1.Introduction

● Background and Motivation

Childhood vaccination is one of the most effective public health interventions for preventing the spread of infectious diseases and protecting the health of children. Vaccination programs have significantly reduced the incidence of serious illnesses, disabilities, and deaths caused by vaccine-preventable diseases like measles, polio, and whooping cough. However, despite the proven benefits of vaccination, many children around the world still miss out on timely vaccinations due to various factors, including forgetfulness, lack of awareness, or logistical challenges.

Relevance:

The automated child vaccination alert system is highly relevant due to the following reasons:

1. Public Health Impact: Timely and complete vaccination is essential to maintain herd immunity, prevent disease outbreaks, and reduce the burden on healthcare systems. Automated alerts can help increase vaccination rates, ultimately contributing to public health.

2. Parental Forgetfulness: Parents and guardians often struggle to keep track of their child's vaccination schedule, leading to missed vaccinations. Automated alerts provide a convenient solution to remind them when vaccines are due.

3. Improved Health Outcomes: A well-implemented system can significantly reduce the occurrence of vaccine-preventable diseases and their associated complications, leading to better health outcomes for children.

4. Reduced Healthcare Costs: Fewer cases of vaccine-preventable diseases result in cost savings for healthcare systems and families who would otherwise bear the expenses of treating sick children.

5. Data Management: Automated systems can efficiently manage and maintain vaccination records, making it easier for healthcare providers to access and update patient information.

6. Accessibility: The system can reach a wide range of parents, including those in remote areas or underserved communities, provided they have access to basic technology.

Real-World Problems Addressed:

1. Vaccination Compliance: A significant problem is low vaccination compliance, where children miss or delay their immunizations, leaving them vulnerable to preventable diseases.

2. Data Management: Healthcare providers often struggle with maintaining accurate and up-to-date vaccination records, which can lead to missed opportunities for vaccination.

3. Communication Gaps: In busy lives, parents can overlook vaccination schedules, and healthcare providers may have limited means to effectively remind them.

4. Health Disparities: There are disparities in vaccination rates among different demographic groups, including those with limited access to healthcare or information.

5. Emergency Preparedness: Rapid response to disease outbreaks or changes in vaccination recommendations can be challenging without efficient communication channels.

6. Data Privacy and Security: Safeguarding vaccination and personal health data is crucial to ensure patient privacy and maintain public trust in vaccination programs.

An automated child vaccination alert system addresses these real-world problems by leveraging technology to improve vaccination rates, streamline data management, enhance communication, and ensure that children receive essential vaccinations on time, thereby reducing the burden of vaccine-preventable diseases on public health systems and families.

● Problem Statement

Nearly 2 Million children under the age of 5 years die every year in India .The Indian Academy of Pediatrics (IAP) estimates that over 50 percent of these are vaccine preventable. In order to prevent the infants

from hazardous diseases such as small pox , hepatitis B , tetanus ,etc. we've taken this initiative .

This website is easy and convenient to use and it maintains the data of users and the main motto of our vaccination model is to provide convenience to the parents. The desired impact of this app is to ensure that the children receive all the mandatory vaccination on time.

At the time of child birth the hospital uploads the child details in this website. The vaccination alerts will be sent to the parents from three days prior to the vaccination date when the hospital clicks on a button in this website.

Vaccination model is specially designed for maintaining the health of babies. So, parents need not to worry about remembering the dates of their child's vaccines, they automatically receive reminder notifications.

The VACCINATION ALERT app provides 3 reminder alerts for the caregiver in a week that the vaccinations are due. The app requires data of both the child and the parent or caregiver. It's the responsibility of the caregiver to provide the correct information at the hospital. The website uses the information provided to schedule the reminder alert.

● Objectives

1. Improve Child Immunization Rates: Ensure that a higher percentage of children receive timely vaccinations by sending automated alerts to parents and guardians.

2. Timely Reminders: Provide timely and personalized vaccination reminders to parents and guardians, helping them stay on schedule with their child's immunization needs.

3. Reduction of Vaccine-Preventable Diseases: Contribute to a reduction in vaccine-preventable diseases by increasing the number of children who are fully vaccinated.

4. Efficient Data Management: Implement a system for efficiently storing and managing vaccination records, ensuring easy access for healthcare providers and parents.

5. Customized Alert Preferences: Allow parents and guardians to set their preferred communication channels (e.g., SMS, email) and frequencies for vaccination reminders, making it more convenient for them.

6. Accessibility and Inclusivity: Ensure that the system is accessible to a wide range of users, including those with different language preferences and varying levels of technology literacy.

7. Compliance with Health Regulations: Ensure that the system adheres to local and national health regulations regarding vaccination schedules and data privacy.

8. Integration with Healthcare Providers: Facilitate communication and coordination between parents, healthcare providers, and public health agencies, streamlining the vaccination process.

9. Data Security and Privacy: Implement robust security measures to protect sensitive vaccination records and personal information of children and their families.

10. User Education: Provide educational resources and information about the importance of vaccination, addressing common concerns and misconceptions.

11. Monitoring and Reporting: Enable tracking and reporting features to monitor vaccination rates, compliance, and system performance, helping public health agencies make data-driven decisions.

12. Scalability and Sustainability: Design the system to accommodate an increasing number of users and changing vaccination requirements, ensuring its long-term sustainability.

13. Cost-Efficiency: Optimize the system to be cost-effective in terms of development, maintenance, and operational expenses.

14. User Satisfaction: Collect feedback from users to continuously improve the system and enhance user satisfaction.

15. Emergency Notifications: Include a feature to send emergency alerts and updates related to vaccines, such as disease outbreaks or changes in vaccination schedules.

16. Cultural Sensitivity: Ensure that the system respects and takes into account cultural and religious differences that may influence vaccination decisions.

17. Community Engagement: Encourage community involvement and support for vaccination programs, promoting a collective responsibility for child health.

18. User Support and Assistance: Provide customer support and assistance to address user inquiries, troubleshoot technical issues, and guide users through the system.

19. Research and Analysis: Enable the system to collect data for research and analysis of vaccination trends, contributing to public health research and policy decisions.

20. Public Awareness: Support public awareness campaigns about the benefits of vaccination and the role of the system in promoting child health.

● Scope and Limitations

SCOPE

The scope of the automated child vaccination alert minor project encompasses the development and implementation of a user-friendly software solution designed to send automated vaccination reminders to parents and guardians. This system will allow hospitals to input the contact information, receiving timely alerts via the parents preferred communication channels (e.g., SMS, email). The project includes the design of a user interface for data input, a database for storing vaccination records, and a notification system. The scope also covers the integration of relevant healthcare guidelines and regulations, ensuring data security, privacy, and accessibility. While the project aims to improve vaccination rates and reduce vaccine-preventable diseases, it focuses on a limited geographic area and a specific user base for the purposes of this minor project.

LIMITATIONS

Certainly, here are some potential limitations of an automated child vaccination alert system for a minor project, presented as bullet points:

1. Limited Outreach: The system's effectiveness heavily relies on the availability of contact information for parents or guardians, which may not cover all children in the target population.

2. Data Accuracy: The accuracy of vaccination records is contingent upon the input of correct information by parents or healthcare providers, and errors or omissions may occur.

3. Dependency on Technology: The system assumes that users have access to and are comfortable using technology (e.g., smartphones, internet) to receive alerts, potentially excluding those without such access.

4. Language and Literacy Barriers: Parents who speak languages other than the supported languages or have limited literacy may not fully benefit from the system.

5. Privacy Concerns: Some parents may be apprehensive about sharing personal and vaccination data through the system, raising privacy concerns that could limit its adoption.

6. User Engagement: Not all parents or guardians may consistently engage with the alerts or prioritize vaccinations, which can affect vaccination compliance.

7. System Reliability: Technical issues, such as server outages or software glitches, can disrupt the system's ability to send timely alerts.

8. Cost Constraints: Developing, hosting, and maintaining the system could be cost-prohibitive for a minor project, potentially limiting its scalability and sustainability.

9. Geographical Reach: The system may not be effective in reaching children in remote or underserved areas where access to healthcare and technology is limited.

10. Parental Compliance: The system cannot guarantee that parents will follow through with vaccination recommendations after receiving alerts.

11. Emergency Situations: While the system can send routine reminders, it may not be equipped to handle emergency situations, such as sudden disease outbreaks or rapid vaccination schedule changes.

12. Maintenance and Updates: Regular maintenance and updates are necessary to keep the system functional and up-to-date with changing vaccination recommendations and technologies.

● Organization of the Document

1. Introduction
    - ● Background and Motivation
    - ● Problem Statement
    - ● Objectives
    - ● Scope and Limitations

2. Literature Review
    - ● Review of Relevant Research Papers and Projects

- Theoretical Framework
- Related Technologies and Tools

3. Methodology
    - Description of the Approach/Method Used
    - Software/Hardware

4. Implementation
    - Detailed Explanation of How the Project Was Executed
    - Code Structure and Architecture
    - Technical Challenges and Solutions
5. Results
    - Presentation of Results (Tables, Charts, Graphs)
    - Analysis and Discussion of Results
    - Comparison with Expected Outcomes
6. Conclusion
    - Summary of Achievements
    - Contributions to the Field
    - Future Work and Recommendations
7. References
    - List of All Cited Sources (Books, Journals, Websites, etc.)
    - Follow a Citation Style (e.g., APA, IEEE)
8. Appendices
    - Additional Technical Details
    - Code Snippets

2. Literature Review

● Review of Relevant Research Papers and Projects

● Theoretical Framework

Theoretical Framework for Automated Child Vaccination Alerts Using React:

1. User-Centered Design Principles:

- Usability: The system should adhere to usability principles, ensuring that it's easy to navigate and understand for users with varying levels of technical proficiency.

- User Feedback: Incorporate user feedback and iterative design processes to continually enhance the UI.

2. Health Behavior Theory:

- Health Belief Model: Consider integrating aspects of the Health Belief Model to understand how parents perceive the severity and susceptibility of vaccine-preventable diseases and design the UI to address these perceptions.

- Theory of Planned Behavior: Explore how parents' attitudes, subjective norms, and perceived behavioral control affect their intention to follow vaccination recommendations, and design the UI to encourage the desired behaviors.

3. Information Design and Visualization:

- Information Hierarchy: Implement a clear information hierarchy in the UI, prioritizing essential information such as upcoming vaccination dates, vaccination schedules, and contact details.

4. Personalization and Tailoring:

- User Profiles: Develop user profiles that allow parents or guardians to customize notification preferences, such as the choice of communication channels (SMS, email).

- Tailored Messaging: Implement personalized messages and reminders based on the child's vaccination history and specific needs.

5. Mobile-First Design:

- Responsive Design: Ensure that the UI is designed with a mobile-first approach to accommodate users accessing the system on various devices, such as smartphones and tablets.

6. Data Privacy and Security:

- Data Protection Framework: Adhere to data privacy regulations and establish strong security measures to protect sensitive user information and vaccination records.

7. Feedback Loop and Iteration:

- User Feedback Mechanism: Create a mechanism for users to provide feedback on the UI and their experience with the system, facilitating continuous improvement.

8. Health Communication Strategies:

- Effective Messaging: Employ best practices in health communication to ensure that the UI conveys the importance of vaccination and addresses common concerns and misconceptions.

9. Integration with Backend and Data Management:

- React as a Frontend Framework: Recognize React as the chosen frontend framework and integrate it seamlessly with the backend, which manages vaccination data and schedules.

● Related Technologies and Tools

Technologies: React , Node , MongoDB , Express

Services : Gmail ,Twilio

The following picture gives the mongoDB snapshot of the hospital.

The following mongoDB snapshot is the details of the children which also includes the parent contact information.

MongoDB is a popular NoSQL database designed for flexibility and scalability. It's an excellent choice for storing user details. In MongoDB, data is organized into collections, each containing documents in BSON format, which is similar to JSON. User details, such as name, email, and profile information, can be stored as documents within a "users" collection. Each document represents an individual user, making it easy to retrieve and update user data. The schema flexibility allows for dynamic fields, making it adaptable to evolving user requirements. Additionally, MongoDB supports powerful querying and indexing for efficient data retrieval, making it suitable for a wide range of applications.

Twilio's messaging service is a cloud-based platform that empowers businesses and developers to send and receive SMS (Short Message Service) and MMS (Multimedia Messaging Service) messages programmatically. It offers a robust API for seamless integration into applications, enabling two-way communication with users via text messages. Twilio's global reach allows for international messaging,

handling complexities like carrier regulations. It supports rich content delivery, including images and videos. Developers can track message status and delivery receipts, ensuring reliable communication. Twilio's message service is commonly used for applications like appointment reminders, notifications, customer support, and marketing campaigns, providing a versatile solution for SMS and MMS communication.

Gmail is a popular email service provided by Google. It offers users a free, web-based email platform with a user-friendly interface. Some key features of Gmail include ample storage space, efficient email organization through labels and filters, powerful search capabilities, and integration with other Google services like Google Drive and Google Calendar. It supports both personal and business accounts, and its mobile app allows access on various devices. Gmail also includes robust spam filters and security features to protect users from phishing and malware. Its widespread use and continuous updates make Gmail a reliable and versatile choice for sending, receiving, and managing emails.

3. Methodology

- **Description of the Approach/Method Used**

Vaccination Alert attempts to find a solution for the parents specially for those who forget about their babies vaccines. This product increases the vaccination rate also.

The main functionalities of this application is:

- Register to the software.
- Login/Logout.
- Send Vaccination Alert to the users.
- Send message for updating infants records.
- Provide Doctors.

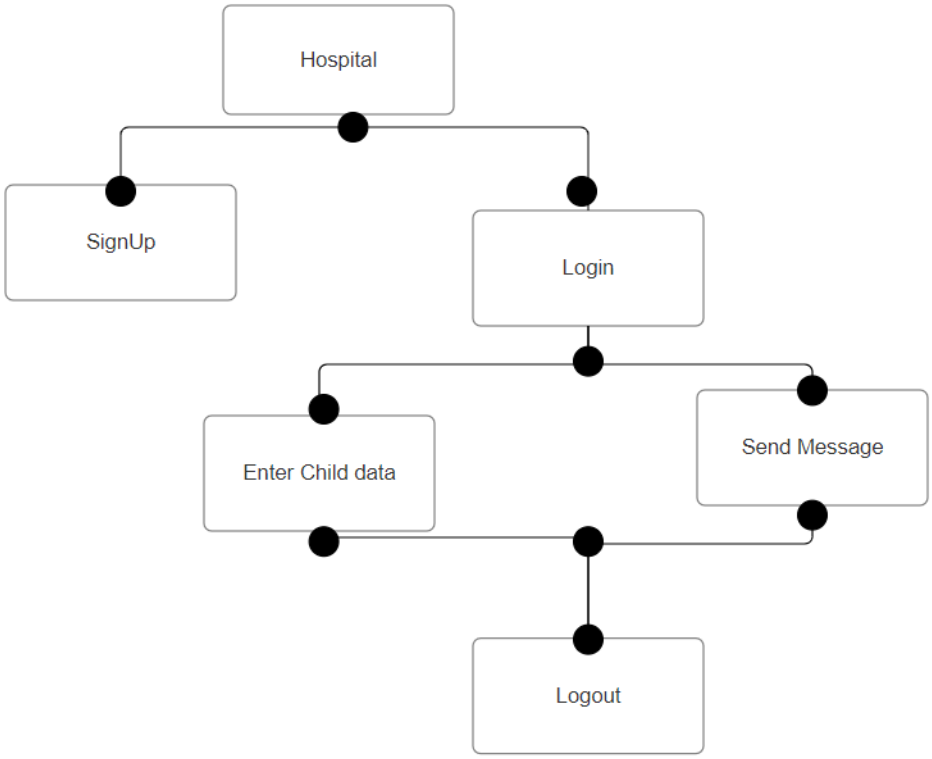
Initially every hospital has to register by providing the credentials. When the child is born the details have to be uploaded in the website by logging in with the registered details.

When a registered hospital logins they can undertake two type of actions.

1.Entering the details of the new born child and saving them.

2.Sending the vaccination alert. It can be done just by clicking a button . The details of the parents who had been notified will be displayed.

3.when the parent have a registered email id the alert is being done through email else it will be done using the message.



● Software/Hardware Used (if applicable)

This system is designed to be transparent to its users and hence all the complexity is hidden from the user, i.e., user has no need to take care about the internal working. The user will interact with system using the GUI.

➢ ***User Interfaces***

- This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.
- SMTP protocol is used for sending mails.

➢ **Hardware Interfaces**

No specific hardware is required. The website needs just a software compatible hardware on which website can run.

- **Operating System:** We have chosen windows operating system for its best support, performance and user friendliness.
- **Database:** To save the records of the users and their details, SQL database is used.

4. Implementation

● Detailed Explanation of How the Project Was Executed

It is a react application connected to the mongoDB database to store the details of the hospital and the child details along with the contact details.

The user interface is built using the components by the integration with the bootstrap in order to enhance the responsiveness.

When we open the editor and we run the react application the website will be running in the port 3000.

There will be a dashboard containing the login button and signup button.

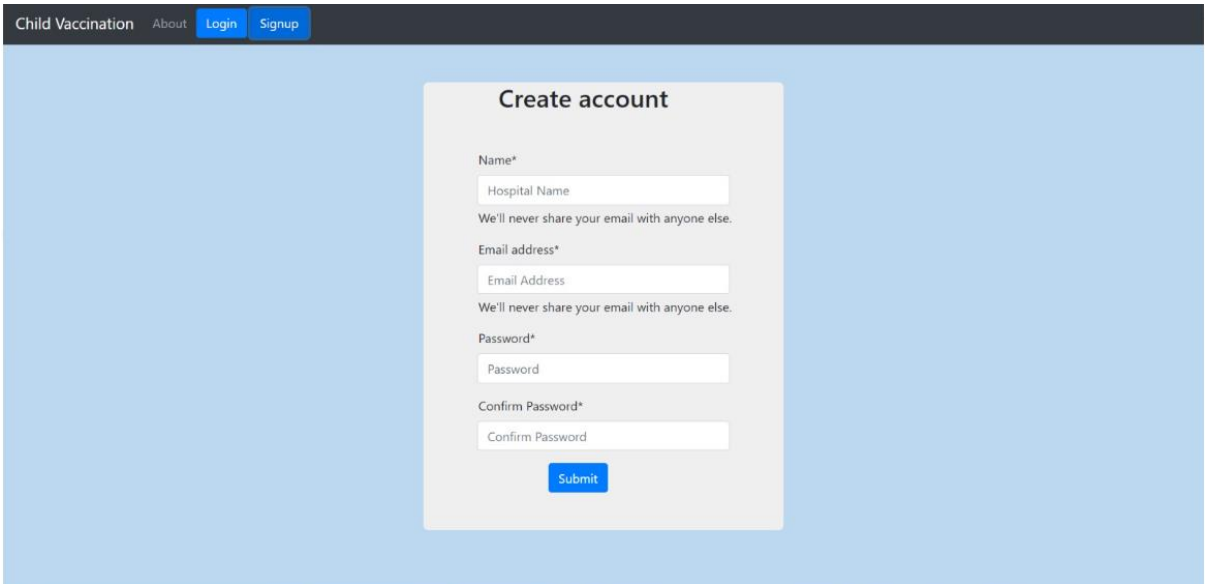
Signup button functionality is similar to register and the registered hospitals can use the login button.

The dashboard will be as follows.

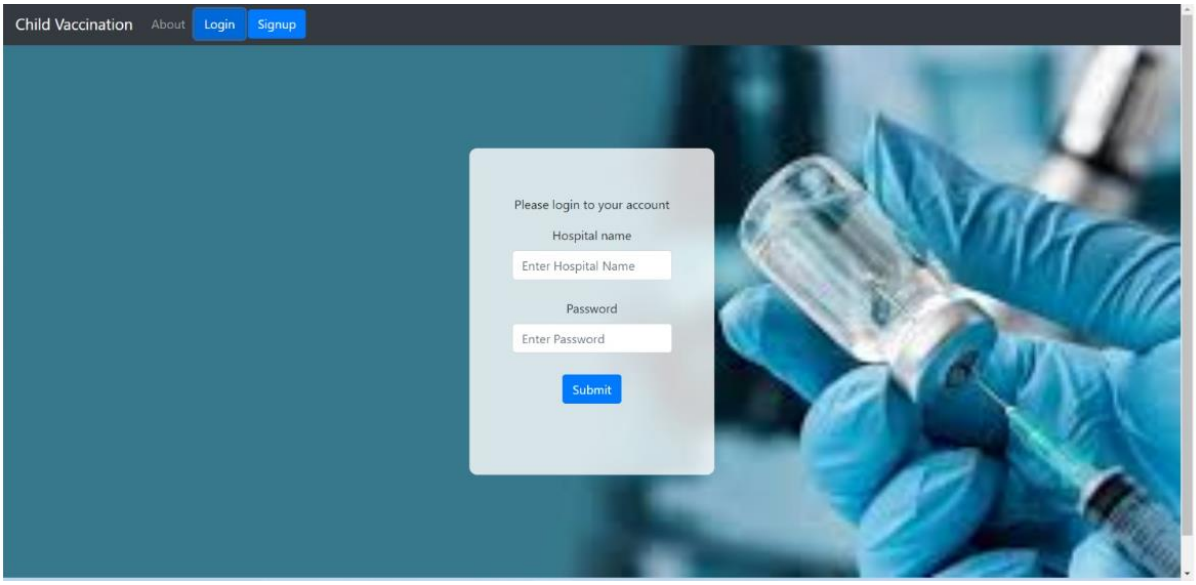
Hospital Functionality:-

- **Login/Logout** –Hospital has authority to login/logout to the system and manage the software.

- **Generate -** Hospital can generate infants record.
- Sign up page will be as follows
- Hospital can create account by entering the credentials.



- **View infants record –** To check infants growth, hospital can view infants record.
- Hospital has to use the specified button to send message to parents.



When a hospital already registered then to upload the child data the hospital can provide the name and password in order to login.

Once the hospital logins then they can perform two types of operations.

They are uploading the child information and send the message to the parents by clicking a button.

While sending the message we are using two types of services.

1.Twilio

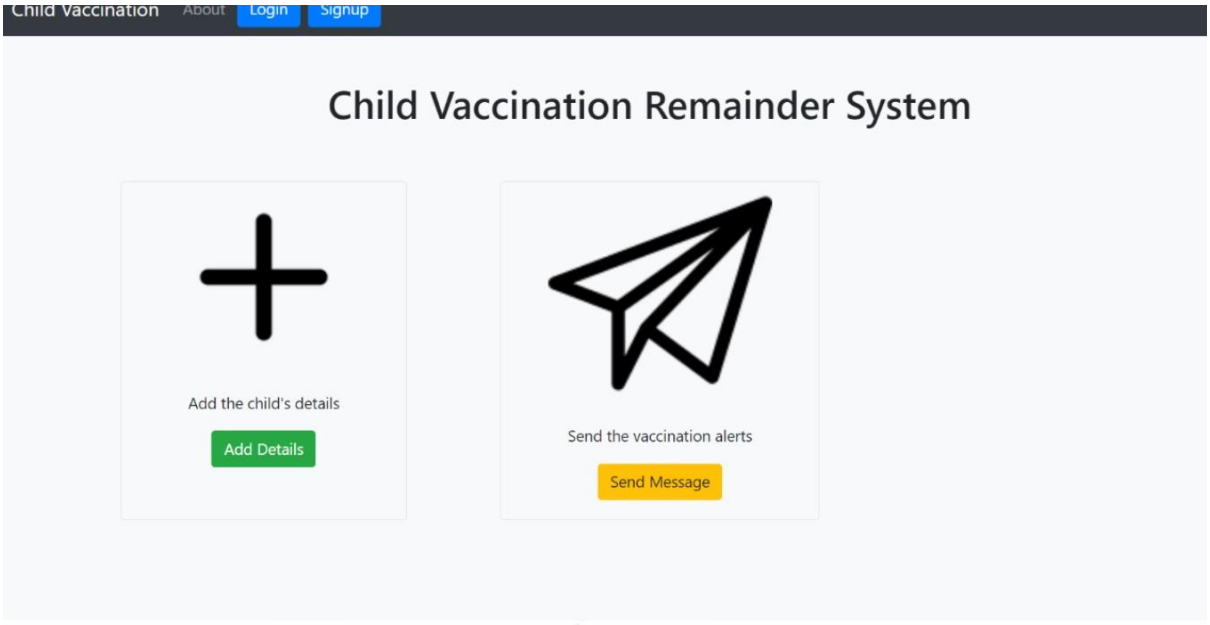
2.Gmail

The parents who have registered mail id the notification will be sent using the gmail service.

And those who don’t have mail id will receive the message alert.

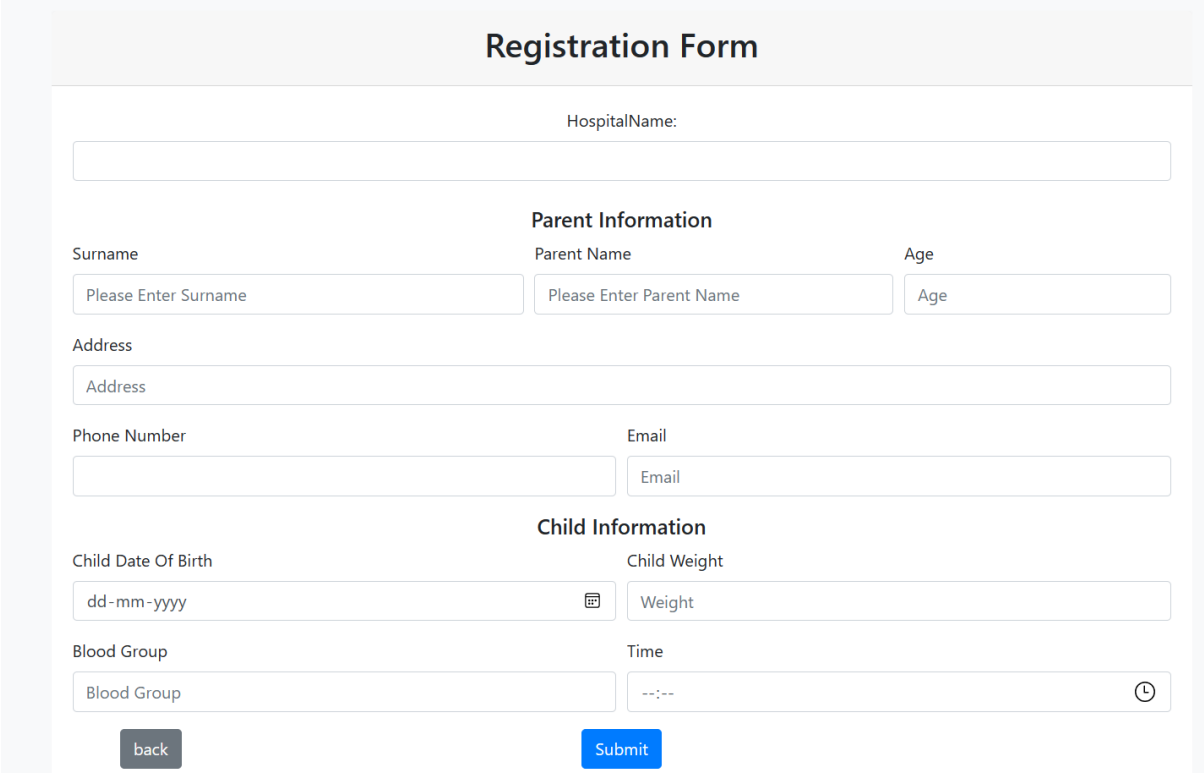
Message contains the information regarding the child vaccination date and the type of vaccine.

These are the options that are visible to the logged in hospital user.

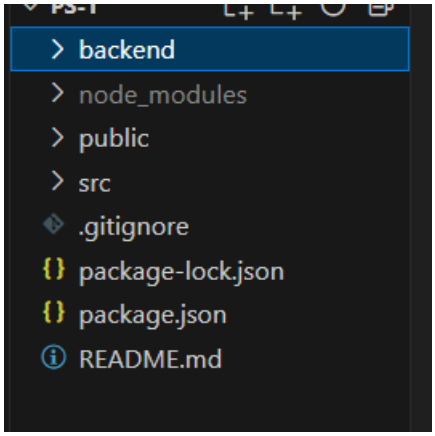


Caretaker's Functionality:-

- **Provide information –** Caretaker will give all the details of their child like age, weight, height, any disabilities etc.
- Provide the contact information to receive the alerts.
- **Receive Notifications**
- The caretaker have to provide the details in the registration form regarding the contact information and the child details.



• Code Structure and Architecture



The components in the application are placed in the components folder inside the src folder.

All the routes and database connectivity is present in the backend folder.

Package.json contains information regarding all the dependencies and installations in the project.

● Technical Challenges and Solutions

The major challenge which we had faced is the user authentication which we had overcome using the jsonwebtoken. It is also helpful in order to store the session data.

The validation of the fields is being handled while storing the data in the database.

5. Results

● Presentation of Results (Tables, Charts, Graphs)

● Analysis and Discussion of Results

Advantages:

- **Vaccination Alert** app is a native app. Also you can access it on browser.
- It’s user-friendly.
- Very less memory is required.

- It decreases the overhead of Parents via sending reminder to them.
- In this software, Caretaker can consult with doctors also.
- At present it's available in English, but very soon it'll be available in English and Hindi, both the languages.

Disadvantages:

- Parents should have knowledge of operating mobile phones on which this app can run.
- Availability of Internet is must.
- Admin has to manually keep updating the information by entering the details it the system.
- Doctor will be available online only.

● Comparison with Expected Outcomes

6. Conclusion

• Summary of Achievements

An automated child vaccination alert system has the potential to achieve several significant milestones and benefits, including:

1. Improved Immunization Rates: The system can significantly boost child immunization rates by sending timely reminders to parents and caregivers about upcoming vaccinations. This ensures that children receive their shots on schedule, reducing the risk of preventable diseases.

2. Timely Notifications: The system is designed to send automated alerts, either via SMS, email, making it easy for parents to remember and act on vaccination appointments.

3. Reduced Missed Vaccinations: By providing continuous reminders and tracking vaccination records, the system helps reduce the number of missed vaccinations, contributing to better community immunity.

4. Healthier Children: By ensuring children are up to date with their vaccinations, the system contributes to overall child health, preventing potentially life-threatening illnesses.

5. Lower Healthcare Costs: Timely vaccinations result in fewer hospitalizations and medical expenses, saving healthcare costs for both families and governments.

6. Data Management: The system can help maintain accurate vaccination records for children, making it easier for healthcare providers to track their immunization history and plan future vaccinations.

7. Real-time Updates: The system can incorporate real-time updates on vaccines and any changes in immunization schedules, ensuring that children are always receiving the most current and effective vaccines.

8. Personalized Reminders: The system can be tailored to send personalized reminders, accommodating specific vaccine schedules and individual child needs.

9. Public Health Impact: By promoting widespread vaccination, the system contributes to community immunity, helping protect vulnerable populations, such as infants and those with compromised immune systems.

10. Research and Reporting: The system can also aid in data collection and reporting for public health agencies, facilitating research and analysis of vaccination trends and coverage rates.

In summary, an automated child vaccination alert system can achieve several essential milestones, from improving immunization rates and child health to reducing healthcare costs and aiding public health efforts. It plays a crucial role in promoting preventive healthcare and protecting children from vaccine-preventable diseases.

• Contributions to the Field

Automated child vaccination alert systems play a crucial role in ensuring that children receive timely vaccinations, which are essential for their health and well-being. Several contributions have been made to this field to enhance the efficiency and effectiveness of such systems:

1. Electronic Health Records (EHR) Integration: Integrating automated child vaccination alert systems with electronic health records helps healthcare providers keep track of a child's vaccination schedule. This ensures that healthcare professionals have up-to-date information on a child's vaccination status, allowing for timely alerts and reminders.

3. Text Messages and Phone Calls: Automated systems can send text messages or make automated phone calls to parents or guardians to remind them about upcoming vaccination appointments. This approach is particularly effective in regions with limited smartphone access.

4. Machine Learning and AI: Machine learning and artificial intelligence can help predict when a child is due for their next vaccination based on their vaccination history and age. These systems can also analyze vaccination coverage and identify areas with low immunization rates.

5. Geospatial Data and Mapping: Integrating geospatial data and mapping tools can help identify under-vaccinated areas and target vaccination campaigns more effectively. This is especially valuable for ensuring equitable access to vaccines.

6. Multilingual Support: In areas with diverse populations, providing vaccination alerts in multiple languages can help reach a broader audience and ensure that language barriers do not impede vaccination efforts.

7. Immunization Registries: Many regions have established immunization registries to maintain comprehensive records of vaccinations. Automated alert systems can access and use data from these registries to send timely reminders to parents and guardians.

8. Data Analytics and Reporting: Automated systems can provide healthcare professionals and public health authorities with data analytics and reporting tools. These tools can help in monitoring vaccination coverage rates and identifying areas that require targeted interventions.

9. Integration with Public Health Initiatives: Some automated alert systems are integrated with broader public health initiatives. For example, they can be linked to campaigns for distributing vaccines, conducting outreach programs, and providing educational resources to parents and communities.

10. Behavioral Insights: Research in behavioral science has contributed to the design of more effective alert messages. Understanding how people make decisions about vaccinations can help create messages that are more likely to motivate parents to get their children vaccinated.

11. Emergency Alert Systems: In times of outbreaks or emergencies, automated alert systems can send urgent notifications to parents and healthcare providers, encouraging immediate vaccination to prevent the spread of diseases.

12. Data Privacy and Security: Efforts have been made to ensure the privacy and security of the data within these systems, as they often contain sensitive healthcare information. Compliance with data protection regulations is crucial.

Automated child vaccination alert systems continue to evolve with advancements in technology, data analysis, and healthcare practices. Their contributions to the field are critical for achieving high vaccination coverage rates and reducing the incidence of vaccine-preventable diseases in children.

● Future Work and Recommendations

Creating an automated child vaccination alert system is a crucial step in ensuring that children receive timely vaccinations, which are vital for their health and the prevention of disease outbreaks. Here are

some future work and recommendations for improving such a system:

1. Integration with Electronic Health Records (EHRs):

- Collaborate with healthcare providers and institutions to integrate the alert system with electronic health records. This will enable real-time access to a child's vaccination history, making the alerts more accurate and tailored to individual needs.

2. Customization and Personalization:

- Develop the system to consider the specific vaccination schedule for each child based on factors like age, medical history, and geographical location. This customization can help in providing more relevant alerts.

3. AI and Machine Learning:

- Implement AI and machine learning algorithms to predict the ideal vaccination schedule for a child based on historical data, health conditions, and demographic information. This can reduce the likelihood of missed vaccinations.

4.Phone call:

-Instead of just a notification alert, it will be advanced in case if a call is made and machine learning model will explain about the vaccination schedule.

5. Multilingual Support:

- Ensure the system supports multiple languages to cater to diverse populations, especially in multicultural areas.

6. Geolocation Services:

- Utilize geolocation services to determine the nearest vaccination centers and provide directions to parents or caregivers. This feature can be particularly useful for remote or underserved areas.

7. Integration with Immunization Registries:

- Work with national and regional immunization registries to automatically update vaccination records and confirm completed vaccinations. This can reduce discrepancies and ensure the most up-to-date information.

8. Feedback Mechanism:

- Include a feedback mechanism that allows parents and healthcare providers to report issues or inaccuracies, fostering continuous improvement.

9. Data Security and Privacy:

- Prioritize data security and privacy by implementing encryption, access controls, and compliance with relevant regulations such as GDPR (General Data Protection Regulation).

10. Community Engagement and Awareness:

- Develop educational materials and campaigns to raise awareness about the importance of child vaccination and the use of the automated alert system.

11. Monitoring and Evaluation:

- Continuously monitor the effectiveness of the system in increasing vaccination rates and reducing vaccine-preventable diseases. Make adjustments based on the results of this evaluation.

12. Healthcare Provider Collaboration:

- Foster collaboration with healthcare providers to ensure they are aware of and engaged with the system. This can enhance the accuracy of data and provide a holistic approach to child healthcare.

By implementing these recommendations, an automated child vaccination alert system can become a powerful tool in promoting child health and reducing the incidence of vaccine-preventable diseases. It is essential to adapt and improve the system continuously to ensure its long-term success and effectiveness.

7. References

- **List of All Cited Sources (Books, Journals, Websites, etc.)**

- https://machinelearningmastery.com/
- https://iq.opengenus.org/
- https://en.wikipedia.org/wiki/Generative_adversarial_network
- https://nevonprojects.com/child-vaccination-management-system-using-python/

8. Appendices

● Code Snippets

```
const twilio = require('twilio');
const nodemailer = require('nodemailer');
const Mail = require('nodemailer/lib/mailer');

const accountSid = 'AC14c3738989716815241cede315cdea0d';
const authToken = '9702cb99cfc9623818b50406d647920a';
const twilioNumber = '+16053706224';
const client = twilio(accountSid, authToken);
```

1. `const twilio = require('twilio');`: This line imports the Twilio Node.js module, which allows for programmatic interaction with Twilio's communication services.

2. `const accountSid` and `authToken`: These variables store your Twilio account's unique identifier (SID) and authentication token, which are used to authenticate your application with Twilio's services.

3. `const twilioNumber`: This variable holds the Twilio phone number that will be used as the sender for outgoing SMS messages and phone calls.

4. `const client = twilio(accountSid, authToken);`: This line initializes a Twilio client by passing in your account SID and authentication token. The client is then used to interact with Twilio's services, such as sending SMS messages or making phone calls.

With these settings in place, you can use the `client` to send SMS messages or initiate phone calls programmatically, which is often used in applications for purposes like two-factor authentication, notifications, and more.

1. `const mongoose = require("mongoose");`: This line imports the Mongoose library, which is a popular Node.js library for MongoDB object modeling and interaction.

2. `const mongoURI = "mongodb://0.0.0.0:27017/project";`: This line defines the URI (Uniform Resource Identifier) for the MongoDB database you want to connect to. It specifies the database server's address (0.0.0.0) and port (27017), as well as the name of the database ("project").

3. `const connectDB = async () => { ... }`: This code defines an asynchronous function named `connectDB` that will attempt to connect to the MongoDB database.

4. Inside the `connectDB` function:

- `mongoose.set('strictQuery', false)`: This line disables the strict mode for queries. In Mongoose, strict mode is a setting that enforces a schema structure. Setting it to `false` allows for more flexible data handling.

- `mongoose.connect(mongoURI)`: This line attempts to connect to the MongoDB database using the specified URI.

- `console.log('Mongo connected')`: If the connection is successful, it logs "Mongo connected" to the console.

- `try { ... } catch (error) { ... }`: This code uses a try-catch block to handle potential errors during the database connection process.

- If there is an error during the connection attempt, the error message is logged to the console using `console.log(error)`.

- `process.exit()`: If an error occurs, this line terminates the Node.js process. This is often used in development to exit the application when a critical error is encountered.

5. Finally, `connectDB()`: This line calls the `connectDB` function, which initiates the database connection when the script is executed.

```
const mongoose = require('mongoose');
const { Schema } = mongoose;

const UserSchema = new Schema({
    name:{
        type: String,
        required: true
    },
    email:{
        type: String,
        required: true,
        unique: true
    },
    password:{
        type: String,
        required: true
    },
    date:{
        type: Date,
        default: Date.now
    }
  });
  const User = mongoose.model('user', UserSchema);
  module.exports = User;
```

Hospital SignUp schema

```
const m  const Schema: typeof mongoose.Schema
const {Schema}=mongoose;
const ChildSchema=new Schema({
        parent_name:{type:String,required:true},
        phone_num:{type:String,required:true},
        birth_date:{type:Date,required:true},
        name:{type:String,required:true},
        bloodgroup:{type:String,required:true},
        weight:{type:String,required:true},
        email:{type:String},
        address:{type:String,required:true},
        age:{type:Number,required:true},
        surname:{type:String,required:true},
        time:{type:String,required:true}

},{versionKey:false});
const myModel=mongoose.model("NEWCOL",ChildSchema)
module.exports=myModel;
```

Child Registration form Schema

List of dependencies

The code snippet specified above are list of dependencies project, typically managed using a package manager like npm (Node Package Manager).Each dependency listed has a name and a version number.

Here's a breakdown of the dependencies listed in `package.json` file:

1. @emailjs/browser (Version 3.11.0): A library for sending email directly from the browser using JavaScript. It is often used for client-side email functionality.

2. @testing-library/jest-dom (Version 5.16.5): Part of the Testing Library ecosystem, this library provides custom Jest matchers for making assertions in your tests.

3. @testing-library/react (Version 13.4.0): Part of the Testing Library ecosystem, this library provides utilities for testing React components.

4. @testing-library/user-event (Version 13.5.0): Another part of the Testing Library ecosystem, this library provides utilities for simulating user events like clicks and keyboard input in your tests.

5. axios (Version 1.4.0): A popular JavaScript library for making HTTP requests from the client-side. It simplifies the process of interacting with APIs.

6. bcrypt (Version 5.1.1): A library for hashing passwords and verifying hashed passwords. It's commonly used for user authentication and password security.

7. bcryptjs (Version 2.4.3): A JavaScript implementation of the bcrypt hashing algorithm, primarily used for password hashing and validation.

8. concurrently (Version 8.2.0): A utility for running multiple commands concurrently in a single terminal window. This is often used in development scripts to run both server and client-side code at the same time.

9. cors (Version 2.8.5): A Node.js package for handling Cross-Origin Resource Sharing (CORS). It's used to configure which domains are allowed to access resources on your server.

10. express (Version 4.18.2): A popular Node.js web application framework used for building server-side applications and APIs.

11. express-validator (Version 7.0.1): A set of Express.js middlewares for data validation. It's often used for validating user input on the server.

12. mongoose (Version 7.4.1): A Node.js library for MongoDB, which simplifies working with MongoDB databases and provides an object data modeling (ODM) layer for Node.js.

13. nodemailer (Version 6.9.4): A module for sending email from Node.js applications. It's commonly used for email notifications and communication.

14. nodemon (Version 3.0.1): A utility that monitors changes in your Node.js application and automatically restarts the server when changes are detected, making it easier for development.

15. react (Version 18.2.0): The core library for building user interfaces in React applications.

16. react-dom (Version 18.2.0): The package that provides the integration between React and the Document Object Model (DOM). It's used for rendering React components in web applications.

17. react-router-dom (Version 6.16.0): A popular routing library for React applications, used for managing navigation and rendering different components based on the URL.

18. react-scripts (Version 5.0.1): A set of scripts and configuration used for managing and building React applications. It includes scripts for development, building, and testing.

19. twilio (Version 4.14.0): The Twilio Node.js module for working with the Twilio API. It allows you to send SMS messages, make phone calls, and perform other communication-related tasks using Twilio's services.

20. web-vitals (Version 2.1.4): A library for measuring web performance, specifically focusing on key user-centric performance metrics.