



Project Report Blockchain

Call patient data system

Nattamet Wongtanitlert	6404062610472
Tanawat Kama	6404062610481
Putthakun Suansamran	6404062610171
Badinchai Palichaiwat	6404062610103

THIS PROJECT IS PART OF THE STUDY OF THE BACHELOR OF SCIENCE IN

COMPUTER SCIENCE

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

FACULTY OF APPLIED SCIENCE

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

THE ACADEMIC YEAR 2023

Introduction

The overlapping of treatment data in hospitals is a common issue, especially in multi-facility healthcare. Often, data overlap occurs because some places still rely on paper-based records. Therefore, we have introduced blockchain technology to help reduce this data overlap and facilitate easier communication between doctors and patients. This makes disease diagnosis easier, reduces redundant payments, and lowers the risk of patients receiving medications they are allergic to. But doctors are unaware of this because some patients cannot remember the medications they have taken and which ones they are allergic to.

2/4/2024

Table of contents

INTRODUCTION	2
CHAPTER 1	4
CUSTOMER JOURNEY	4
CHAPTER 2	7
PAIN POINT	7
CHAPTER 3	9
DESIGN WIREFRAME.....	9
CHAPTER 4	13
SYSTEM ARCHITECTURE	13
CHAPTER 5	15
DATA STRUCTURE	15
CHAPTER 6	18
SERVICE AND TOOLS.....	18
CHAPTER 7	20
TIMELINE AND MANPOWER	20
CHAPTER 8	21
WEB FUNCTIONALITY.....	21

Chapter 1

Customer journey

1.1 Customer journey



- **Outpatient Customer Journey:**
 - Receiving a Queue Card:
 - Description of the process of receiving a queue card upon arrival at the outpatient facility.
 - Importance of efficient queuing systems in managing patient flow.
 - Filling in Patient History:
 - Discussion of the patient's interaction with administrative staff to complete necessary paperwork and provide medical history.
 - Importance of an accurate and comprehensive patient history for effective diagnosis and treatment.

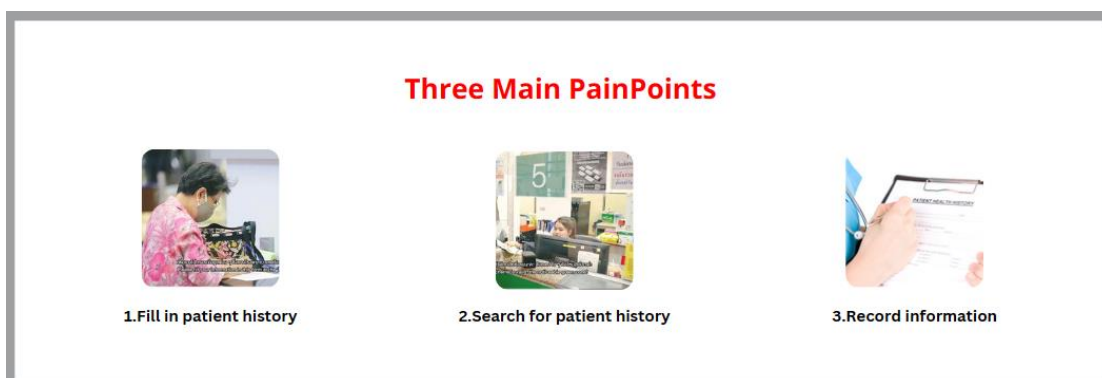
- Screening Nurse:
 - Overview of the role of the screening nurse in assessing patient's vital signs, chief complaints, and preliminary assessments.
 - The importance of initial screening in triaging patients and identifying urgent cases.
- Seeing a doctor:
 - An explanation of the process of meeting with the doctor for consultation and examination.
 - Importance of timely access to healthcare providers and effective communication during the visit.
- Treatment or medication:
 - Description of the final steps in the outpatient journey, which may include receiving treatment, prescriptions, or referrals.
 - Importance of clear instructions and follow-up plans for patient adherence and recovery.
- **Inpatient Customer Journey:**
 - Searching for Patient History:
 - Explanation of how healthcare providers access and review patient histories upon admission.
 - The importance of comprehensive electronic health records (EHR) systems in facilitating efficient care delivery.
 - Screening Nurse:
 - Like the outpatient journey, discuss the role of the screening nurse in the inpatient setting.
 - Seeing a doctor:
 - Overview of the process of consulting with the attending physician or specialist for inpatient care.
 - Importance of multidisciplinary care teams in coordinating treatment plans and addressing complex medical needs.

- Treatment or medication:
 - Description of the treatment process for inpatients, which may involve procedures, surgeries, or medication administration.
 - Importance of patient safety protocols and monitoring during hospitalization.

Chapter 2

Pain point

Identification of Pain Points



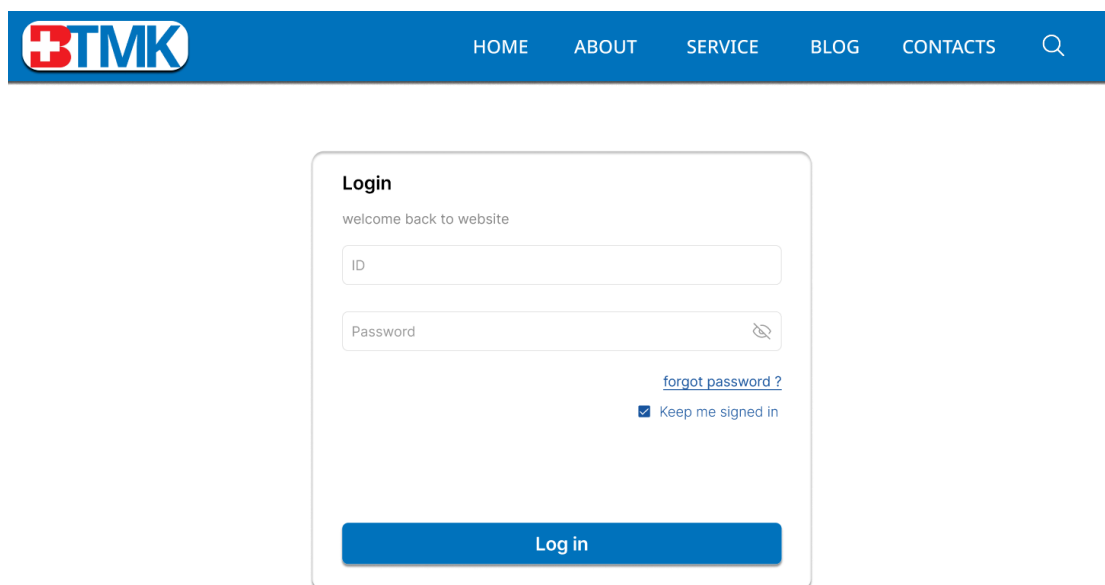
- **Difficulty in Accessing Comprehensive Patient History:**
 - One of the primary challenges encountered in healthcare facilities is the process of collecting and accessing a complete patient history. Patients are often required to fill out extensive forms with their medical history upon each visit. However, this process can be time-consuming and cumbersome for both patients and healthcare providers.
- **Inefficiency in Retrieving Patient Records:**
 - Another significant pain point arises from the inefficiency in retrieving patient records, particularly in hospital settings. Healthcare professionals often struggle to locate and access relevant patient information efficiently, leading to delays in treatment and potential compromises in patient care.
- **Risk of Errors Due to Outdated or Paper-Based Records:**
 - The reliance on outdated or paper-based medical records exacerbates the risk of errors in healthcare delivery. Manual record-keeping systems are prone to inaccuracies, omissions, and illegible handwriting, which can compromise patient safety and quality of care.

- Challenges in Recalling Past Medical Information:
 - Patients may face challenges in recalling their past medications, treatments, or allergic reactions accurately. This lack of accurate information can impede healthcare providers' ability to make informed decisions and tailor treatment plans effectively.

Chapter 3

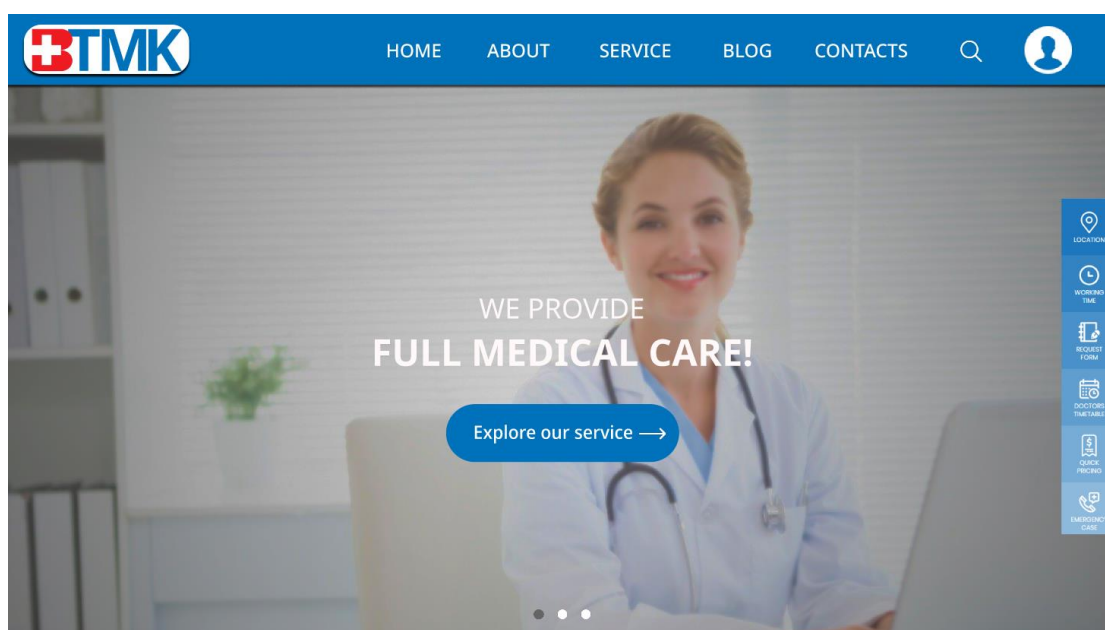
Design wireframe

1.Login page




The login page features a blue header with the BTMK logo on the left and navigation links (HOME, ABOUT, SERVICE, BLOG, CONTACTS) and a search icon on the right. The main content area contains a white login box with a blue border. Inside the box, the heading "Login" is followed by the text "welcome back to website". Below this are two input fields: "ID" and "Password" (with a toggle icon). A link for "forgot password ?" and a checkbox for "Keep me signed in" are positioned below the password field. A large blue "Log in" button is at the bottom of the box.



2.Home page



3. User create record



HOMEABOUTSERVICEBLOGCONTACTS



Create New Record

ID

Full Name

Phone

Gender

Date of Birth

Height

CM

Weight

KG

House Address


Emergency Contact

Emergency Contact Name



Emergency Contact Phone


Confirm

4. Profile of Patient



HOMEABOUTSERVICEBLOGCONTACTS





Moddy Lovehee

ID

1104300879022

Phone

099-888-7788

Gender

Male

PERSONAL DETAILS

Full Name

Moddy Lovehee

Birthdate

01/10/1999

Height

180 CM

Weight

80 KG

Address

1234, Bangkok 11Wongsawang 19100 Kmutnb

Emergency Contact

Full Name

Nong BigCInwza

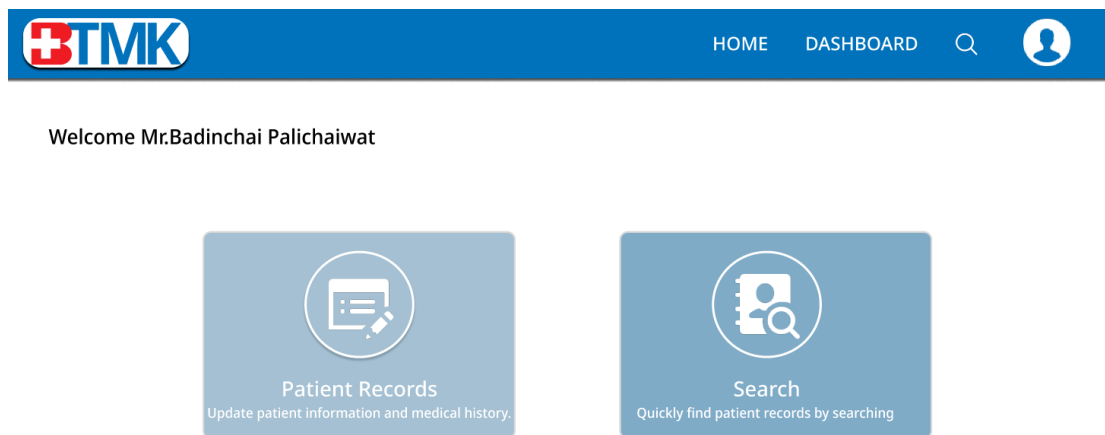
Phone

099-888-7799



Cancel

Edit


6. Doctor main page




TMK

HOME DASHBOARD  

Welcome Mr.Badinchai Palichaiwat

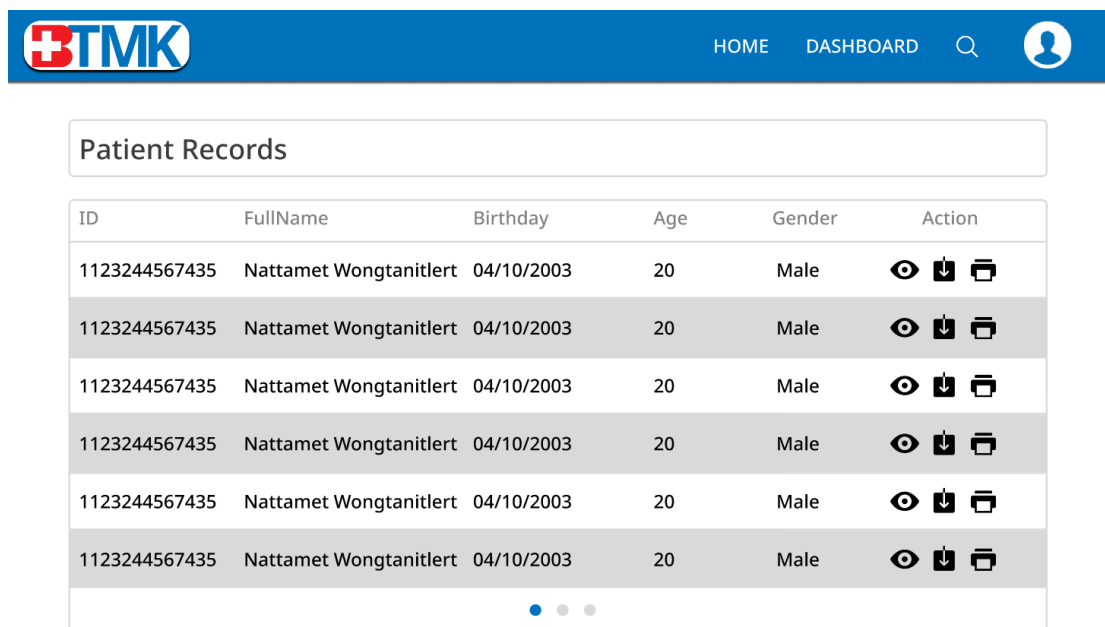


Patient Records
Update patient information and medical history.





Search
Quickly find patient records by searching



















6. Patient records page






TMK




HOME DASHBOARD  


Patient Records

ID	FullName	Birthday	Age	Gender	Action
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  
1123244567435	Nattamet Wongtanitlert	04/10/2003	20	Male	  




  

6. Doctor search page


HOME DASHBOARD  



5. Doctor edit patient information

HOME DASHBOARD  

Edit information



Moddy Lovehee

ID 1104300879022

Phone 099-888-7788

Gender Male

PERSONAL DETAILS

Full Name	Birthdate	Height	Weight
Moddy Lovehee	01/10/1999	180 CM	80 KG

Address

1234, Bangkok 11Wongsawang 19100 Kmutnb

Emergency Contact

Full Name	Phone
Nong BigCInwza	099-888-7799

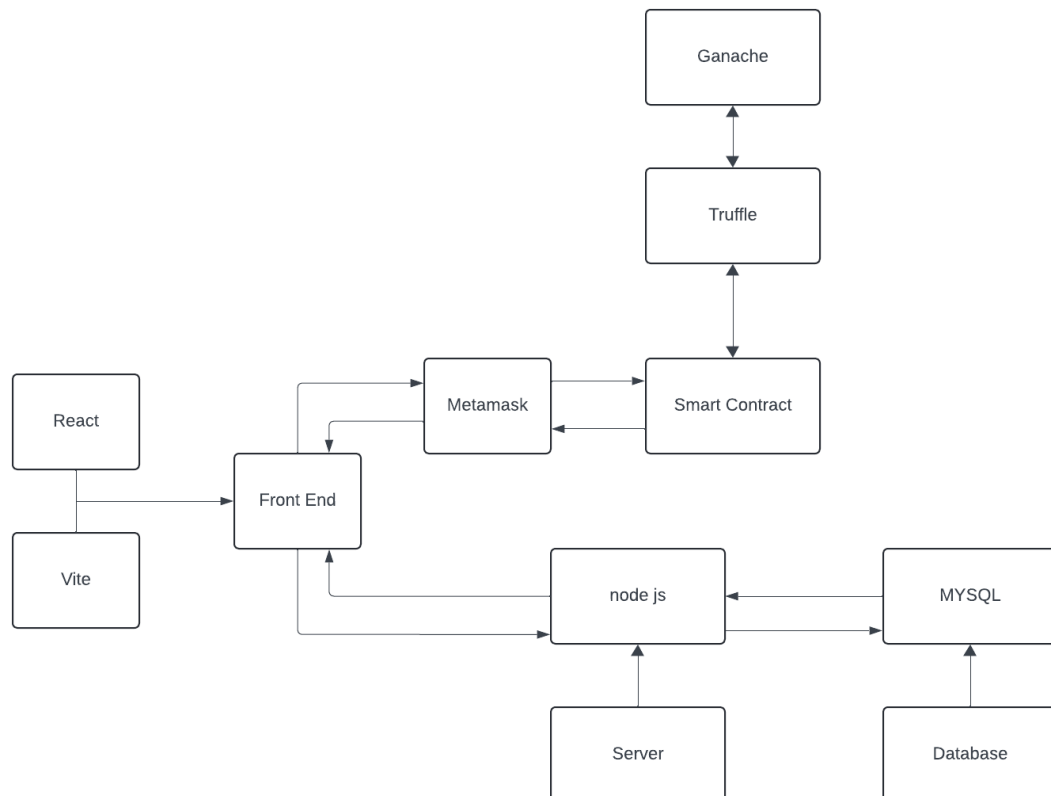
Cancel

Confirm

Chapter 4

System architecture

1.System architecture diagram



System Workflow

- Frontend Development:
 - Developers use React with Vite for frontend development, creating user interfaces and client-side logic for interacting with the DApp.
- Smart Contract Development:
 - Smart contracts are developed using Truffle's suite of tools, including Truffle CLI for project management, compiling contracts, and deploying them to the Ganache local blockchain network.

- Backend Development:
 - Backend logic is implemented using Node.js, which handles API requests from the frontend and interacts with the MySQL database for data storage and retrieval.

- Database Management:
 - MySQL databases are managed using XAMPP, providing a local environment for development and testing of database-related functionality.

Chapter 5

Data structure

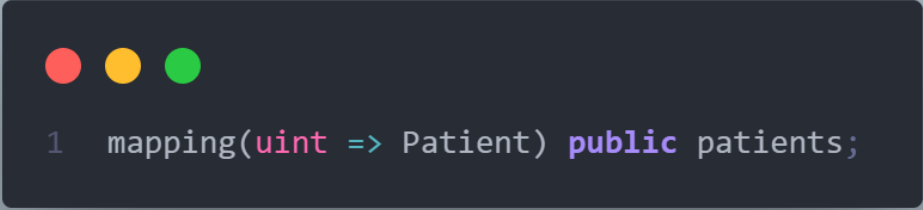
1.Data Structure

A code editor window with a dark background and light-colored text. The code is written in Solidity and defines a contract named PatientRecord. Inside the contract, there is a struct named Patient with several fields: uint256 id, string name, string gender, uint8 age, string bloodType, string phoneNumber, string drugAllergy, and string congenitalDisease. The code is numbered from 1 to 13.

```
1  pragma solidity >=0.4.22 <0.9.0;
2
3  contract PatientRecord {
4      struct Patient {
5          uint256 id;
6          string name;
7          string gender;
8          uint8 age;
9          string bloodType;
10         string phoneNumber;
11         string drugAllergy;
12         string congenitalDisease;
13     }
```

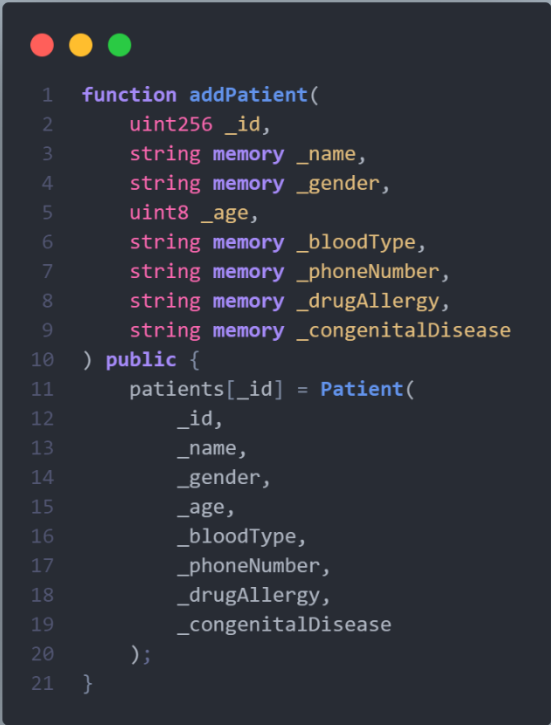
This Patient struct represents the attributes or properties of a patient record. Here's a brief explanation of each field:

- id: An unsigned integer representing a unique identifier for each patient.
- name: A string representing the name of the patient.
- gender: A string representing the gender of the patient.
- age: An unsigned 8-bit integer representing the age of the patient.
- bloodType: A string representing the blood type of the patient.
- phoneNumber: A string representing the phone number of the patient.
- drugAllergy: A string representing any allergies to drugs that the patient may have.
- congenitalDisease: A string representing any congenital diseases or conditions the patient may have.



```
1 mapping(uint => Patient) public patients;
```

Mapping: The patients mapping is declared with uint keys (representing patient IDs) mapped to Patient structs. This mapping allows for efficient storage and retrieval of patient records based on their IDs.



```
1 function addPatient(  
2     uint256 _id,  
3     string memory _name,  
4     string memory _gender,  
5     uint8 _age,  
6     string memory _bloodType,  
7     string memory _phoneNumber,  
8     string memory _drugAllergy,  
9     string memory _congenitalDisease  
10 ) public {  
11     patients[_id] = Patient(  
12         _id,  
13         _name,  
14         _gender,  
15         _age,  
16         _bloodType,  
17         _phoneNumber,  
18         _drugAllergy,  
19         _congenitalDisease  
20     );  
21 }
```

Adding Patient Records: The add Patient function is used to add new patient records to the contract. It takes parameters corresponding to the fields of the Patient struct and initializes a new Patient struct with those values, which is then stored in the patients mapping.


```
1  function getPatient(uint256 _id)
2      public
3      view
4      returns (
5          uint256,
6          string memory,
7          string memory,
8          uint8,
9          string memory,
10         string memory,
11         string memory,
12         string memory
13     )
14 {
15     Patient memory patient = patients[_id];
16     return (
17         patient.id,
18         patient.name,
19         patient.gender,
20         patient.age,
21         patient.bloodType,
22         patient.phoneNumber,
23         patient.drugAllergy,
24         patient.congenitalDisease
25     );
26 }
```

Retrieving Patient Records: The `getPatient` function allows retrieving patient records based on their IDs. It takes an ID as input and returns the corresponding patient's information by fetching the `Patient` struct from the `patients` mapping and returning its individual fields.

Chapter 6

Service and Tools

- Front End (React with Vite):
 - React is a popular JavaScript library for building user interfaces, offering a component-based architecture and efficient rendering. Vite is a build tool that provides fast development server and optimized production builds. Together, they form the front-end layer responsible for user interaction and presentation.
- Blockchain Development Tools:
 - Ganache and Truffle are commonly used tools in Ethereum blockchain development.
 - Ganache provides a local blockchain environment for development and testing, enabling developers to deploy contracts, mine blocks, and inspect transactions locally.
 - Truffle is a development framework that offers tools for compiling, deploying, and testing smart contracts. It simplifies the development process by providing project scaffolding and automated testing utilities.
- Smart Contracts:
 - Smart contracts are self-executing contracts with predefined rules and conditions written in blockchain-specific languages such as Solidity. They run on the

blockchain and automatically enforce the terms of the agreement when triggered by specific conditions.

- Backend (Node.js with MySQL):
 - Node.js is a JavaScript runtime environment that allows developers to run JavaScript on the server-side. It's used here to build the backend logic and handle requests from the frontend.
 - MySQL is a relational database management system used for storing and managing data. It provides structured storage for user accounts, transaction records, and other relevant information.
- Development Environment (XAMPP):
 - XAMPP is a cross-platform web server solution stack that includes Apache, MariaDB (a MySQL database server), and PHP. It's used for local development and testing of the backend components.

Chapter 7

Timeline and Manpower

Timetable

Months/Weeks	1	2	3	4
January	-	brainstorm	Find pain point	Select Topic
February	Find Customer journey	Brainstorm	UX/UI design	UX/UI design
March	Create System architecture	Coding smart contract	Coding smart contract/backend	Create user interface/backend
April	Test system and function/write document	Create a slide		

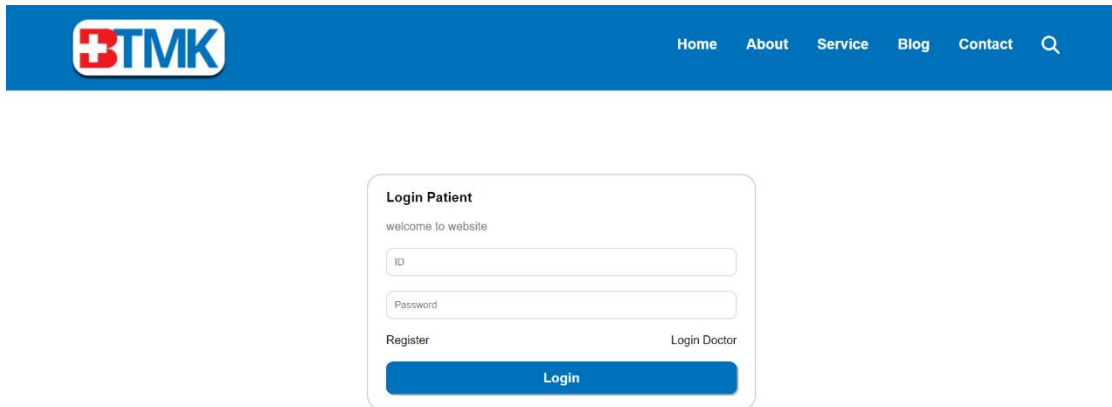
Manpower

- 1.Tanawat Kama (Main developer (Smart contract)-Frontend developer)
- 2.Putthakun Suansamrarn (developer (Smart contract)-Main Frontend developer)
- 3.Natthamet Wongtanitlert (main UX/UI design-developer-documents-slide presentation)
- 4.Badinchai Palichaiwat (sub UX/UI design-developer-documents-slide presentation)

Chapter 8

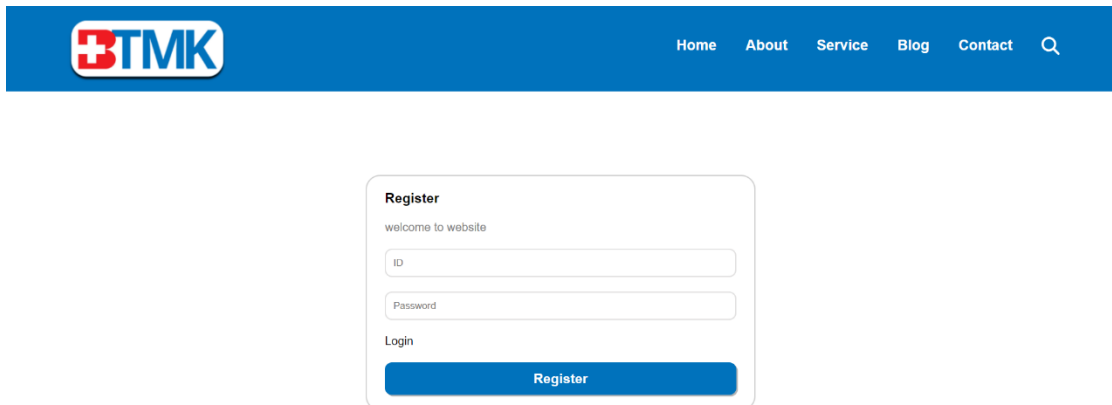
Web Functionality

Login Patient Page



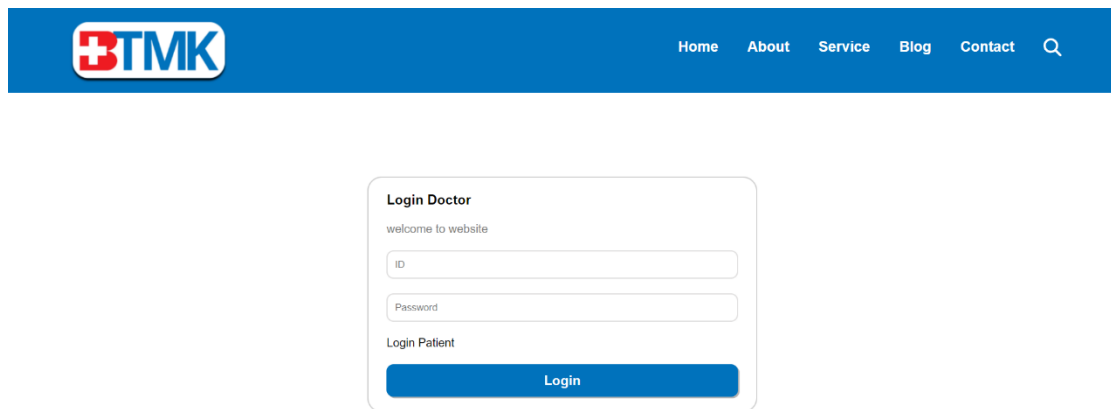
The screenshot shows the 'Login Patient' page of a website. At the top is a blue header with the 'BTMK' logo on the left and navigation links 'Home', 'About', 'Service', 'Blog', 'Contact' on the right, followed by a magnifying glass icon. The main content area features a white box with a light gray border. Inside this box, the title 'Login Patient' is at the top, followed by the text 'welcome to website'. Below this are two input fields: 'ID' and 'Password'. At the bottom of the box, there are two links: 'Register' and 'Login Doctor'. A large blue button labeled 'Login' is positioned at the bottom center of the white box.

Register Patient Page



The screenshot shows the 'Register' page of the same website. It has the same blue header with the 'BTMK' logo and navigation links. The main content area features a white box with a light gray border. Inside, the title 'Register' is at the top, followed by 'welcome to website'. Below this are two input fields: 'ID' and 'Password'. At the bottom left of the box is a link labeled 'Login'. A large blue button labeled 'Register' is positioned at the bottom center of the white box.

Login Doctor Page



The screenshot shows the 'Login Doctor' page of the TMK website. The header is blue with the TMK logo on the left and navigation links (Home, About, Service, Blog, Contact) and a search icon on the right. The main content area is white and contains a login form. The form has a title 'Login Doctor', a subtitle 'welcome to website', and two input fields labeled 'ID' and 'Password'. Below these fields is a link 'Login Patient' and a blue 'Login' button.

Login Doctor
welcome to website

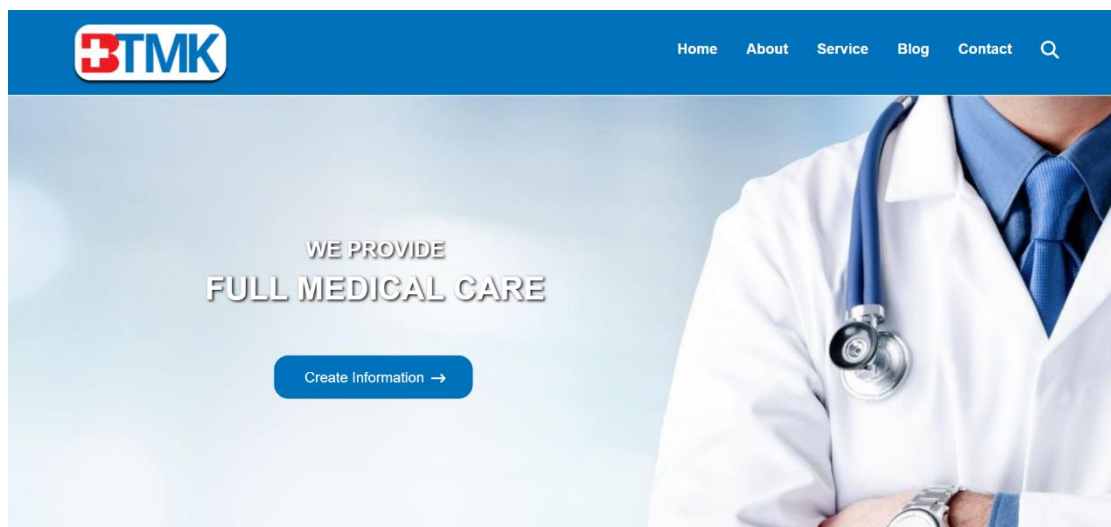
ID

Password



[Login Patient](#)

[Login](#)

Patient Main Page





Patient Create Information Page


[Home](#) [About](#) [Service](#) [Blog](#) [Contact](#) 

AddInformation

ID	Full Name	Gender
<input type="text"/>	<input type="text"/>	<input type="text" value="Female"/>
Age	Phone	Blood type
<input type="text"/>	<input type="text"/>	<input type="text"/>
DrugAllergy		
<input type="text"/>		
CongenitalDisease		
<input type="text"/>		

Patient Information Page

[Home](#) [About](#) [Service](#) [Blog](#) [Contact](#) 



OOP

ID	5555
Phone	0123456789
Gender	female

PERSONAL DETAIL

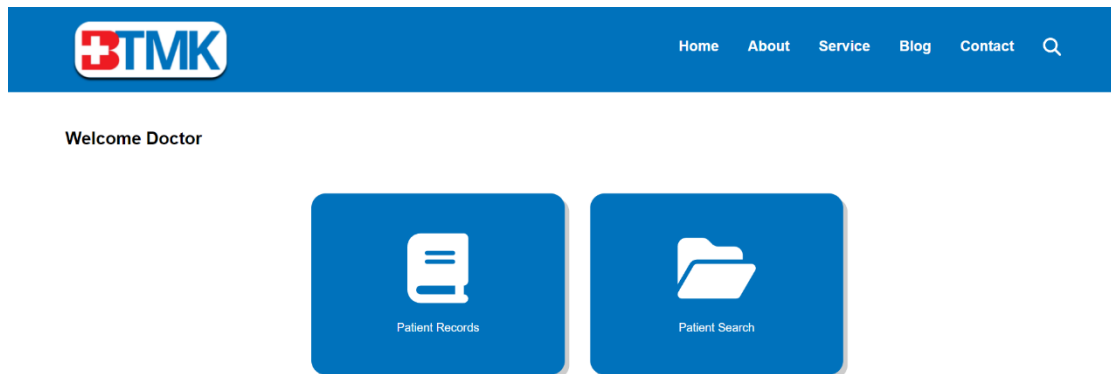
Full Name	Age	Blood Type
OOP	32	B

Emergency Contact

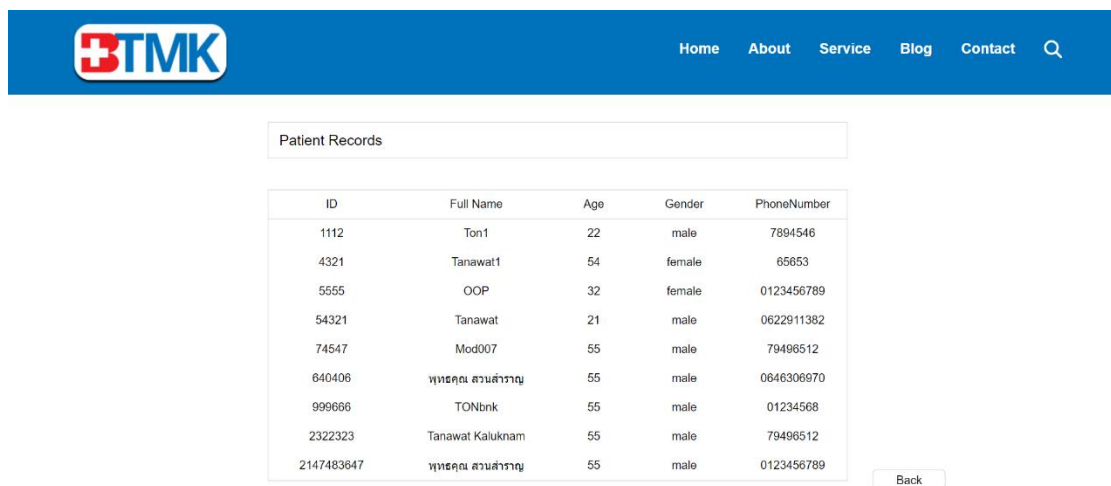
drugAllergy	congenitalDisease
Blockchain	Hard Work

Back



Doctor Main Page



Doctor Use Patient Record Function Page






Doctor Use Patient Search Function Page

[Home](#) [About](#) [Service](#) [Blog](#) [Contact](#) 

Patient's ID Card Number

Patient Data For Doctor Function Page

[Home](#) [About](#) [Service](#) [Blog](#) [Contact](#) 



Tanawat2

ID	4321
Phone	54123
Gender	female

PERSONAL DETAIL

Full Name	Age	Blood Type
Tanawat2	56	A

Emergency Contact

drugAllergy	congenitalDisease
TON1	MOD1