CONSULTATION INSTRUCTIONS CAPTURE:CARE COMPANION

Sai Ram Varma Budharaju
CIS6930
Gainesville, FL, USA
sbudharaju@ufl.edu

Sai Praneeth Kurmelkar
CIS6930
Gainesville, FL, USA
kurmelkars@ufl.edu

***Abstract*—Developed a health app with distinct doctor/patient logins, secure backend authentication, and data management. Features dynamic UIs for signup, login, patient profiles, and appointments, integrated with backend APIs for seamless operation.**

***Index Terms*—Mobile Health Application, Patient-Doctor Interaction, Electronic Health Records (EHR), User Authentication in Healthcare, Digital Prescription Management**

I. Introduction

This section must introduce the work. For that, it must contain one paragraph (not exactly but this can be used as a suggestion) for each of the following parts:

- 1. Contextualization:In recent years, the digitization of healthcare has become paramount, driven by a need for improved accessibility, efficiency, and patient engagement. Our world, increasingly reliant on digital solutions, faces unique challenges in healthcare management, particularly in remote patient-doctor interactions and data handling.
- 2. Problem: The primary issue addressed by this project is the lack of an integrated, user-friendly platform that facilitates remote communication between patients and healthcare providers, while securely managing health records and appointments. Current systems often exist in silos, hindering seamless data exchange and patient engagement.
- 3. Consequences (impacts): The absence of such a system leads to disjointed patient care, inefficiencies in healthcare delivery, and potential compromises in data security. Patients may struggle to access timely medical advice, and healthcare providers may find it challenging to track and manage patient information effectively, impacting the overall quality of care.
- 3.5. Related work: Existing solutions, such as standalone telemedicine apps and EHR systems, offer partial remedies but often lack comprehensive integration. These solutions may focus on specific aspects like appointment scheduling or record-keeping, without a holistic approach to patient-doctor interaction.
- 4. Proposed solution: This project introduces a mobile health application that uniquely combines patient and doctor portals within a single platform. It offers features like secure login for different user roles, real-time communication, appointment scheduling, and access to electronic health records. The application emphasizes user experience and data security.
- 5. Benefits: Compared to existing solutions, this application provides a more cohesive experience, ensuring ease of use, data integrity, and enhanced communication between patients and doctors. It streamlines healthcare delivery, making it more efficient and patient-centric..

The rest of the paper will delve into the detailed architecture of the mobile health web application, user interface design, data management strategies, security protocols implemented, and a comparative analysis with existing solutions. It will also include user feedback, impact assessment, and future development plans for the application.

II. Motivations

In an era where healthcare is increasingly reliant on digital platforms, there remains a significant gap in catering to the needs of specific patient groups, such as the elderly or those with memory impairments. The current digital healthcare landscape is fragmented and often fails to address the unique challenges faced by these individuals. For instance, elderly patients or those with conditions like memory loss frequently struggle to recall medical advice or instructions given during doctor visits. Caregivers, too, face difficulties in keeping track of the varied and often complex medical directives needed for proper patient care. This situation results in:

Ineffective Communication: Critical health instructions and advice are often forgotten or misunderstood, leading to poor health management. Increased Caregiver Burden: Caregivers struggle to manage and recall the plethora of medical instructions, which can lead to errors in patient care. Risk of Medical Non-Compliance: Patients are at risk of not adhering to medical advice due to forgetfulness or misunderstanding, potentially leading to adverse health outcomes.

Example: Imagine an elderly patient with mild cognitive impairment who visits their doctor for a routine check-up. The doctor prescribes new medications and provides specific health instructions. However, by the time the patient returns home, they have forgotten most of the doctor's advice. The caregiver, who wasn't present during the appointment, is unable to provide assistance, leading to confusion and potential non-compliance with the medical regimen.

The mobile health application is designed to bridge this communication gap by offering a platform where medical instructions, prescriptions, and health conditions are not only stored but also easily accessible to both patients and caregivers. The app caters specifically to those who need assistance in remembering medical advice, schedules, and prescriptions.

Rationale Behind the Approach:

Enhanced Recall and Adherence: By providing a digital repository of medical advice and instructions, the app ensures that critical information is always accessible, aiding in memory recall and adherence to medical advice.

Empowering Caregivers:Caregivers can access the patient's health data and doctor's instructions anytime, facilitating better care coordination and management.

Patient-Centric Design: The app is tailored for ease of use, considering the specific needs of elderly patients and those with memory impairments. Large text, simple navigation can be included to accommodate users with varying levels of tech proficiency.

In summary, the app aims to provide a comprehensive solution that addresses the unique challenges faced by patients and their caregivers, ensuring effective health management and improved quality of life.

III. Proposed Mobile App Solution

A. System Architecture

The "CareCompanion" app is a comprehensive system designed to facilitate an interactive and secure platform for patients and doctors. This system architecture comprises various interconnected components, each contributing to the app's overall functionality.

Frontend Architecture:

The frontend of the app is built using React, a widely-used JavaScript library ideal for creating dynamic user interfaces, particularly for single-page applications. The user interface is composed of several pages, including the Login page, Signup page, DoctorDashboard, AppointmentsList, and PatientOverview,Home page,Precautions,Instructions and a Visit Summary page each dedicated to specific functionalities within the app. To enhance consistency and efficiency across different parts of the app, reusable components such as form fields, buttons, and navigation bars are employed. React's state management is utilized for handling dynamic data, enabling user interactions and real-time updates to the user interface.

Backend Architecture:

On the backend, Node.js in conjunction with Express is used to establish a robust and scalable server environment. MongoDB, a NoSQL database, serves as the backbone for data storage, offering flexibility and high performance. The backend is structured with various API endpoints defined under the routes directory, responsible for handling operations like user authentication, patient record management, and other user interactions. In the models directory, the data schema for the database is defined to ensure data integrity and structure. Security and session control are managed through authentication, which verify user tokens

Communication and Data Flow:

The frontend and backend communicate via RESTful API endpoints, enabling efficient data exchange. When requests are sent from the client to the server, the backend processes these requests, interacts with the database, and sends back responses. These responses are then used by the frontend to update the user interface accordingly. A key aspect of this system is the use of JWT (JSON Web Tokens) for securing API endpoints, ensuring that only authenticated users can access sensitive information.

Security and Scalability:

Security is a paramount aspect of the "CareCompanion" app, addressed through best practices like encrypted passwords using bcryptjs and secure token-based user authentication. The architecture's design allows for scalability, with a clear separation of concerns between the client and server. This separation means the backend can manage increased loads or additional services without direct impact on the frontend.

B. App Design

The design of the "CareCompanion" app was driven by an in-depth analysis of the needs and challenges faced by both doctors and patients. Consultations with healthcare professionals and patients highlighted a significant gap in post-consultation communication and record-keeping. Many patients reported difficulties in recalling specific instructions from their doctors, while doctors sought a more efficient way to track patient histories and provide updates. This feedback was pivotal in shaping the app's features and interface. Our emphasis was on creating an intuitive, accessible, and responsive interface that simplifies complex medical data into understandable formats.

User Interface (UI) Design:

The app's UI was meticulously crafted to ensure ease of use and accessibility. The design choices were influenced by the following principles:

Clarity and Simplicity: The UI avoids clutter, focusing on presenting information in a clear and concise manner. This was particularly important for elderly users or those less familiar with digital interfaces.

Consistency: A uniform design language across all screens enhances the user experience and navigability.

Accessibility: Large buttons, readable text fonts, and intuitive icons ensure that the app is user-friendly for individuals with varying levels of tech-savviness.

Screen Flow and Descriptions:

Login and Signup Screens: Simplified forms for quick access. The signup screen includes an option to register as a doctor or a patient, ensuring appropriate access and functionalities for each user type.

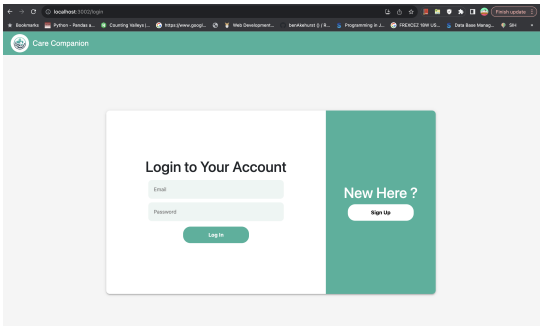

Fig. 1. Login page.

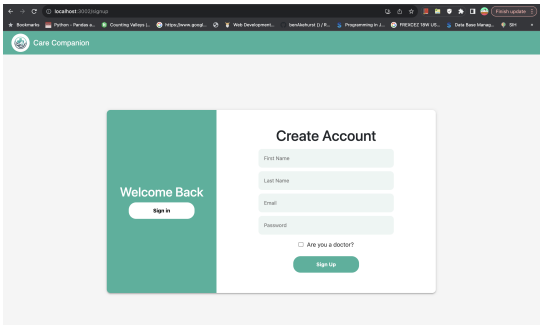

Fig. 2. Sign Up Page for new User.

Doctor Dashboard: Provides doctors with a list of diseases and an option to connect to patient through email id to enable quick access to patient records.

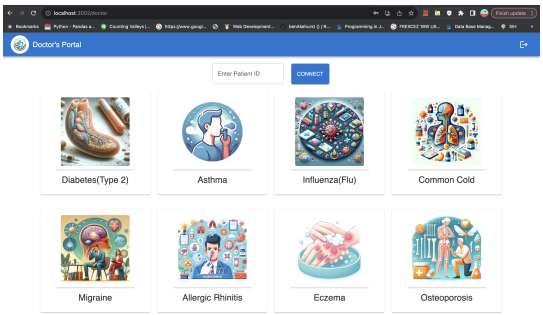

Fig. 3. Doctor Dashboard with option to connect to Patient with email.

Appointment List: A clear and organized list of appointments, when selected by patients will display doctors notes or prescriptions post-consultation.

Patient Overview: Allows patients to view their medical history, appointments, and doctor's notes along with drug prescription and additional instructions that doctors give to patient to keep the health condition in control. This section is instrumental in helping patients recall medical advice and

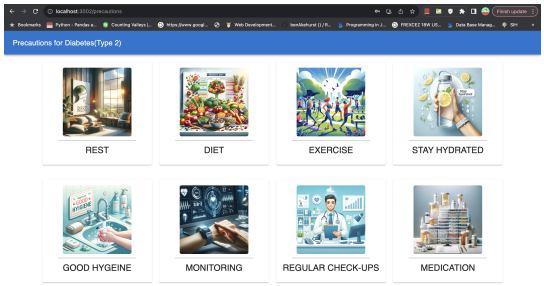

Fig. 4. Doctor screen with option to select instructions tiles.

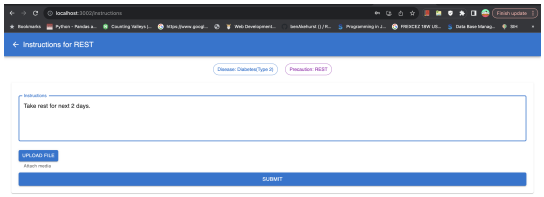

Fig. 5. Doctor screen with option to summarize the visit and giving instructions.

instructions.

The youtube link for the app demonstarting the features and design is attached below:

https://www.youtube.com/watch?v=2CVcKpHLLF0

The Github link for the app code base is attached below:

https://github.com/saipraneethkurmelkar/CareCompanion

The design decisions were backed by user feedback and best practices in UI/UX design. The color palette was chosen for its calming effect, and the layout was structured to prioritize key information. Interactive elements like buttons were designed for easy navigation, considering users with limited dexterity.

C. App Implementation

Development Platform and Framework The development of "CareCompanion" was carried out using a combination of popular and robust technologies. The client-side of the application, primarily the user interface, was developed using React, a JavaScript library renowned for its efficiency and flexibility in building interactive UIs. For the server-side, Node.js was employed, providing a scalable and powerful environment for handling backend processes. The database management was facilitated through MongoDB, known for its agility in handling large volumes of data and complex

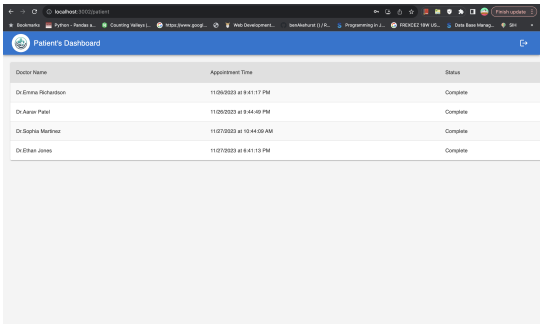

Fig. 6. Patient Dashboard showing all appointments

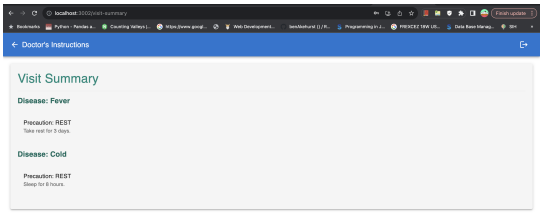

Fig. 7. Patient Dashboard with summary of the visit and instructions.

queries.

Throughout the development of the application, a range of resources played a pivotal role in guiding and enhancing the process:

React Documentation: This was a primary resource for understanding and implementing best practices within the React framework, particularly focusing on component lifecycle and state management. It offered in-depth knowledge and examples which were essential in building an interactive and dynamic user interface.

Node.js Guides: To establish a robust server-side environment, Node.js guides were referred to extensively. These resources were crucial in setting up the backend architecture, ensuring efficient server-client communication, and integrating the application with the MongoDB database.

MongoDB Tutorials: These tutorials were key to understanding MongoDB's capabilities, enabling the design and implementation of effective database schemas. They provided insights into advanced querying and data modeling, which were vital for managing the app's data storage and retrieval processes.

In addition, online forums such as Stack Overflow and React communities provided valuable insights and solutions to common development challenges.

Challenges and Resolutions:

Several challenges were encountered during development:

State Management in React: Managing states across multiple components was initially challenging. This was addressed by using React's Context API for global state management.

Asynchronous Operations: Handling asynchronous operations in Node.js, especially in database interactions, required careful structuring of promises and async/await functions.

Responsive Design: Ensuring the app was responsive across various devices required meticulous CSS styling and testing on multiple screen sizes.

Security Considerations: Implementing secure authentication and data handling processes in both frontend and backend was crucial. JWT (JSON Web Tokens) were used for secure user authentication.

IV. Discussion

Problem Context:

Our application was designed to address a critical gap in healthcare communication, specifically targeting the challenges faced by patients and caregivers in recalling medical advice and instructions given by doctors. This issue is particularly pronounced in cases involving elderly patients or those with conditions affecting memory and comprehension.

Existing Solutions:

Current solutions in the market primarily focus on appointment scheduling and medical record management, with less emphasis on post-consultation communication. While these tools are beneficial, they often overlook the essential aspect of reinforcing doctor-patient communication after the appointment.

Our Contribution:

Our application bridges this gap by providing a platform where doctors can record key instructions and medical advice, which patients or caregivers can later access at their convenience. This approach ensures that crucial medical guidance is not lost or misunderstood, enhancing patient care and compliance with medical advice.

Future Work:

Looking ahead, the application could be further developed to include features like:

Voice-to-Text Transcription: Integrating voice recognition technology to automatically transcribe doctor's instructions would streamline the process of updating patient records.

Push Notifications: Reminders for medication and follow-up appointments could be sent to patients, improving treatment adherence.

Integration with Wearable Devices: Syncing with devices that monitor vital signs could provide real-time health data to both patients and doctors, allowing for more proactive healthcare management.

Telehealth Features: Incorporating video consultation capabilities would make the app more versatile, especially in providing care to remote or mobility-impaired patients.

Multilingual Support: To cater to a diverse patient base, adding multilingual support would make the app more accessible and user-friendly.

Feedback Mechanism: Implementing a system for patients to provide feedback on their health progress and treatment

experience would be valuable for continuous care improvement.

By building upon these ideas, the application can evolve into a more comprehensive healthcare platform, addressing a wider range of patient needs and further simplifying the healthcare experience.

V. Distribution of Work

Contribution Area	Sai Ram Varma	Sai Praneeth
Frontend Development	1	1
Backend Development	1	1
Requirement Analysis	1	1
Presentation	1	1
Report Writing	1	1

TABLE I
Contribution of Team Members

VI. Related Work

In the realm of mobile health applications, several notable solutions have emerged, each with unique features catering to different aspects of healthcare management:

Athenahealth: Athenahealth offers a telehealth solution that seamlessly integrates with existing healthcare systems. It focuses on streamlining the healthcare process, from scheduling appointments to conducting follow-ups. This integration enhances patient engagement and operational efficiency for healthcare providers. [1]

DoctorConnect: This patient engagement system is designed to complement and integrate with current Electronic Health Record (EHR) and Practice Management Systems (PMS). It optimizes the entire patient journey, starting from appointment scheduling to post-visit follow-ups, thereby enhancing the overall patient experience. [2]

UVO Health: UVO Health focuses on virtual post-procedure check-ins, device instructions, and home visits. It aims to improve patient compliance and education while reducing hospital readmissions through virtual check-ups. This approach not only increases the efficiency of healthcare delivery but also enables healthcare providers to see a greater number of patients conveniently and quickly. [3]

Each of these applications contributes significantly to the evolving landscape of mobile health, offering innovative solutions to streamline healthcare processes and enhance patient care.

VII. Conclusions and Future Work

Concluding the project, we've successfully developed a comprehensive digital health application that addresses the critical need for efficient patient-doctor communication and health management. The app's dual-interface for doctors and patients enhances the usability and ensures tailored experiences for each user type. While the current version provides robust functionality, including patient profile management, condition tracking, and secure access controls, there are several avenues for future enhancement:

Integration with Wearable Devices: Future iterations could include syncing with wearable health devices to automatically update health metrics in real-time, providing a more comprehensive health overview.

Telemedicine Features: Incorporating video conferencing capabilities would allow virtual consultations, making the app more relevant in scenarios where in-person visits are not feasible.

Advanced Analytics and AI: Implementing AI-driven analytics could offer predictive insights into patient health trends and potential risks, aiding in preventative care.

User Feedback Incorporation: Continuous improvement based on user feedback will be vital. Regular updates addressing user concerns and suggestions can significantly enhance the app's effectiveness and user satisfaction.

Multilingual Support: To cater to a diverse user base, adding multilingual support will make the app more accessible and user-friendly for non-English speakers.

Expansion of Medical Conditions and Treatments: Continuously updating the database with a wider range of medical conditions and treatment options will make the app more comprehensive.

These suggestions aim to build upon the current foundation, steering the app towards a more integrated, user-centric, and technologically advanced future, ultimately contributing to better health outcomes and enhanced patient care.

References

[1] "Athenahealth telehealth," https://www.athenahealth.com.

[2] "Doctorconnect patient engagement system," https://www.doctorconnect.net.

[3] "Uvo health," https://www.uvohealth.com.