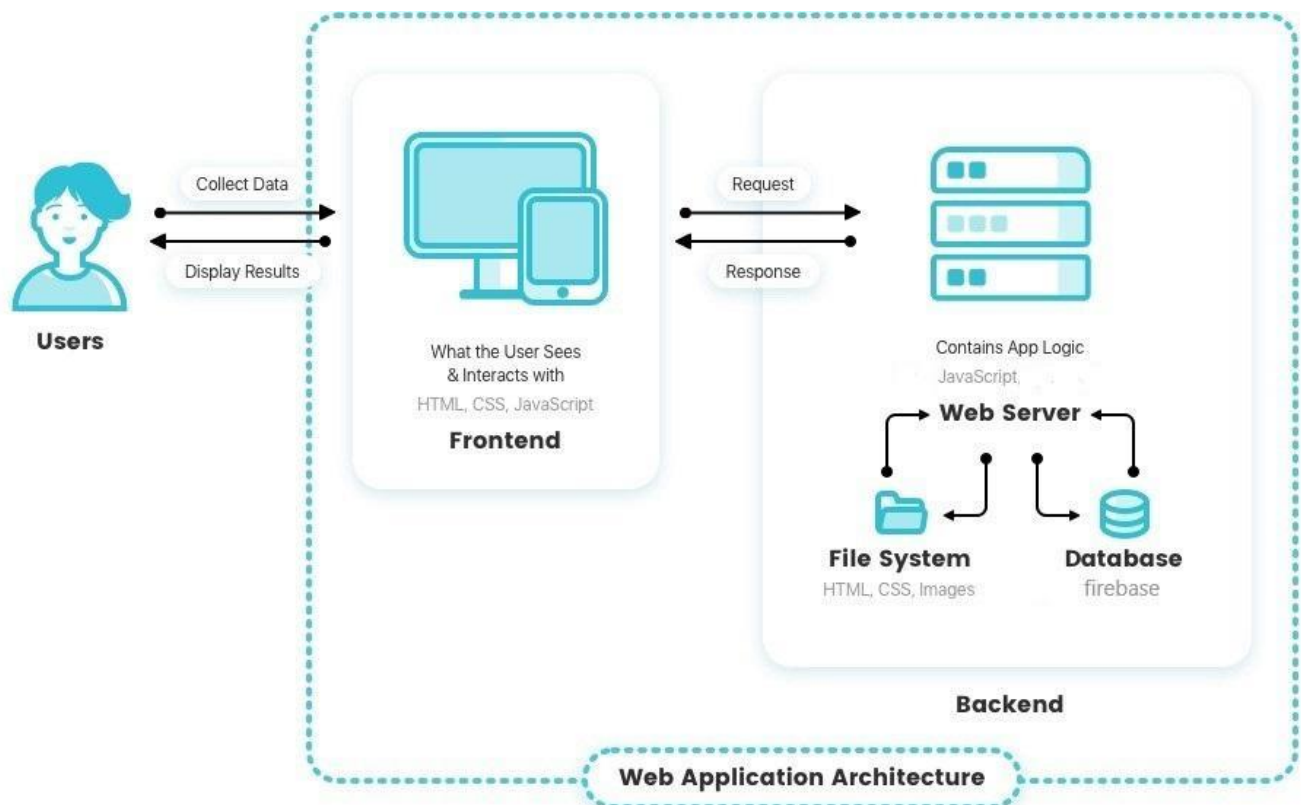


Figure 4.5.0.a Architecture Diagram

4.6 TECHNOLOGY DESCRIPTION:

4.6.1 FIREBASE

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiment.

Firebase offers a number of services, including:

- Analytics – Google Analytics for Firebase offers free, unlimited reporting on as many as 500 separate events. Analytics presents data about user behavior in iOS and Android apps, enabling better decision-making about improving performance and app marketing.
- Authentication – Firebase Authentication makes it easy for developers to build secure authentication systems and enhances the sign-in and onboarding experience for users. This feature offers a complete identity solution, supporting email and password accounts, phone auth, as well as Google, Facebook, GitHub, Twitter login and more.
- Cloud messaging – Firebase Cloud Messaging (FCM) is a cross-platform messaging tool that lets companies reliably receive and deliver messages on iOS, Android and the web at no cost.
- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time. The data is synced across all clients in real time and is still available when an app goes offline.

- Crashlytics – Firebase Crashlytics is a real-time crash reporter that helps developers track, prioritize and fix stability issues that reduce the quality of their apps. With crashlytics, developers spend less time organizing and troubleshooting crashes and more time building features for their apps.

- Performance – Firebase Performance Monitoring service gives developers insight into the performance characteristics of their iOS and Android apps to help them determine where and when the performance of their apps can be improved.

- Test lab – Firebase Test Lab is a cloud-based app-testing infrastructure. With one operation, developers can test their iOS or Android apps across a variety of devices and device configurations. They can see the results, including videos, screenshots and logs, in the Firebase console.

Firebase use cases include:

- Create onboarding flows – developers can give users a quick, intuitive sign-in process using Firebase Authentication. They allow users to sign into their apps via their Google, Twitter, Facebook or GitHub accounts in less than five minutes. Developers can also track each step of their onboarding flows to enhance the user experience. Additionally, developers can use Google Analytics for Firebase to log events at each step of their onboarding flows, create funnels to determine where users are dropping off and use remote configuration to make changes to their apps to see how those changes affect conversions.

- Customize a “welcome back” screen – developers can use personalization to give every user the best experience by customizing the initial screen based on a user’s preferences, usage history, location or language. Developers can define audiences based, in part, on user behaviors and show targeted content

to each audience.

- Progressively roll out new features – developers can launch new features with minimal risk by first testing those features on a few users to see how they work and how users respond. Then, when developers are satisfied, they can roll out their apps to the rest of their users.

4.6.2 VISUAL STUDIO CODE

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. It is based on the Electron framework, which is used to develop Node.js web apps that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

Language support

Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Data collection

Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. In addition, because of the open-source nature of the app, the telemetry code is accessible to the public, who can see exactly what is collected. According to Microsoft, the data is shared with Microsoft-controlled affiliates and subsidiaries, although the law enforcement may request it as part of a legal process.

5. IMPLEMENTATION AND TESTING

5.1 Implementation

The scenarios are evaluated as follows:

5.1.1 Registration of User:

The person is provided with a registration page where he has to supply his email, password, gender, birthdate, username

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content=
"width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content=
"ie=edge">
  <title>Register Page</title>
  <link rel="stylesheet" href=
"https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/c
ss/bootstrap.min.css"
  integrity=
"sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFe
WPGFN9MuhOf23Q9Ifjh"
  crossorigin="anonymous">
  <script src=
"https://code.jquery.com/jquery-3.4.1.slim.min.js"
  integrity=
"sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imG
FAV0wwj1yYfoRSJoZ+n"
  crossorigin="anonymous"></script>
  <script src=
"https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/u
md/popper.min.js"
  integrity=
"sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTt
mI3UksdQRVvoxMfooAo"
```

Figure 5.1.1(a) register

```

crossorigin="anonymous"></script>
  <script src=
"https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/j
s/bootstrap.min.js"
  integrity=
"sha384-wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj
0Uod8GCExl30g8ifwB6"
  crossorigin="anonymous"></script>
  <script src="register.js"></script>
  <link rel="stylesheet" href="styles1.css">
</head>
<body">

  <div class="login-box">

    <h1>REGISTER</h1>

    <div class="textbox">
      <input type="text" id="first"
placeholder="Full Name">
    </div>
    <div class="textbox">
      <input type="email" id="mail"
placeholder="Email Id">
    </div>
    <div class=" textbox">
      <input type="date" style="
color: grey;" id="dob" placeholder="dd/mm/yyyy">

```

Figure 5.1.1(b) register


```

        </div>
        <div class=" textbox">
            <input type="text" id="number"
placeholder="Mobile Number">
        </div>
        <div class=" textbox">
            <input type="text" id="uname"
placeholder="Username" aria-describedby=
"basic-addon1">
        </div>
        <div class=" textbox">
            <input type="password" id=
"pass" placeholder="password">
        </div>
        <div class="textbox">
            <select id="nation" >
                <option >usa</option>
                <option >africa</option>
                <option >india</option>
                <option >canada</option>
                <option >dubai</option>
                <option selected>
--select nation--</option>
            </select>
        </div>
        <div >
            <input type="submit" onclick="
register()" class="button btn-success" value="Submit"
>

```

Figure 5.1.1(c) register

```

>
    </div>
    <p align="center">
Already have account?<a href="login.html" style="
color: green">Login Here!</a></p>

    </div>
    <script src=
"https://www.gstatic.com/firebasejs/7.15.4/firebase.j
s"
></script>

<!-- TODO: Add SDKs for Firebase products that you wa
nt to use
    https://firebase.google.com/docs/web/setup#av
ailable-libraries -->
    <script src=
"https://www.gstatic.com/firebasejs/7.15.4/firebase-a
nalytics.js"
></script>

    <script>
// Your web app's Firebase configuration
var firebaseConfig = {
    apiKey:
"AIzaSyDzN_ZyJb0BSK7CwdChl8GZeZRsnKYaXfA",
    authDomain:

```

Figure 5.1.1(d) register

```
<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey:
    "AIzaSyDzN_ZyJb0BSK7CwdChl8GZeZRsNkYaXfA",
    authDomain:
    "miniproject-fe143.firebaseio.com",
    databaseURL:
    "https://miniproject-fe143.firebaseio.com",
    projectId: "miniproject-fe143",
    storageBucket:
    "miniproject-fe143.appspot.com",
    messagingSenderId: "974920227667",
    appId:
    "1:974920227667:web:24cb351949c4afa9af5fa6",
    measurementId: "G-6TLPERNQYB"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>
</body>
</html>
```

Figure 5.1.1(e) register

5.1.2 Login:

Once the user has successfully registered himself he can login and view his profile

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content=
"width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href=
"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstra
p.min.css"
  integrity=
"sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1
dKGj7Sk"
  crossorigin="anonymous">
  <script src=
"https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity=
"sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+Or
CXaRkfj"
  crossorigin="anonymous"></script>
  <script src=
"https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.mi
n.js"
  integrity=
"sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvo
xMfooAo"
  crossorigin="anonymous"></script>
  <script src=
```

Figure 5.1.2(a) Login


```

"https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.
min.js"
  integrity=
"sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7B
h/kR0JKI"
  crossorigin="anonymous"></script>
  <link rel="stylesheet" href="styles.css">
</head>

<body>

  <div class="m-5 login-box">

    <h1>Login!!</h1>
    <div class=" textbox ">
      <input type="text" id="uname" placeholder=
"Username">
    </div>
    <div class=" textbox">
      <input type="password" id="pass" placeholder=
"Password">
    </div>
    <button type="submit" onclick="login()" class=
"button ">log in</button></a>
    <p align="center">New to the app? <a href=
"Register.html" style="color:#4caf50"> Register Here!</a></p>

```

Figure 5.1.2(b) Login

```

<!-- The core Firebase JS SDK is always required and must be list
ed first -->
  <script src=
"https://www.gstatic.com/firebasejs/7.15.4/firebase.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
  https://firebase.google.com/docs/web/setup#available-libr
aries -->
  <script src=
"https://www.gstatic.com/firebasejs/7.15.4/firebase-analytics.js"
></script>

<script>
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyDzN_ZyJb0BSK7CwdChl8GZeZRsNkYaXfA",
  authDomain: "miniproject-fe143.firebaseio.com",
  databaseURL: "https://miniproject-fe143.firebaseio.com",
  projectId: "miniproject-fe143",
  storageBucket: "miniproject-fe143.appspot.com",
  messagingSenderId: "974920227667",
  appId: "1:974920227667:web:24cb351949c4afa9af5fa6",
  measurementId: "G-6TLPERNQYB"
};
// Initialize Firebase

```

Figure 5.1.2(c) Login

```

<script>
// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyDzN_ZyJb0BSK7CwdChl8GZeZRsnKYaXfA",
  authDomain: "miniproject-fe143.firebaseio.com",
  databaseURL: "https://miniproject-fe143.firebaseio.com",
  projectId: "miniproject-fe143",
  storageBucket: "miniproject-fe143.appspot.com",
  messagingSenderId: "974920227667",
  appId: "1:974920227667:web:24cb351949c4afa9af5fa6",
  measurementId: "G-6TLPERNQYB"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
firebase.analytics();
</script>

<script src="login.js"></script>

</body>
</html>

```

Figure 5.1.2(d) Login

5.1.3 Validation of Username:

The username that is predefined and given to the user is validated using some mechanisms supported by firebase.

5.1.4 Validation of Password:

Password is validated using Firebase mechanisms and it should be minimum of 8 characters.

5.1.5 Firebase Reference:

The user is provided with a simple and robust interface where he is provided with an option to add the vaccine from the recommendation tab. After he clicks the add button a Firebase Instance or an Object is created which acts as a mediator between user and firebase, the list present in history is updated with the name of the vaccine. The list on the history tab is maintained on the backend using Firebase real-time database. The vaccine on which the user clicks the add button is removed from his recommendation as to prevent him from confusion.


```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>profile</title>
  <script src="home.js"></script>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity=
"sha384-Vkoo8x4CGs03+Hhvx8TQ5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849b1E2+poT4WnyKhv5vZF5SrPo01EjwBvKU7imGFAV0wwj1yYfoR5Joz+n" crossorigin
="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity=
"sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOVY5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity=
"sha384-wfSDf2E80Y2D1uUdJ003uMBJnjuUD4Ih7YwaYd1iQktj0Uod8GCExl30g8ifwB6" crossorigin="anonymous"></script>
  <link rel="stylesheet" href="profile_style.css">
</head>
<body onload="render()">

  <script src="https://www.gstatic.com/firebasejs/7.15.4/firebase.js"></script>

  <!-- TODO: Add SDKs for Firebase products that you want to use
    https://firebase.google.com/docs/web/setup#available-libraries -->
  <script src="https://www.gstatic.com/firebasejs/7.15.4/firebase-analytics.js"></script>

  <script>
    // Your web app's Firebase configuration
    var firebaseConfig = {
      apiKey: "AIzaSyDzN_ZyJb0BSK7CwdChl8GZeZRsNkYaXfA",
      authDomain: "miniproject-fe143.firebaseio.com",
      databaseURL: "https://miniproject-fe143.firebaseio.com",
      projectId: "miniproject-fe143",
      storageBucket: "miniproject-fe143.appspot.com",
      messagingSenderId: "974920227667",
      appId: "1:974920227667:web:24cb351949c4afa9af5fa6",
      measurementId: "G-6TLPERNQYB"
    };
    // Initialize Firebase
    firebase.initializeApp(firebaseConfig);
    firebase.analytics();
  </script>
  <script src="home.js"></script>

```

Figure 5.1.5(a) firebase reference

```

<form method="post">
  <div class="row">
    <div class="col-md-4">
      <div class="profile-img">
        
        <div class="file btn btn-lg btn-primary">
          Change Photo
          <input type="file" name="file"/>
        </div>
      </div>
    </div>
    <div class="col-md-6" id="pro">
      <!-- profile part -->
    </div>
    <div class="col-md-2">
      <input type="submit" class="profile-edit-btn" name="btnAddMore" value="Edit Profile"/>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4">
      <!-- INFO PART -->
      <div id="about" class="profile-work"></div>
    </div>
    <div class="col-md-8">
      <div class="tab-content profile-tab" id="myTabContent">
        <div class="tab-pane fade " id="his" role="tabpanel" aria-labelledby="profile-tab">
          <!-- HISTORY PART -->
          <div id="history"></div>
        </div>
        <div class="tab-pane fade active show" id="req" role="tabpanel" aria-labelledby="required-tab">
          <!-- VAC PART -->
          <style>
            .circle {
              height: 22px;
              width: 22px;
              border-radius: 50%;
            }
          </style>
          <div id="required"></div>
        </div>
      </div>
    </div>
  </div>
</form>
<script>

```

Figure 5.1.5(b) firebase reference

```

<div class="col-md-8">
  <div class="tab-content profile-tab" id="myTabContent">
    <div class="tab-pane fade " id="his" role="tabpanel" aria-labelledby="profile-tab">
      <!-- HISTORY PART-->
      <div id="history"></div>
    </div>
    <div class="tab-pane fade active show" id="req" role="tabpanel" aria-labelledby="required-tab">
      <!-- VAC PART-->
      <style>
        .circle {
          height: 22px;
          width: 22px;
          border-radius: 50%;
        }
      </style>
      <div id="required"></div>
    </div>
  </div>
</div>
</form>
<script>
  function abcd(id){

    //console.log(id);
    document.getElementById(id).className="btn btn-success";
    document.getElementById(id).innerHTML="Added";
    var user=localStorage.getItem("username");
    var obj={};
    obj[id]=1;
    //console.log(obj);
    var db=firebase.database().ref().child(user).child("history");
    db.update(obj);
    var db=firebase.database().ref().child(user).child("vaccine").child(id);

    db.update({"taken":"1"});
    document.getElementById(id+"d").style="display:none";
    //window.location.reload();
  }
</script>
</body>
</html>

```

Figure 5.1.5(c) firebase reference

5.2 TESTING

5.2.1 Front page:

REGISTER

Full Name

Email Id

dd-mm-yyyy

Mobile Number

Username

password

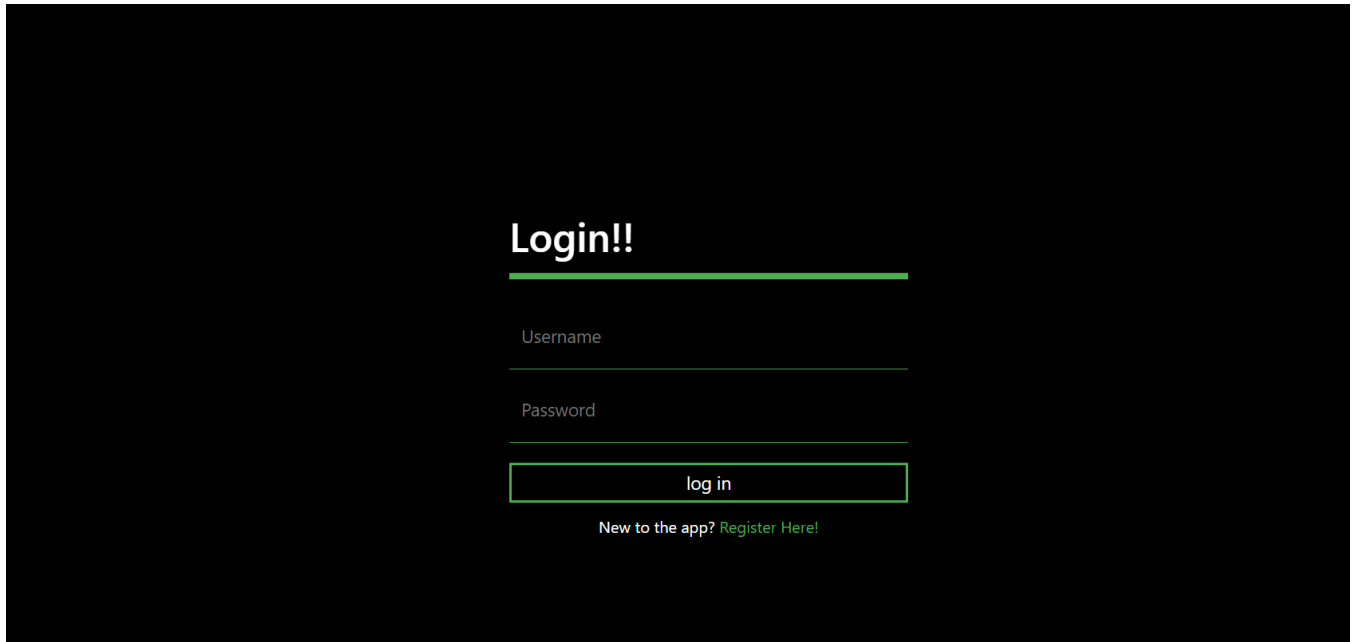
--select nation--

Submit

Already have account? [Login Here!](#)

Figure 5.2.1(a) register UI

5.2.2 Login page:

A login page UI mockup with a dark background. The title "Login!!" is centered in white, followed by a thick red horizontal line. Below the line are two input fields: "Username" and "Password", each with a red underline. A red-bordered button labeled "log in" is centered below the fields. At the bottom, the text "New to the app? Register Here!" is displayed, with "Register Here!" in red.

Login!!

Username

Password

log in

New to the app? [Register Here!](#)

Figure 5.2.2(a) Login UI

5.2.3 Profile page(Recommendations):

When the user logs in the recommended tab is displayed, it gives him information about vaccines to take and when to take them, when the user takes a vaccine and clicks ADD NOW, the vaccine disappears from the recommendation tab and is recorded in history.

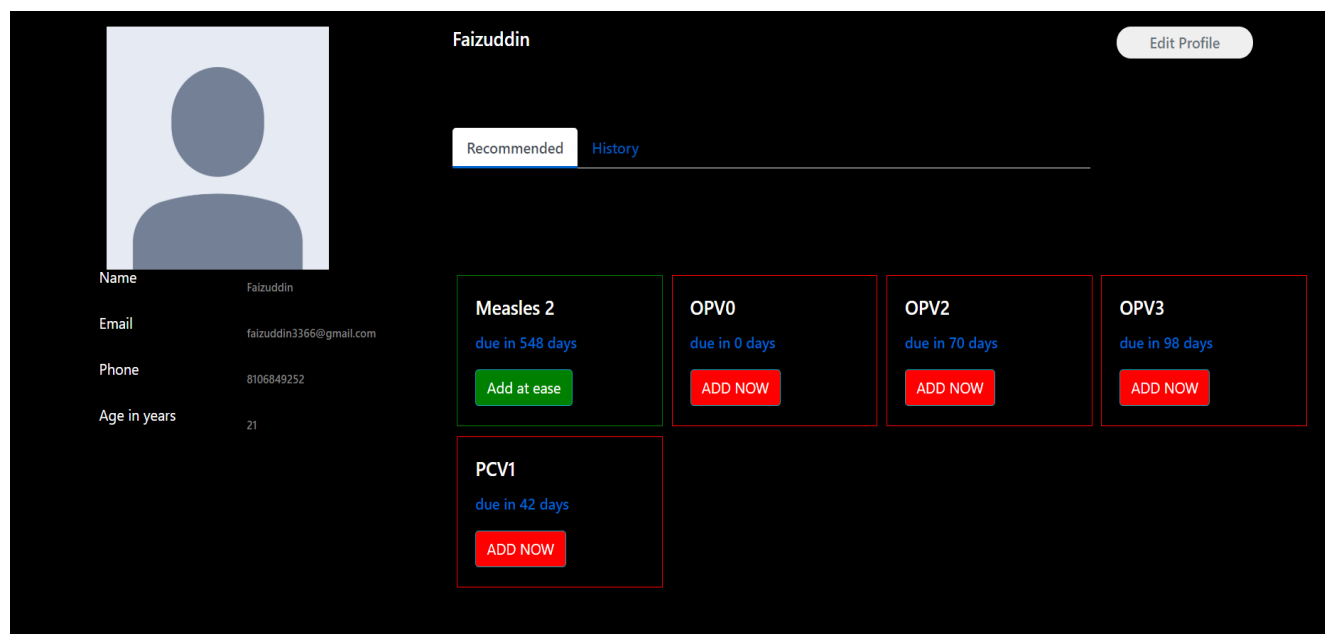


Figure 5.2.3(a) Home UI

5.2.3 Profile page(history):

The history records the vaccines taken by the user

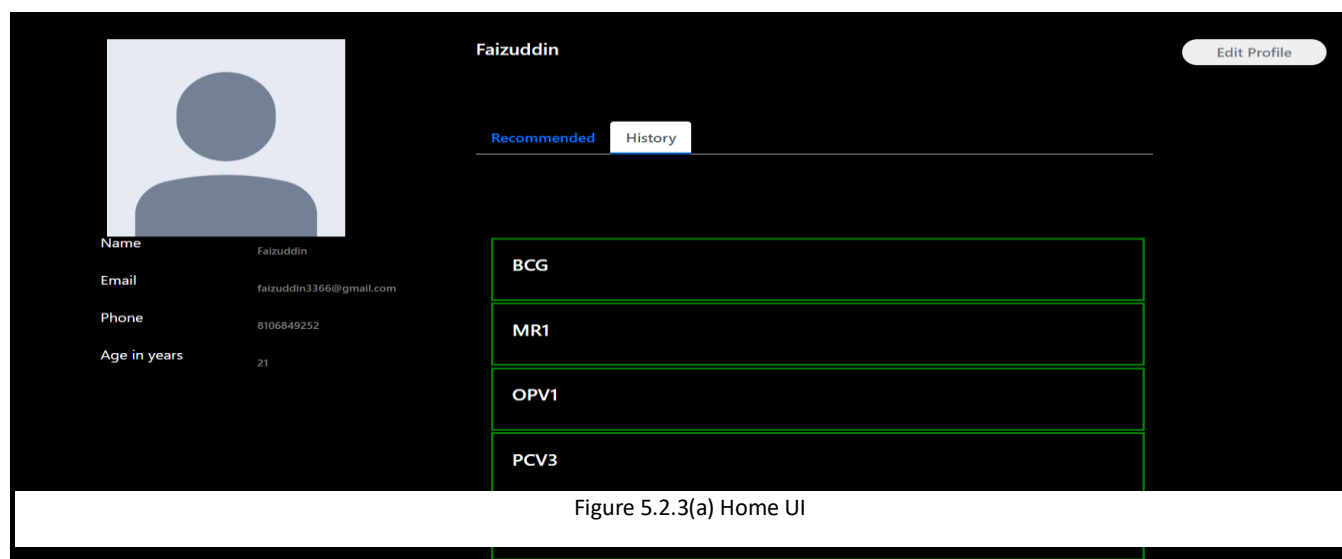


Figure 5.2.3(a) Home UI

6. CONCLUSION & FUTURE SCOPE

6.1 Conclusion:

- In the modern world, the use of computers is becoming rampant, utilizing this factor will be increased.
- This report presents simple, cost-effective and most importantly effective online vaccination records
- This application is useful to both migrants and doctors.
- To safeguard people's lives and health, application for recommending vaccinations and keeping record of them is key as this will effectively bridge the communication gap between the doctors and the general public in fighting diseases.

6.2 Future Scope:

- ☐ We can further enhance this technology and make it a mobile application so people can access that with much more ease than before
- ☐ The other feature we can add to this is live chat with a few doctors so people can speak to a doctor in their times of need

REFERENCES

1. **Firestore API:** <https://firebase.google.com>
2. **Vaccination records:** <https://www.who.int/topics/vaccines/en/>
3. **Bootstrap:** <https://getbootstrap.com/docs/4.0/getting-started/introduction/>

