Narayana Health

Internship-Report

Topic : Health-Insurance Website

Project Overview

During my internship, I developed a full stack MERN (MongoDB, Express.js, React, Node.js) web application dedicated to selling health insurance plans. The project aimed to create a user-friendly platform where customers can explore, compare, and purchase health insurance plans with ease. The key features of the website include:

1. **Plan Selection:** Users can choose from three distinct health insurance plans:
   - **Basic Plan:** Offers essential coverage for routine health care needs.
   - **Standard Plan:** Provides a balanced mix of coverage and benefits.
   - **Premium Plan:** Includes extensive coverage with the highest level of benefits and wellness features.

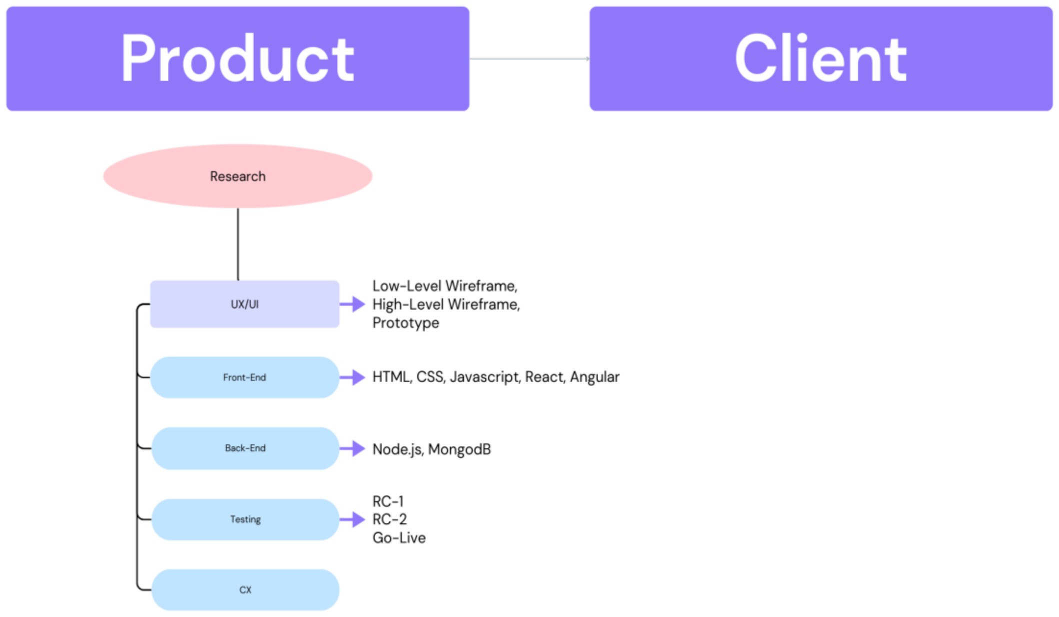
2. **User Authorization:** Implemented using the Firebase console to ensure secure access. Users receive a One-Time Password (OTP) on their mobile number for authentication, ensuring a secure and streamlined login process.

3. **User Interface:** The website boasts a clean and intuitive design, allowing users to easily navigate through the different plans, compare benefits, and make informed decisions.

Technical Stack:

- **Front-end:** Developed using React.js for a responsive and dynamic user experience.
- **Back-end:** Utilized Node.js and Express.js for server-side operations and API management.
- **Database:** Employed MongoDB to store user data, plan details, and transaction records.
- **Authentication:** Used Firebase for secure OTP-based user authentication.

This project provided hands-on experience with full stack development, enhancing my skills in building secure, efficient, and user-friendly web applications. The resulting website successfully meets the goal of simplifying the process of purchasing health insurance plans for users.

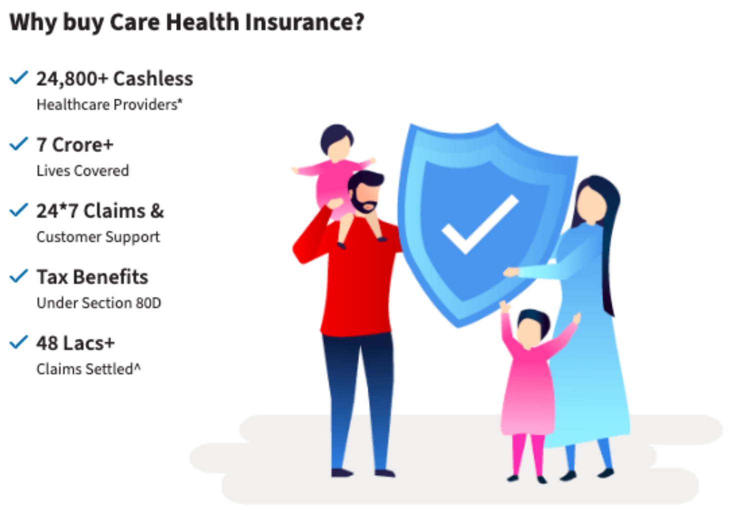


After the Product team meets with the client, the UX/UI is designed by the team according to the client requirements and the user flow diagram is determined. Research plays a vital role in the design part which is then realized in three stages-

Low Level Wireframe, High Level Wireframe, Prototype. After the design is confirmed, the Development team that consists of Front-End and Back-End Developers code the website according to the functionality needs and the design. The Website is then tested in stages and after every bug is solved, the Website is made live for the customer. The Customer Experience is taken as feedback and the website is managed accordingly.

Research

1. <u>Using data to attract Customers:</u>

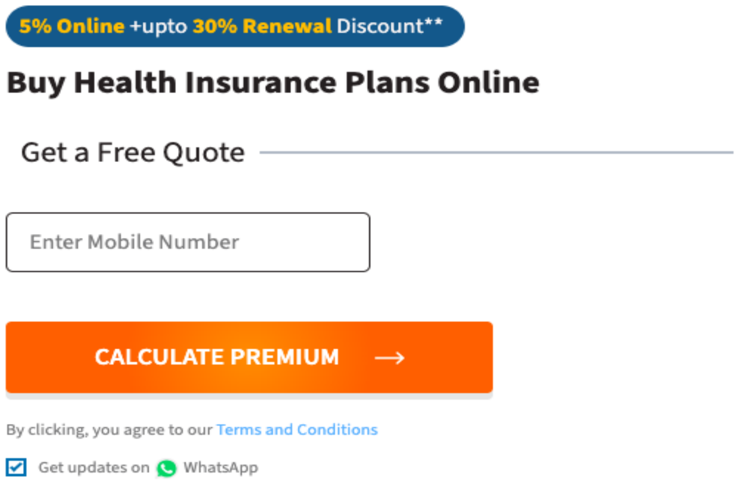


Why buy Care Health Insurance?

- ✓ 24,800+ Cashless
  Healthcare Providers*
- ✓ 7 Crore+
  Lives Covered
- ✓ 24*7 Claims &
  Customer Support
- ✓ Tax Benefits
  Under Section 80D
- ✓ 48 Lacs+
  Claims Settled^

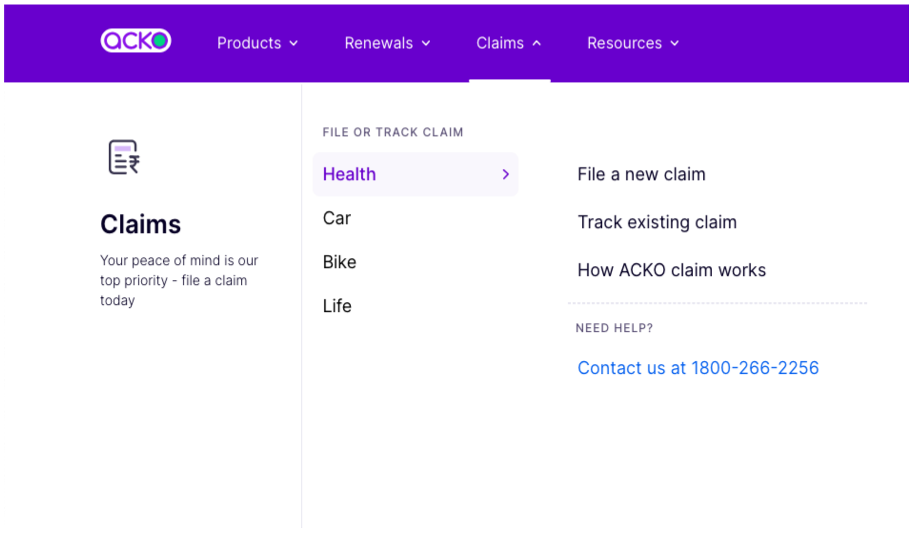
We need to make the customer satisfied with the prospect of buying the health insurance from our website. So, we'll need to back our claim with data and numbers. And, to list all the benefits makes the user experience better.

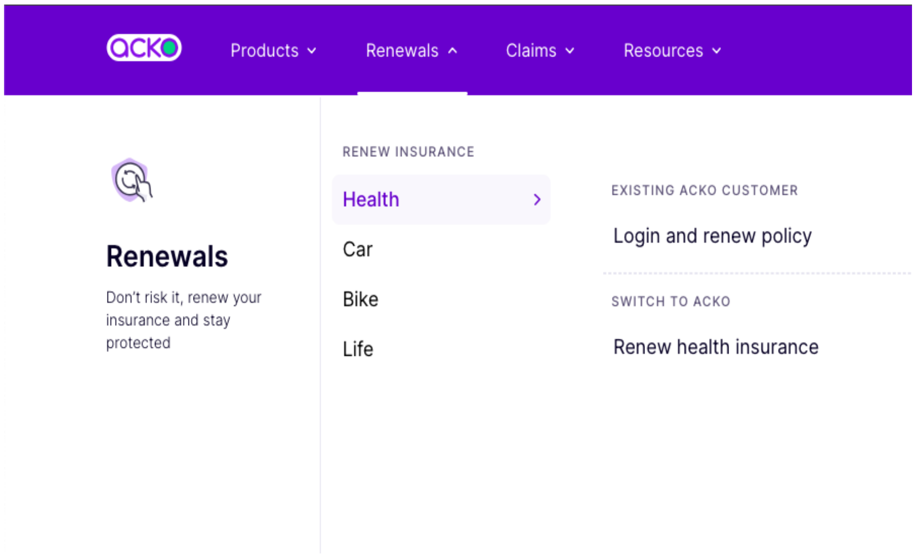
2. Free Consultation:


5% Online +upto 30% Renewal Discount**
Buy Health Insurance Plans Online
Get a Free Quote
Enter Mobile Number
CALCULATE PREMIUM
By clicking, you agree to our Terms and Conditions
Get updates on WhatsApp

In one of the websites, it was observed that a free consultation is offered. The user has to just enter the mobile number and the user then gets called by the official to facilitate user experience and to explain all the plans and benefits. Adding this section to our website could be beneficial, But the resources needed to fulfill this functionality makes things complicated. We can also provide the users with a link to a toll-free number which will direct the call to our official who can entertain the customer.

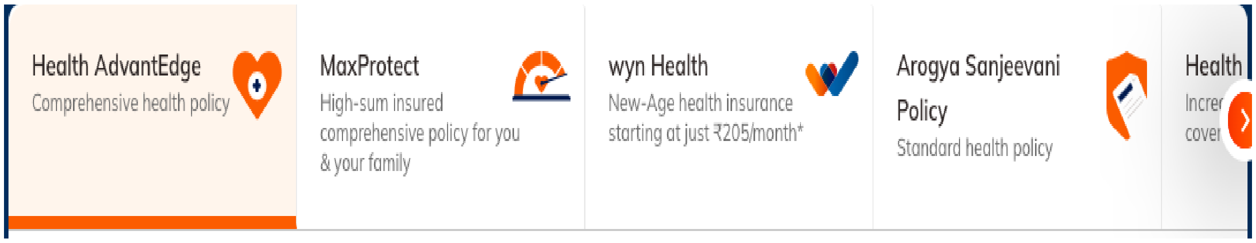
3.Visible Streamlined Services:





Any website with easy to access services is destined to bring better user experience. It is seen in the images how the Claims are easy to manage for the user as well as the renewals. These are the services that are provided by the company, which the users can avail through the website and every functionality is on the fingertips of the user and in the navbar in an arranged way.

4. <u>Plans:</u>

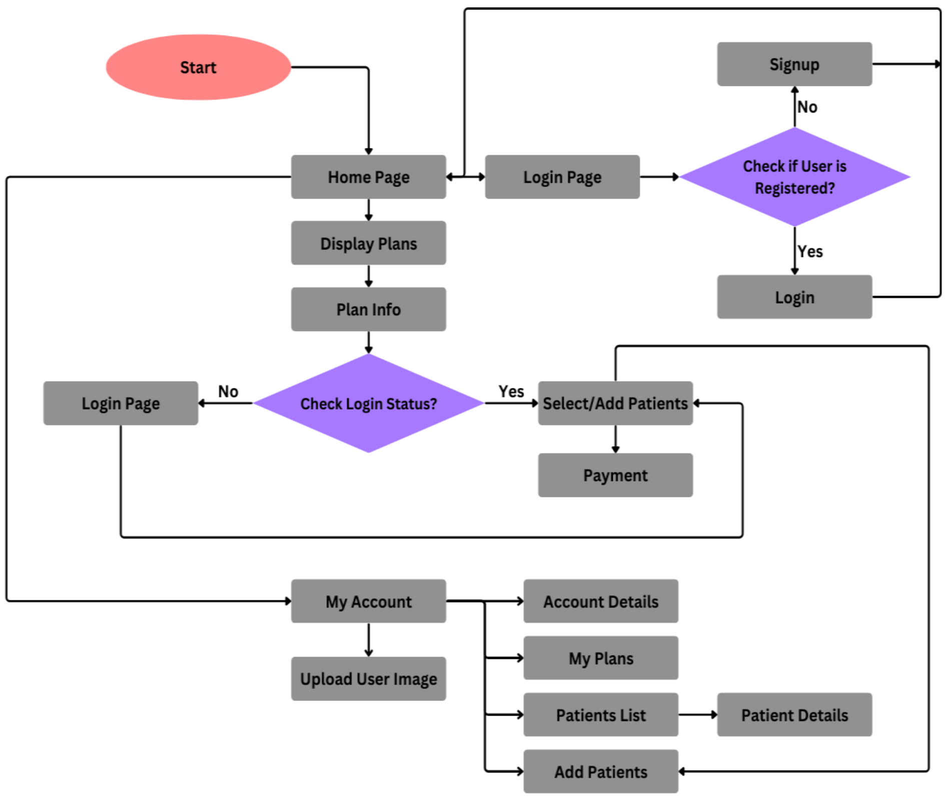

Health AdvantEdge
Comprehensive health policy
MaxProtect
High-sum insured comprehensive policy for you & your family
wyn Health
New-Age health insurance starting at just ₹205/month*
Arogya Sanjeevani Policy
Standard health policy
Health

Showing the different types of plans available in an interactive and binding manner with the description is very important regarding good user engagement. Explaining the terms of the Insurance such that the user does not remain in doubt will help in users proceeding to buy the plan. This enhances the user experience.

5. <u>Login Page:</u>

We must create a login portal in order to personalize user experience and in order to maintain our database. The login portal would contain Phone Number and OTP. If the user is not registered already, they would have to Sign up using their personal details such as Name, Phone Number, and e-mail ID.

User Flow



Database Collections

User Collection:

- _id: ObjectId

- Name: String
- PhoneNumber: String
- Email: String
- Date: Date
- Photo: String

Patient Collection:

- _id: ObjectId
- User_id: String
- Name: String
- Gender: String
- Email: String
- Dob: Date
- Address: String
- Reg_date: Date
- Show: Boolean

Plans Collection:

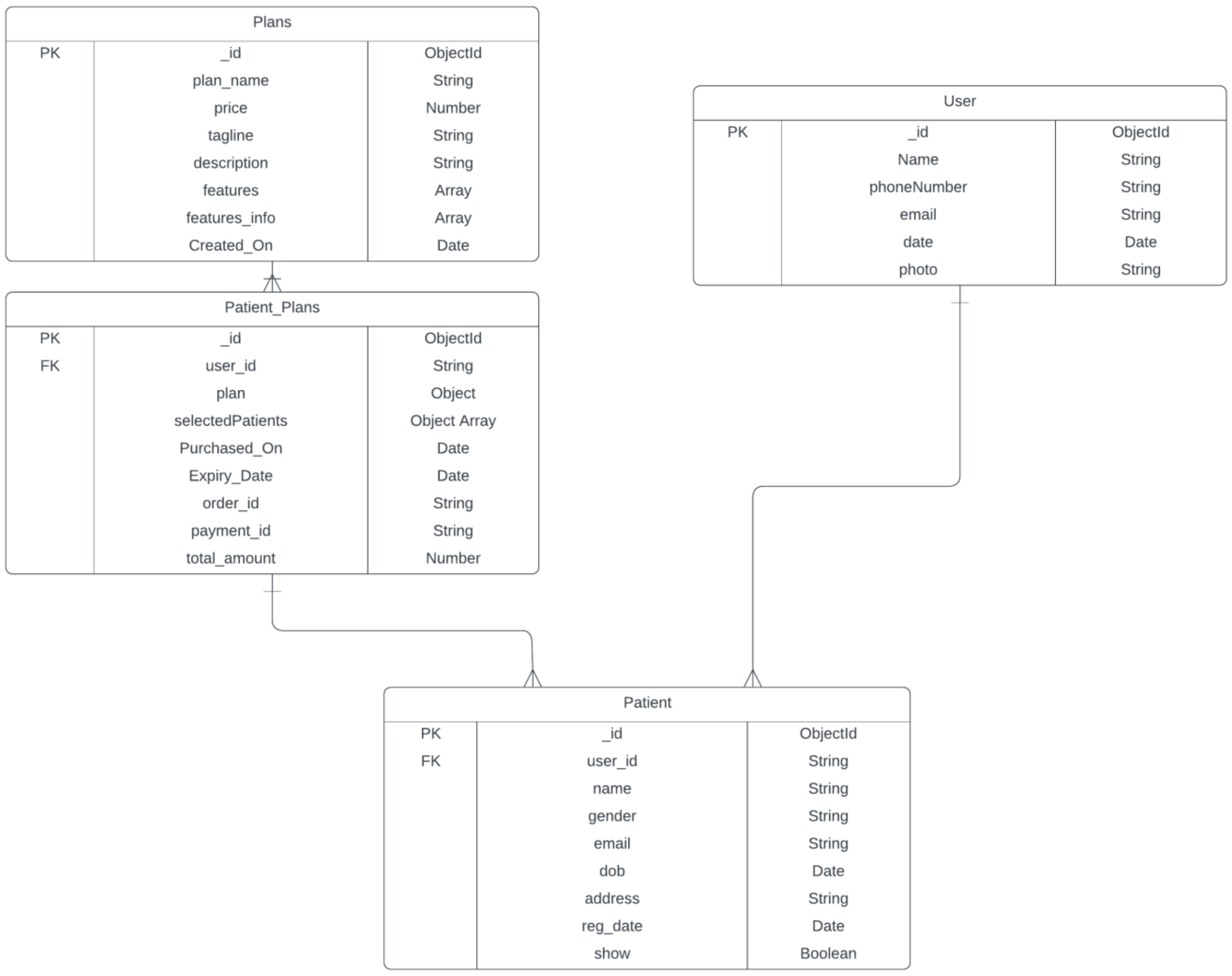
- _id: ObjectId
- Plan_name: String
- Price: Number
- Tagline: String
- Description: String
- Features: Array
- Features_info: Array
- Created_On: Date

Patient_Plans Collection:

- _id: ObjectId
- User_id: String
- Plan: Object
- SelectedPatients: Object Array
- Purchased_On: Date
- Expiry_Date: Date
- Order_id: String

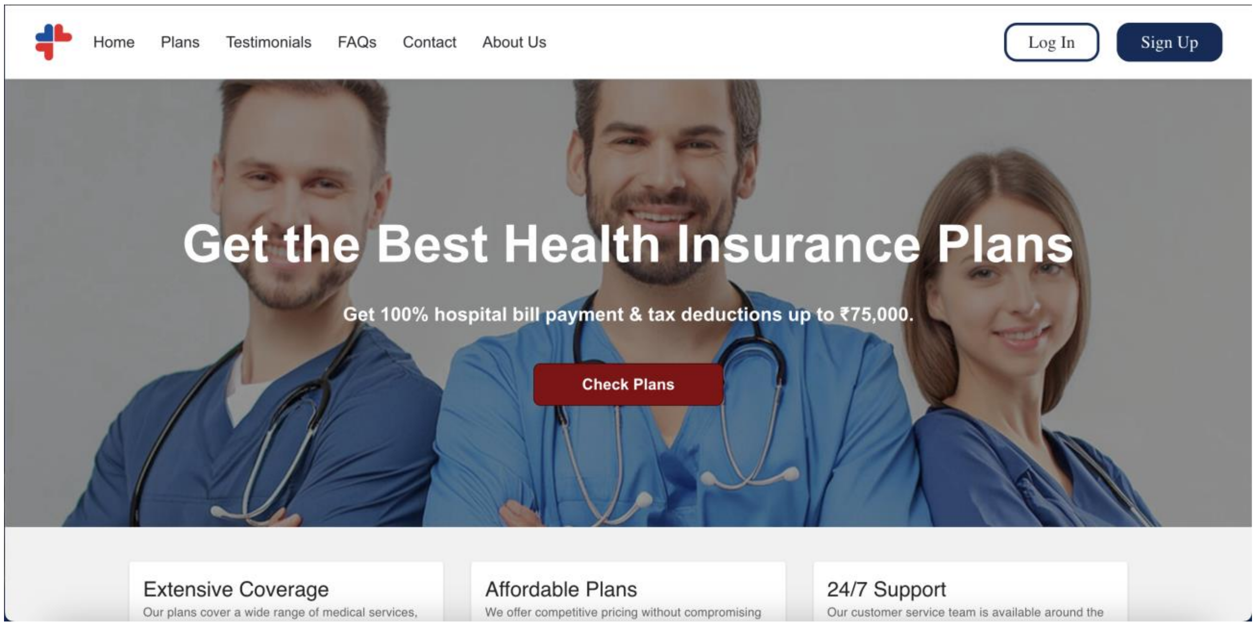
- Payment_Id: String
- Total_Amount: Number

Entity Relationship Diagram



About the Website

Home Page:



Get the Best Health Insurance Plans

Get 100% hospital bill payment & tax deductions up to ₹75,000.

Extensive Coverage

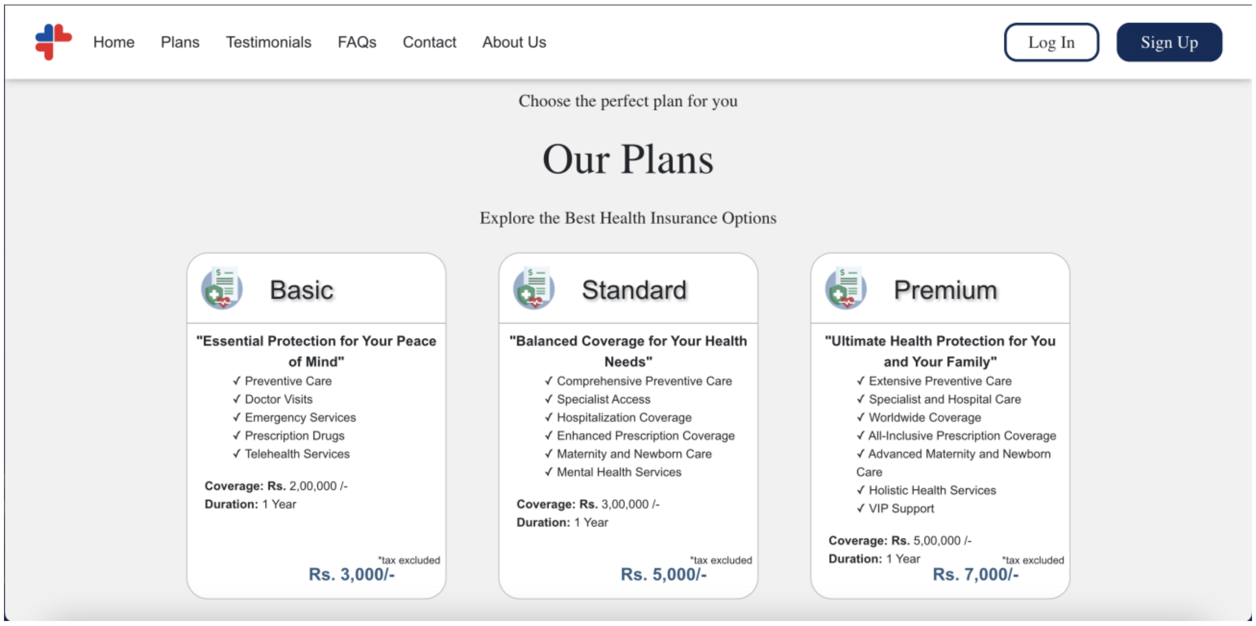
Our plans cover a wide range of medical services,

Affordable Plans

We offer competitive pricing without compromising

24/7 Support

Our customer service team is available around the



Choose the perfect plan for you

Our Plans

Explore the Best Health Insurance Options

Basic

"Essential Protection for Your Peace of Mind"

✓ Preventive Care
✓ Doctor Visits
✓ Emergency Services
✓ Prescription Drugs
✓ Telehealth Services

Coverage: Rs. 2,00,000 /-
Duration: 1 Year

*tax excluded
Rs. 3,000/-

Standard

"Balanced Coverage for Your Health Needs"

✓ Comprehensive Preventive Care
✓ Specialist Access
✓ Hospitalization Coverage
✓ Enhanced Prescription Coverage
✓ Maternity and Newborn Care
✓ Mental Health Services

Coverage: Rs. 3,00,000 /-
Duration: 1 Year

*tax excluded
Rs. 5,000/-

Premium

"Ultimate Health Protection for You and Your Family"

✓ Extensive Preventive Care
✓ Specialist and Hospital Care
✓ Worldwide Coverage
✓ All-Inclusive Prescription Coverage
✓ Advanced Maternity and Newborn Care
✓ Holistic Health Services
✓ VIP Support

Coverage: Rs. 5,00,000 /-
Duration: 1 Year

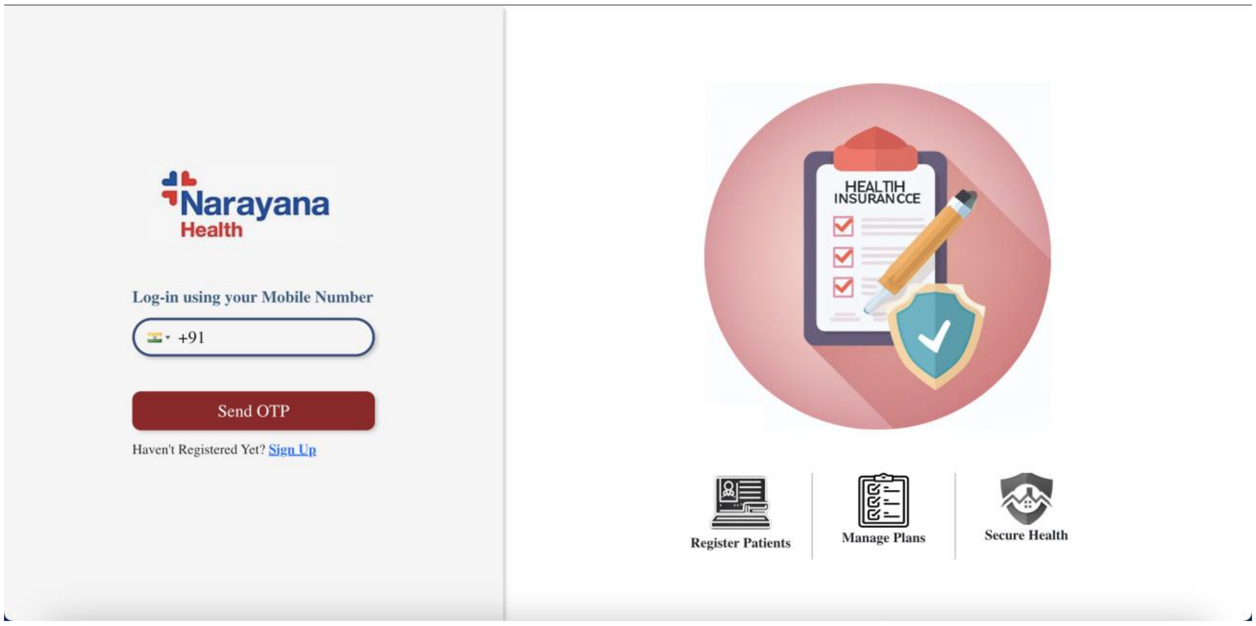
*tax excluded
Rs. 7,000/-

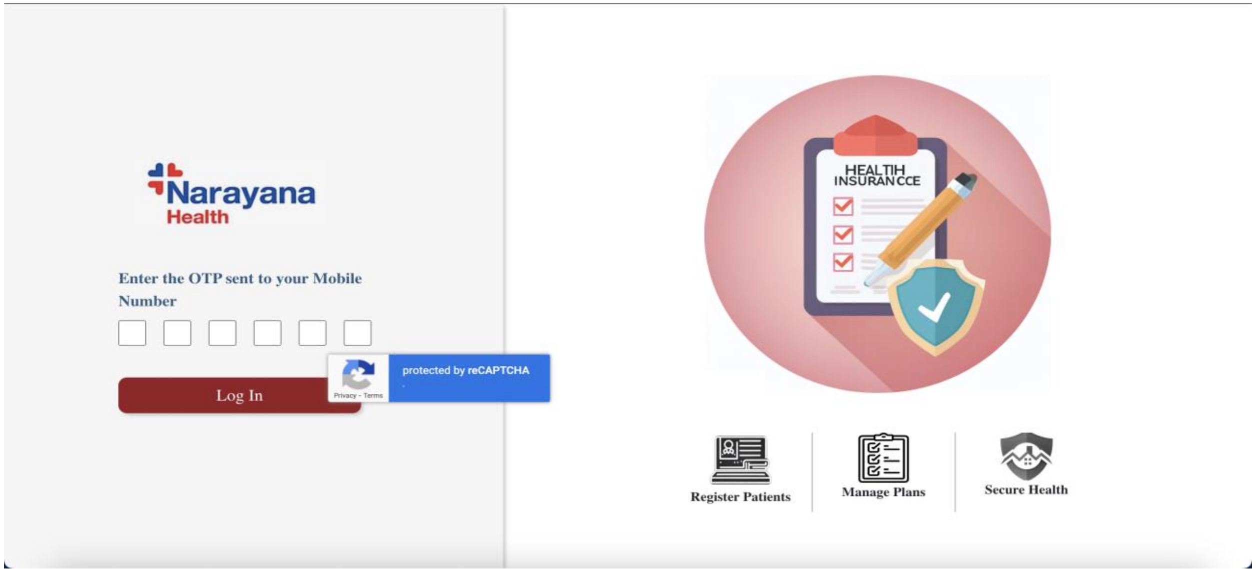
The home page of this health insurance website offers users an intuitive and welcoming interface to explore and purchase health insurance plans. It features three distinct plans designed to cater to various needs and budgets:

1. **Basic Plan:** Provides essential coverage for routine health care needs, ensuring basic medical support.
2. **Standard Plan:** Offers a balanced mix of coverage and benefits, catering to more comprehensive health care requirements.
3. **Premium Plan:** Provides the highest level of benefits and wellness features, offering extensive coverage and convenience.

Users can easily navigate through the available plans, compare their benefits, and select the most suitable option. The design and layout are crafted to prioritize user experience, making it straightforward for users to find and understand the information they need to make informed decisions about their health insurance.

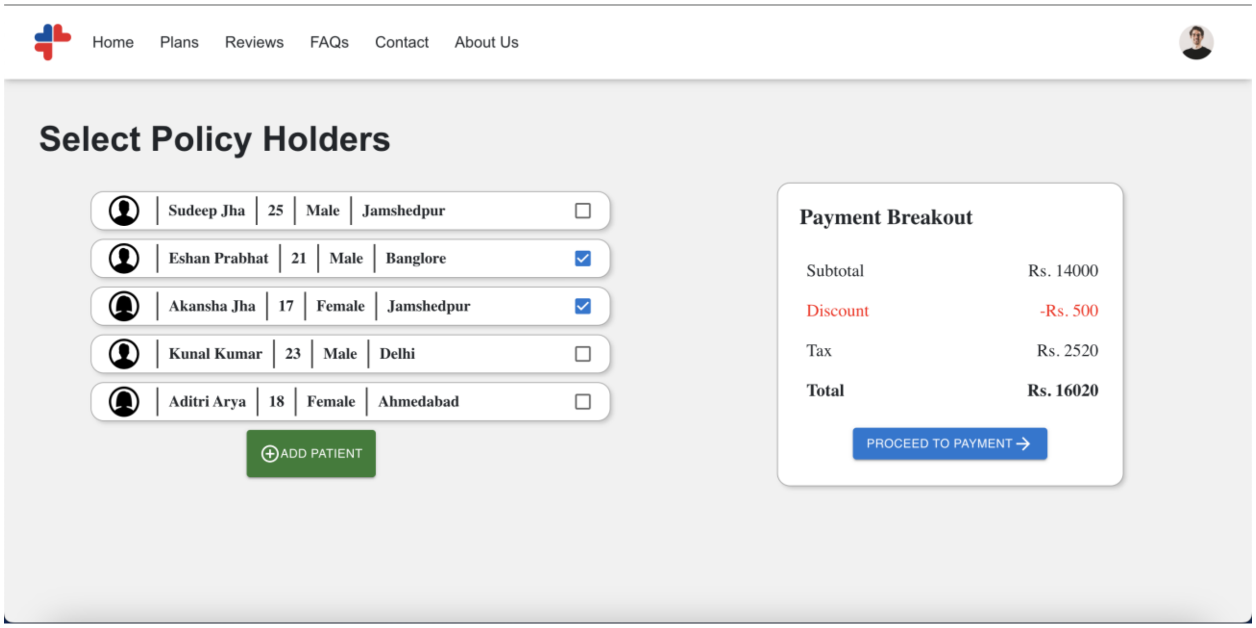
Authentication:





For user authorization, the health insurance website utilizes the Firebase console to ensure secure and efficient access. The authorization process is streamlined through the use of One-Time Passwords (OTP). When a user enters their mobile number, an OTP is sent to that number via SMS. The user then inputs the received OTP on the website to verify their identity and gain access to their account. This method ensures a high level of security and convenience, protecting user data while providing a seamless login experience

Purchase Of Plans:



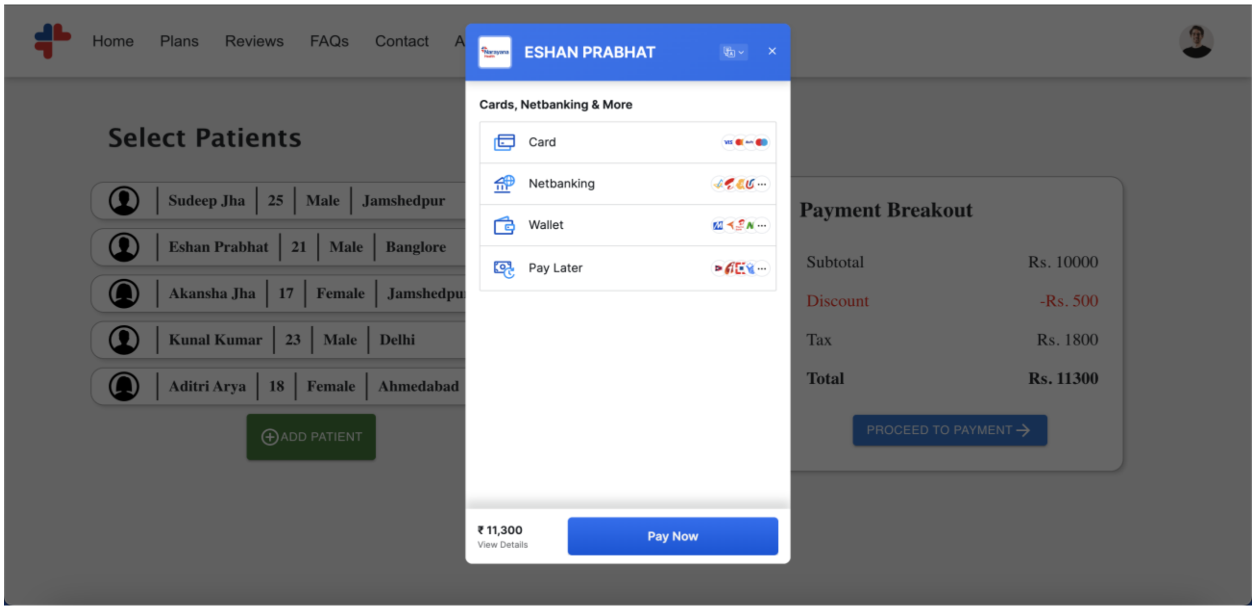
Select Policy Holders

Sudeep Jha	25	Male	Jamshedpur	☐
Eshan Prabhat	21	Male	Banglore	☑
Akansha Jha	17	Female	Jamshedpur	☑
Kunal Kumar	23	Male	Delhi	☐
Aditri Arya	18	Female	Ahmedabad	☐

Payment Breakout

Subtotal	Rs. 14000
Discount	-Rs. 500
Tax	Rs. 2520
Total	**Rs. 16020**

In this part, we must select the policy holders for whom we need to buy this health Insurance plan. On selecting the patients, you will be able to click the button to proceed to the payment part where a popup will appear of a razorpay portal where we can complete the payment.

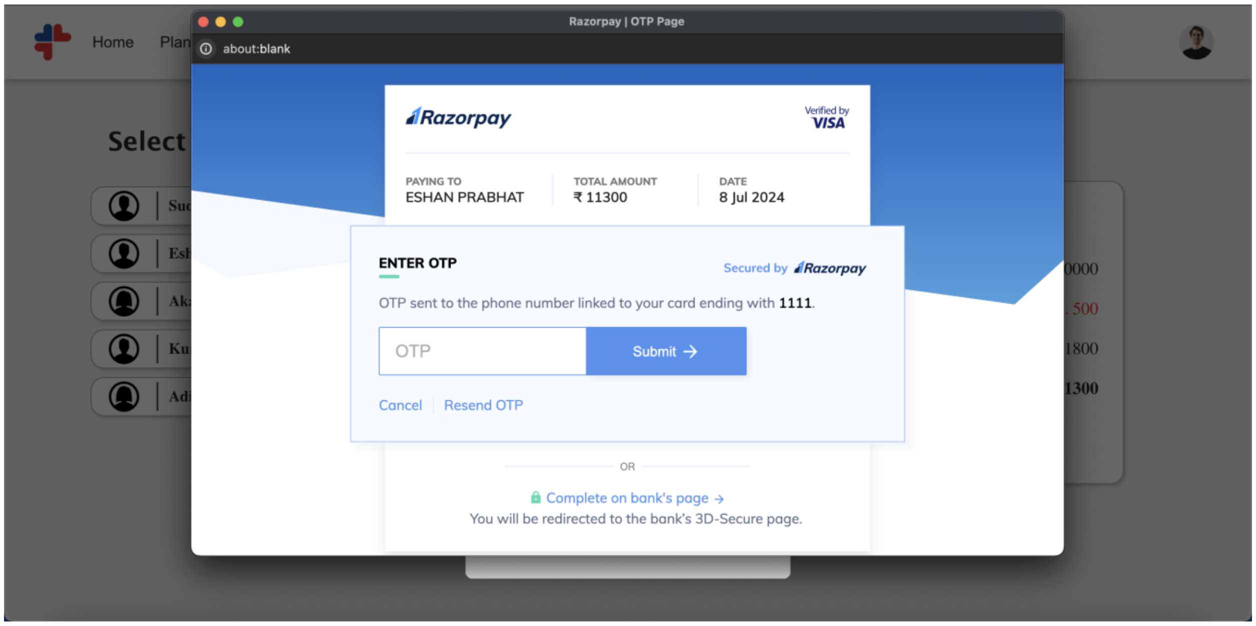


Select Patients

Sudeep Jha	25	Male	Jamshedpur
Eshan Prabhat	21	Male	Banglore
Akansha Jha	17	Female	Jamshedpu
Kunal Kumar	23	Male	Delhi
Aditri Arya	18	Female	Ahmedabad

Payment Breakout

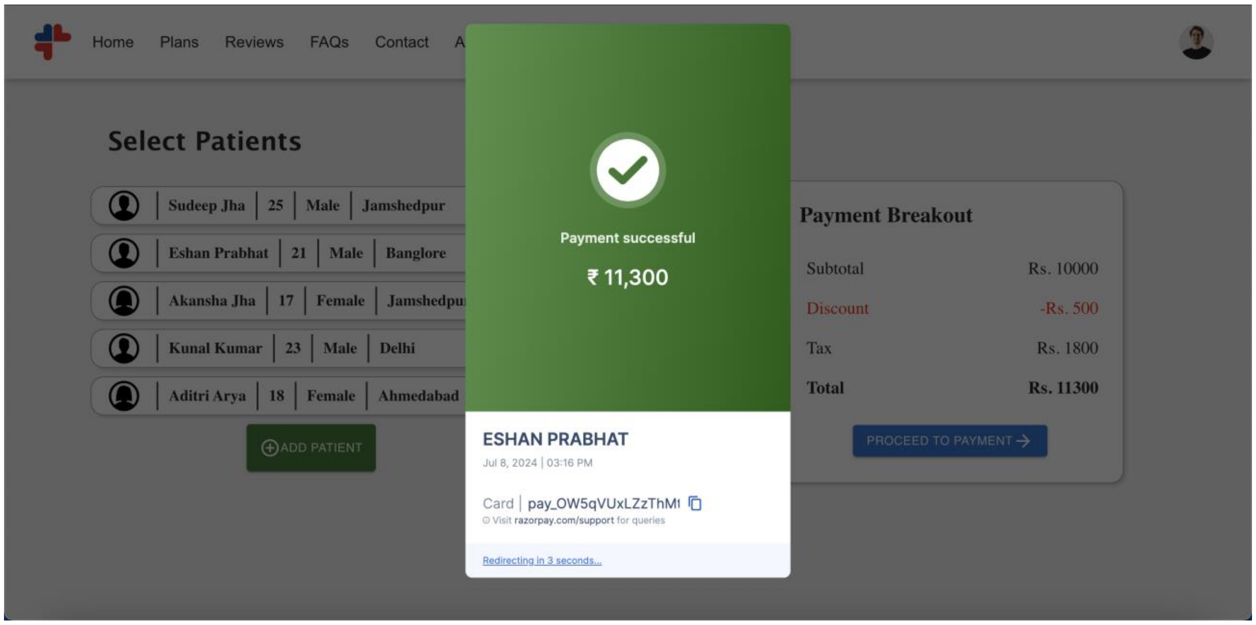
Subtotal	Rs. 10000
Discount	-Rs. 500
Tax	Rs. 1800
Total	Rs. 11300



PAYING TO ESHAN PRABHAT | TOTAL AMOUNT ₹ 11300 | DATE 8 Jul 2024

OTP sent to the phone number linked to your card ending with 1111.

You will be redirected to the bank's 3D-Secure page.



Select Patients

Sudeep Jha	25	Male	Jamshedpur
Eshan Prabhat	21	Male	Banglore
Akansha Jha	17	Female	Jamshedpu
Kunal Kumar	23	Male	Delhi
Aditri Arya	18	Female	Ahmedabad

Payment successful

₹ 11,300

ESHAN PRABHAT

Jul 8, 2024 | 03:16 PM

Card | pay_OW5qVUxLZzThMt

Visit razorpay.com/support for queries

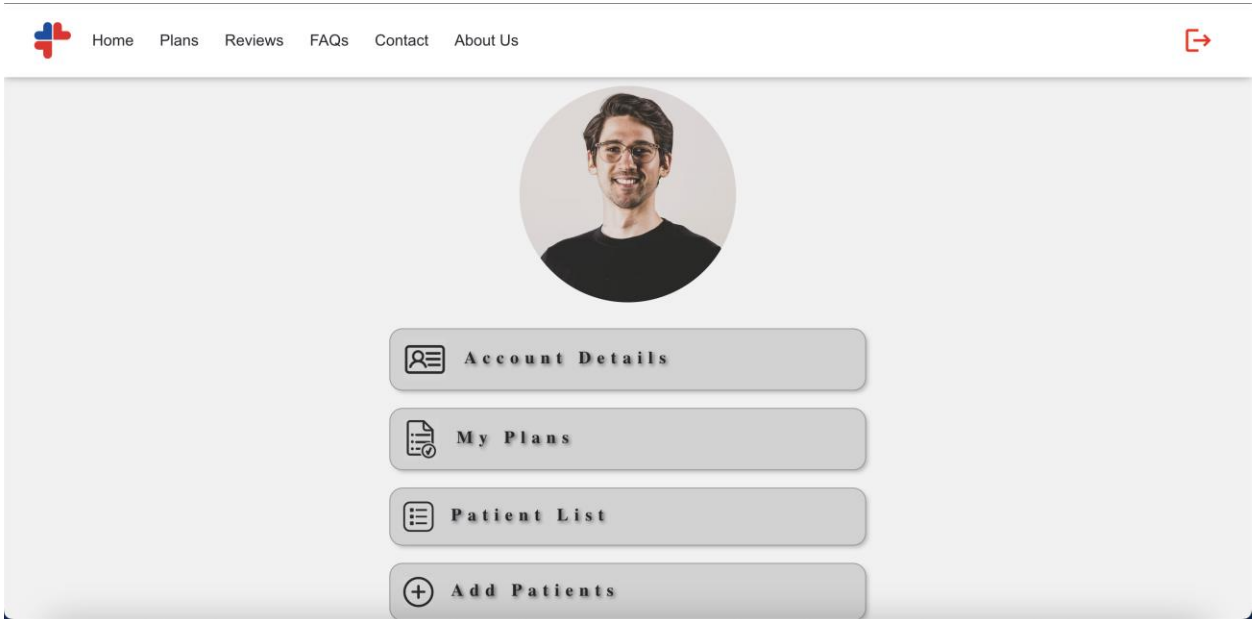
Redirecting in 3 seconds...

Payment Breakout

Subtotal	Rs. 10000
Discount	-Rs. 500
Tax	Rs. 1800
Total	**Rs. 11300**

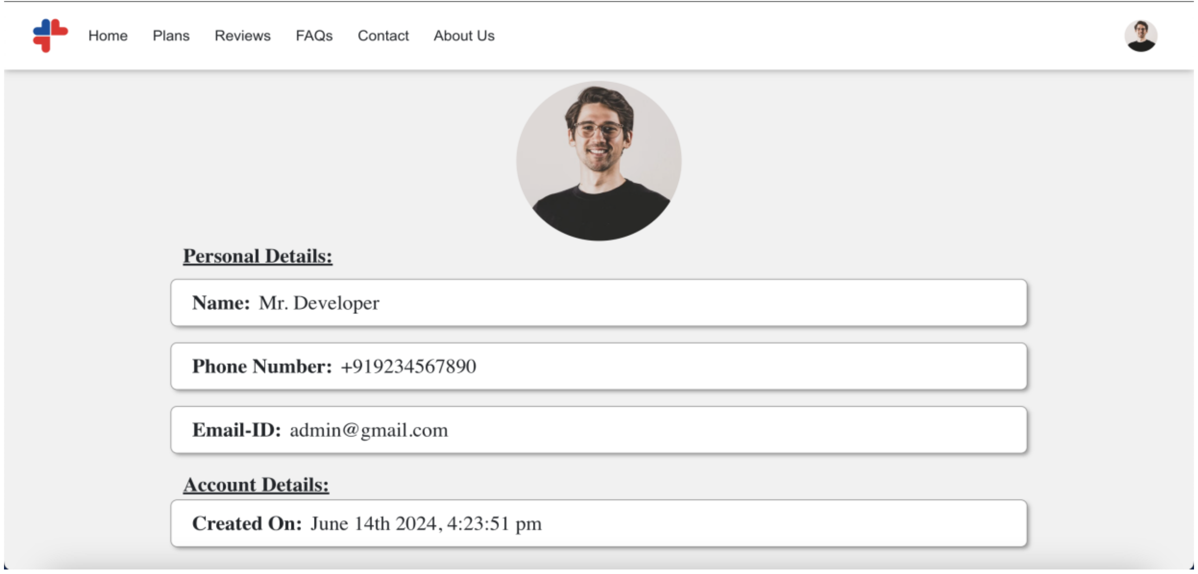
After the completion of the payment process, the plan becomes active, and it is added in the “My Plans” section of the website.

Account Section:



In the Account Section of the Web Application, we can Navigate through the “Account Details”, “My Plans”, “Patient List” and “Add Patient” sections of the web application. And we can upload and change the user profile images which is uploaded to an AWS bucket and then fetched from the AWS bucket to display in the Website.

Account Details:



Personal Details:

Name: Mr. Developer

Phone Number: +919234567890

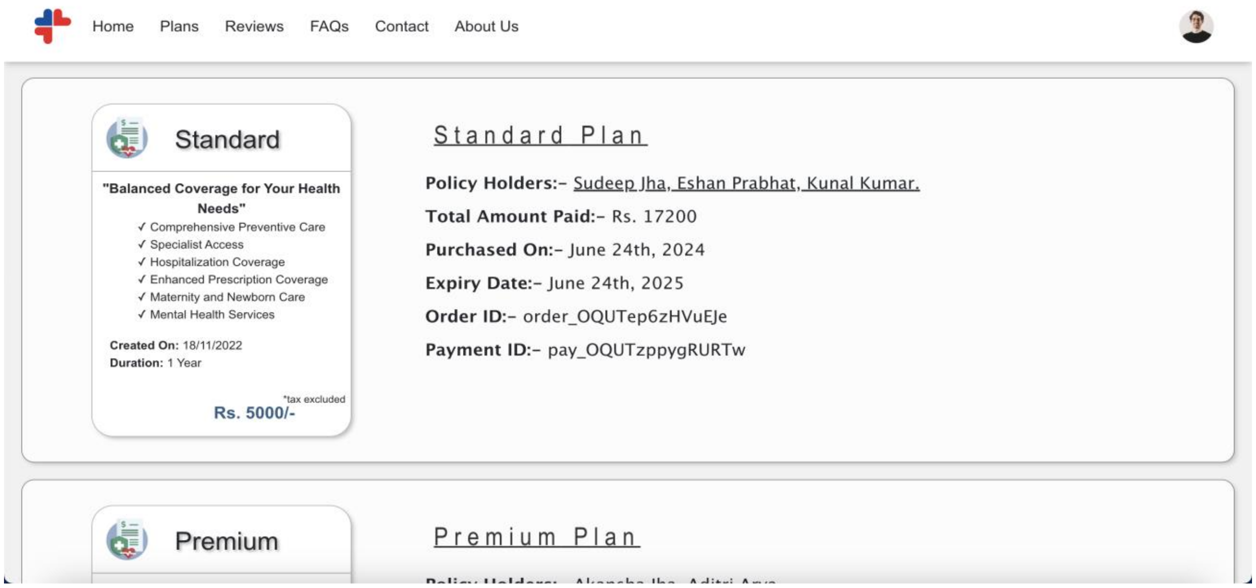
Email-ID: admin@gmail.com

Account Details:

Created On: June 14th 2024, 4:23:51 pm

Information about the logged in user is displayed on this screen.

My Plans:



Standard

"Balanced Coverage for Your Health Needs"

✓ Comprehensive Preventive Care
✓ Specialist Access
✓ Hospitalization Coverage
✓ Enhanced Prescription Coverage
✓ Maternity and Newborn Care
✓ Mental Health Services

Created On: 18/11/2022
Duration: 1 Year

*tax excluded
Rs. 5000/-

Standard Plan

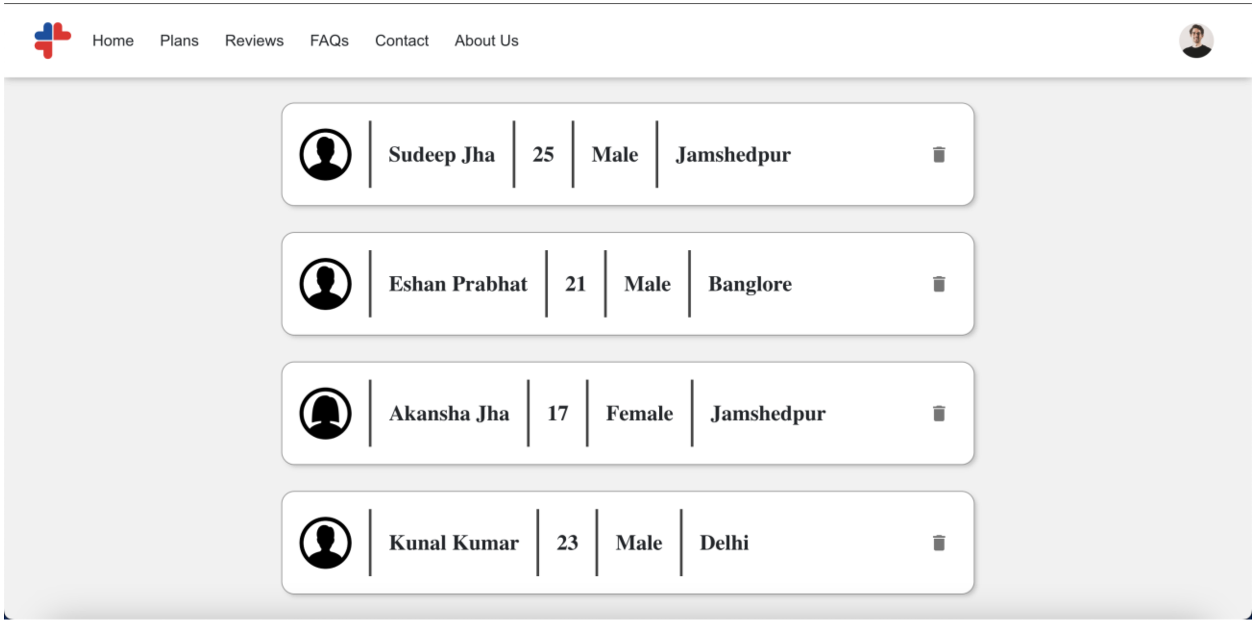
Policy Holders:- Sudeep Jha, Eshan Prabhat, Kunal Kumar.
Total Amount Paid:- Rs. 17200
Purchased On:- June 24th, 2024
Expiry Date:- June 24th, 2025
Order ID:- order_OQUTep6zHVuEJe
Payment ID:- pay_OQUTzppygRURTw

Premium

Premium Plan

Active Plans belonging to the logged in user are displayed on this screen along with all the relevant details.

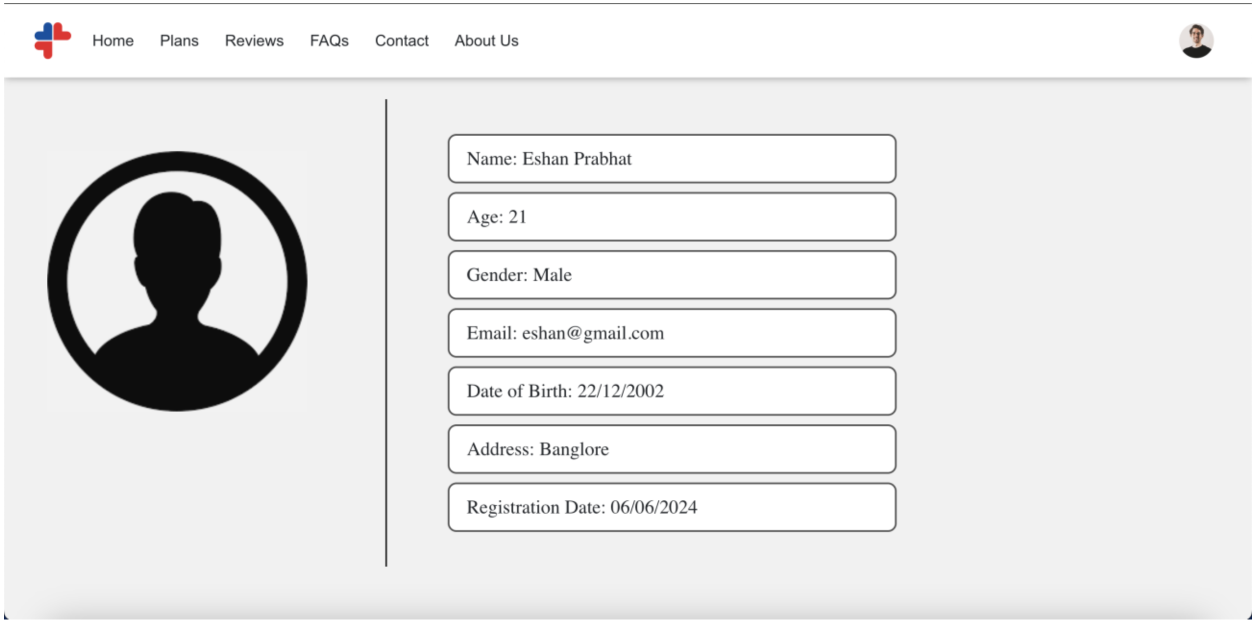
Patient List:



Sudeep Jha	25	Male	Jamshedpur
Eshan Prabhat	21	Male	Banglore
Akansha Jha	17	Female	Jamshedpur
Kunal Kumar	23	Male	Delhi

Clicking on any of the patient cards, we can view the detailed information about that patient. This page also allows us to delete any patient record which will not be displayed after deletion. We do not actually delete the record from the database but use the soft delete method by which we change the show field of the particular patient to false which is true by default. And the API is programmed in such a way that it displays only the records which have the show field as true.

Patient Info:



Name: Eshan Prabhat

Age: 21

Gender: Male

Email: eshan@gmail.com

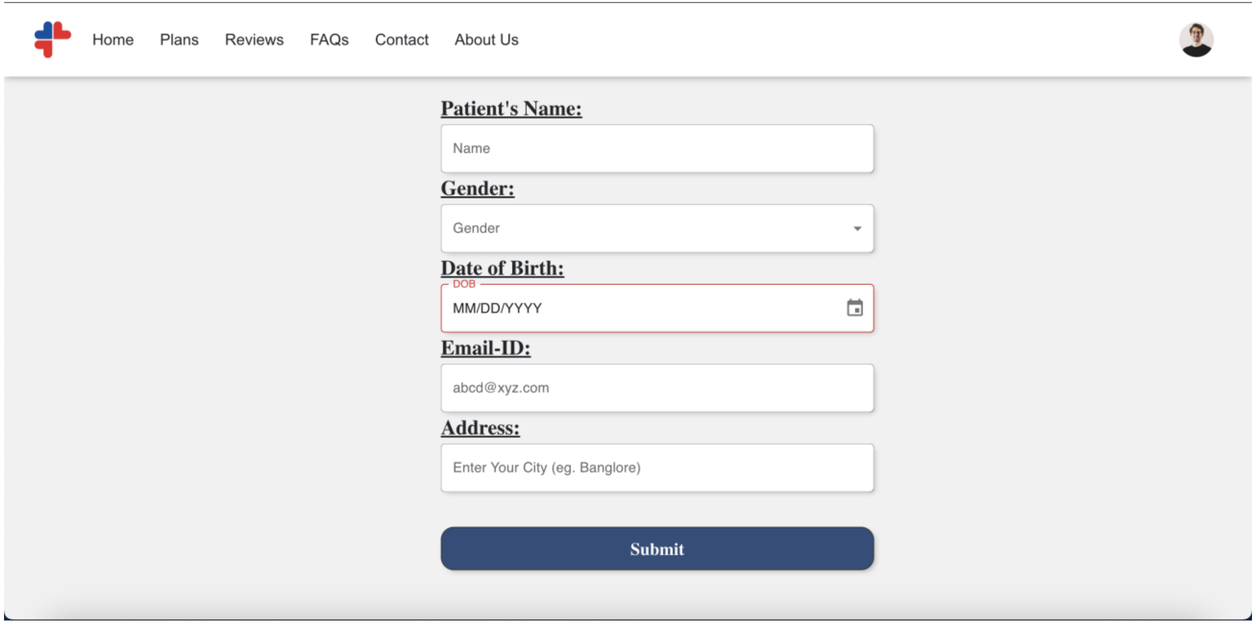
Date of Birth: 22/12/2002

Address: Banglore

Registration Date: 06/06/2024

Detailed Patient Information is displayed on this screen

Add Patient:



Patient's Name:

Name

Gender:

Gender

Date of Birth:

DOB

MM/DD/YYYY

Email-ID:

abcd@xyz.com

Address:

Enter Your City (eg. Banglore)

We can add patients through this section of the web application.

API Documentation

Overview:

This API provides endpoints for managing users, patients, health insurance plans, and patient plan subscriptions. It also includes integration with Razorpay for payment processing and AWS S3 for file uploads.

Base URL:

```
http://localhost:<PORT>
```

Endpoints

User Routes

Get All Users

- **URL:** /api/users
- **Method:** GET
- **Description:** Retrieve all users.
- **Response:**
  - 200 OK: Returns an array of user objects.

Create a User

- **URL:** /api/users
- **Method:** POST
- **Description:** Create a new user.
- **Request Body:**
  - name (string, required)
  - PhoneNumber (string, required)

        - email (string, required)
- **Response:**
    - 201 Created: Returns the created user object.

Get a User by ID

- **URL:** /api/users/:id
- **Method:** GET
- **Description:** Retrieve a user by their ID.
- **Response:**
    - 200 OK: Returns the user object.
    - 404 Not Found: User not found.

Update a User

- **URL:** /api/users/:id
- **Method:** PATCH
- **Description:** Update user details and upload user photo.
- **Request Body:**
    - name (string, optional)
    - email (string, optional)
    - photo (file, optional)
- **Response:**
    - 200 OK: Returns the updated user object.
    - 404 Not Found: User not found.

Plan Routes

Get All Plans

- **URL:** /api/plans
- **Method:** GET

- **Description:** Retrieve all health insurance plans.
- **Response:**
    - 200 OK: Returns an array of plan objects.

Create a Plan

- **URL:** /api/plans
- **Method:** POST
- **Description:** Create a new health insurance plan.
- **Request Body:**
    - plan_name (string, required)
    - price (number, required)
    - tagline (string, required)
    - description (string, required)
    - features (array, required)
    - features_info (string, required)
    - created_on (date, required)
- **Response:**
    - 201 Created: Returns the created plan object.
    - 400 Bad Request: Missing required fields.

Get a Plan by ID

- **URL:** /api/plans/:id
- **Method:** GET
- **Description:** Retrieve a health insurance plan by its ID.
- **Response:**
    - 200 OK: Returns the plan object.
    - 404 Not Found: Plan not found.

Patient Routes

Create a Patient

- **URL:** /api/patients
- **Method:** POST
- **Description:** Create a new patient.
- **Request Body:**
    - user_id (string, required)
    - name (string, required)
    - gender (string, required)
    - email (string, required)
    - dob (date, required)
    - address (string, required)
    - Reg_date (date, required)
    - Show (boolean, required)
- **Response:**
    - 201 Created: Returns the created patient object.
    - 400 Bad Request: Missing required fields.

Get a Patient by ID

- **URL:** /api/patients/:id
- **Method:** GET
- **Description:** Retrieve a patient by their ID.
- **Response:**
    - 200 OK: Returns the patient object.
    - 404 Not Found: Patient not found.

Delete a Patient

- **URL:** /api/patients/:id
- **Method:** DELETE
- **Description:** Soft delete a patient by setting their show attribute to false.
- **Response:**
  - 204 No Content: Patient successfully deleted.
  - 404 Not Found: Patient not found.

Get Patients by User ID

- **URL:** /api/patients/user/:userId
- **Method:** GET
- **Description:** Retrieve all patients associated with a user ID.
- **Response:**
  - 200 OK: Returns an array of patient objects.
  - 404 Not Found: No patients found for the given user ID.

<u>Patient Plan Routes</u>

Create a Patient Plan

- **URL:** /api/patient-plans
- **Method:** POST
- **Description:** Create a new patient plan.
- **Request Body:**
  - user_id (string, required)
  - plan_id (string, required)
  - Start_Date (date, required)
  - Expiry_Date (date, required)
- **Response:**

- 201 Created: Returns the created patient plan object.
- 400 Bad Request: Missing required fields.

Get All Patient Plans

- **URL:** /api/patient-plans
- **Method:** GET
- **Description:** Retrieve all patient plans.
- **Response:**
  - 200 OK: Returns an array of patient plan objects.

Get Patient Plans by User ID

- **URL:** /api/patient-plans/user/:userId
- **Method:** GET
- **Description:** Retrieve all patient plans associated with a user ID.
- **Response:**
  - 200 OK: Returns an array of patient plan objects.
  - 404 Not Found: No patient plans found for the given user ID.

Razorpay Payment Route

Create Razorpay Order

- **URL:** /razorpay
- **Method:** POST
- **Description:** Create a new Razorpay order.
- **Request Body:**
  - amount (number, required) - Amount in INR
- **Response:**

- 200 OK: Returns the created Razorpay order details.
- 500 Internal Server Error: Error creating Razorpay order.

Static File Route

Serve Logo Image

- **URL:** /logo.jpg
- **Method:** GET
- **Description:** Serve the logo image file.
- **Response:**
    - 200 OK: Returns the logo image.

Error Handling

All endpoints follow a consistent error response structure:

```
{
  "status": "fail",
  "message": "<error-message>"
}
```

Environment Variables:

Make sure to set the following environment variables in your .env file:

- PORT: The port on which the server will run.
- DATABASE: MongoDB connection string with a placeholder for the password.
- DATABASE_PASSWORD: Password for the MongoDB connection.
- RAZORPAY_KEY_ID: Razorpay API key ID.

- RAZORPAY_KEY_SECRET: Razorpay API key secret.
- AWS_BUCKET_REGION: AWS S3 bucket region.
- AWS_ACCESS_KEY: AWS access key.
- AWS_ACCESS_SECRET: AWS secret access key.
- AWS_BUCKET_NAME: AWS S3 bucket name.

Deployment:

- **The User Interface is Deployed on Netlify.com**
  Link: https://myhealthinsurancenh.netlify.app

- **The Server is deployed on Render.com**

Links:

- **Github: https://github.com/eshanprabhat/nh_project**