C4GT's DMP 2024 – Proposal

Name	GAURAV PATIDAR
Email ID	Gaurav05082002@gmail.com
Phone Number	7689816680
GitHub ID	https://github.com/Gaurav05082002
Discord ID	Gaurav05082002
Current occupation *(Working Professionals - add current organization & years of exp)*	Student
Education Details *(College Name - Degree Name and branch of engineering or other course/specialization)*	IIT KHARAGPUR DUAL DEGREE AGRICULTURAL ENGINEERING & COMPUTER SCIENCE ENGINEERING (Minor)
Technical skills with level *(Mention tech skills/languages known/UI-UX and level - Novice/Intermediate/Expert)*	Expert (ReactsJs , Js ,Html , Css , MERN , Mongodb , Golang, C ,C++ , Node Js , AWS , Python , SQL) Intermediate (Docker , Php)

Title:

STATE OF WELFARE MICROSITE

Summary

Our approach to the State of Welfare Microsite project is centered around designing and implementing a robust digital infrastructure to support the efficient hosting of information and knowledge base that is gained from surveys . We aim to develop comprehensive data pipelines that seamlessly collect, process, integrate, and analyze various forms of healthcare-related information, ensuring accuracy, security, and compliance with regulatory standards. By leveraging modern technologies and best practices, we seek to create a scalable and adaptable platform that empowers stakeholders to make informed decisions, enhance welfare services , and drive positive outcomes across the welfare ecosystem.

Project Detail

1. **Project Overview:**

   a. **Understanding of the project**

   The project involves creating a comprehensive microsite that aggregates and shares insights on welfare services in the country. This initiative, in collaboration with various stakeholders from community organizations, government, and private sector, aims to provide a 360-degree view of the current welfare landscape through data from surveys, participant inputs, and expert analyses. The goal is to make this information publicly accessible and interactive, allowing citizens to contribute their feedback and engage with the platform. The microsite will be built with an emphasis on user accessibility, robust data structures, and interactive features, and will be licensed under Creative Commons to ensure broad usage and distribution.

   To effectively disseminate this information, we plan to develop a microsite. This microsite will serve as a knowledge base that is accessible to all citizens, enabling them to view insights, interact with the site (potentially through a chat feature), and share their own inputs on the welfare ecosystem.

   The project goals are:

   - Design the microsite in coordination with the SOWR working group.
   - Design the underlying data structures.
   - Develop the data pipelines for different forms of data and content.
   - Build the MVP of the site for demonstration to the SOWR working group.

   b. **Problems**

Data Integration Challenges: Integrating diverse forms of data (structured and unstructured) from multiple sources may present technical challenges, especially in ensuring consistency and accuracy.

User Accessibility: Ensuring the microsite is easily accessible and user-friendly for a wide range of users, including those with limited technical skills or disabilities, can be challenging.

Scalability: The microsite must be scalable to handle potentially large amounts of data and user interactions without performance degradation.

Data Privacy and Security: Safeguarding the privacy and security of the data, especially if it includes sensitive information, is crucial.

c. Solutions

Robust Data Architecture: Design a robust data architecture that can seamlessly integrate structured and unstructured data. Utilize ETL (Extract, Transform, Load) processes and data normalization techniques to ensure data consistency and accuracy.

User-Centric Design: Employ a user-centric design approach, involving usability testing and feedback loops, to ensure the microsite is intuitive and accessible. Implement accessibility standards (e.g., WCAG) to cater to users with disabilities.

Scalable Infrastructure: Use cloud-based infrastructure (e.g., AWS) and scalable technologies (e.g., microservices architecture) to ensure the microsite can handle varying loads efficiently. Implement content delivery networks (CDNs) to optimize performance and load times.

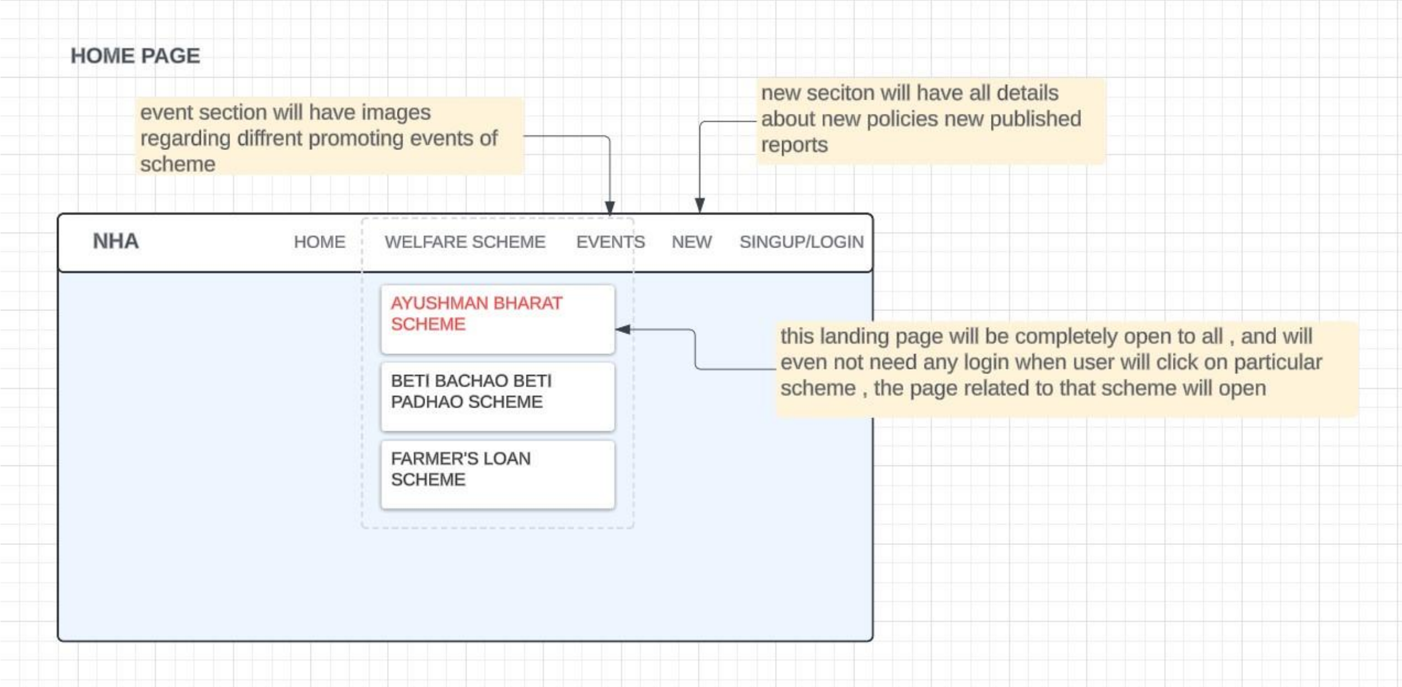
Data Privacy and Security Measures: Implement robust data privacy and security measures, including encryption, secure authentication mechanisms, and compliance with data protection regulations. Conduct regular security audits and vulnerability assessments to safeguard against potential threats.

Demo Implementation of Product Logic taking example of SOWR working group for AYUSHMAN BHARAT WELFARE SERVICE (ABWS)

During the development of the microsite, there will be repetition of many components. Hence, we should go with a template-oriented library. We will use REACTJS. One more feature that makes it feasible for our case is that we will make a website and will make it responsive via MEDIA QUERIES IN CSS. We can make the application a mobile app by using the IONIC CAPACITOR LIBRARY.

In case of ABWS, we will have three main stack holders
Samaj (Public , users , who will take the benefits of scheme)
Sarkar (govt authorities who handles the resource , here national health authority)
Bazaar (Hospitals and clinics who connects the resources to Users)

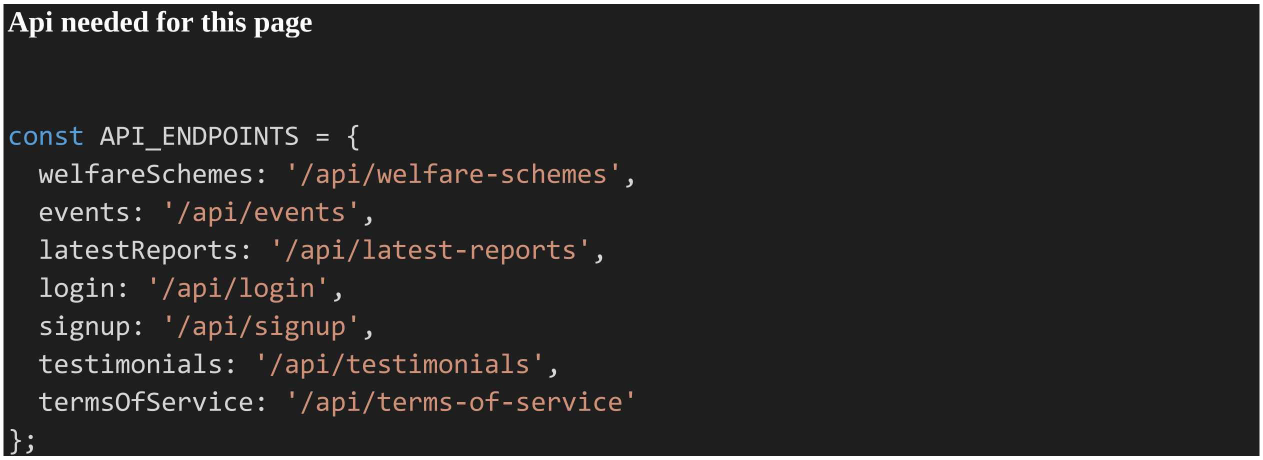
Our homepage will serve as the initial landing page, where users can choose the scheme they wish to explore for insights or conduct surveys. (The page will be created by react tech stack)



event section will have images regarding diffrent promoting events of scheme

new seciton will have all details about new policies new published reports

this landing page will be completely open to all , and will even not need any login when user will click on particular scheme , the page related to that scheme will open

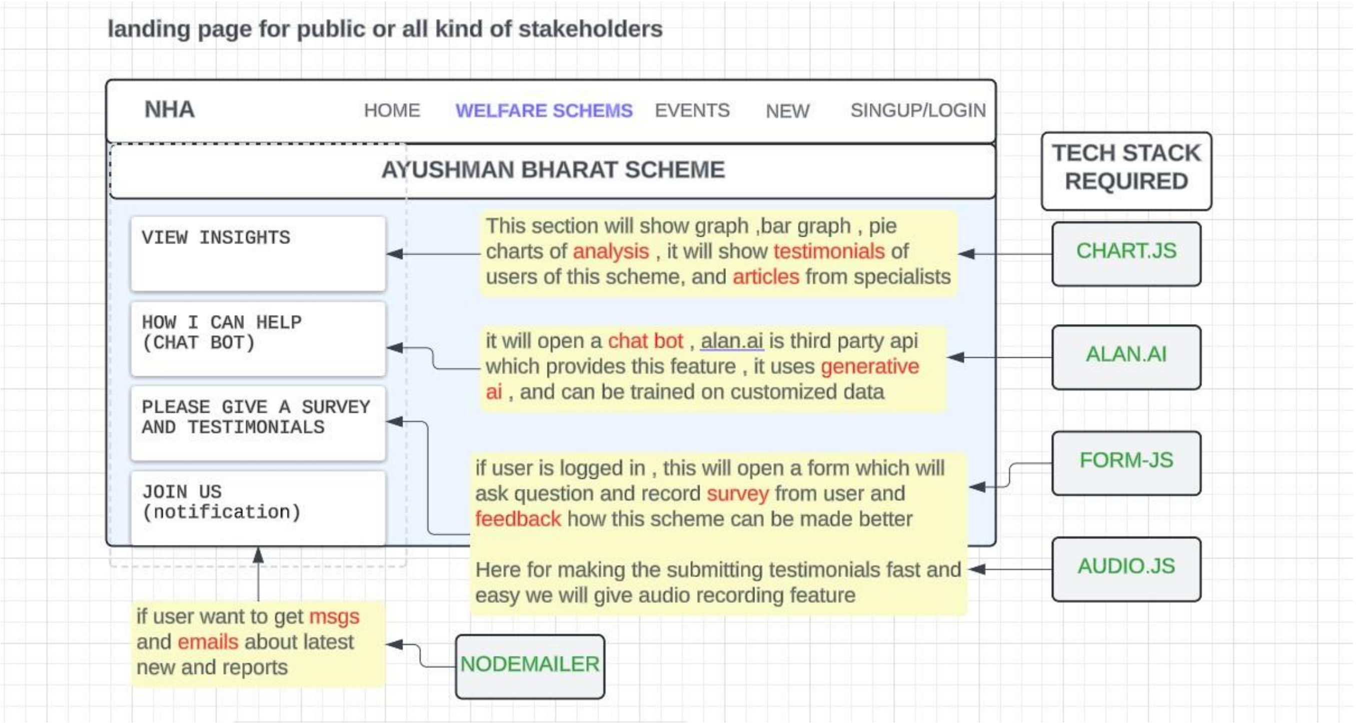


Api needed for this page

```
const API_ENDPOINTS = {
  welfareSchemes: '/api/welfare-schemes',
  events: '/api/events',
  latestReports: '/api/latest-reports',
  login: '/api/login',
  signup: '/api/signup',
  testimonials: '/api/testimonials',
  termsOfService: '/api/terms-of-service'
};
```

Once the Scheme/Service is selected there will be three different kind of landing page respective to three different type of stack holders. (samaj , sarkar , bazar)

First type of landing page (open for all and can be used without login)



This section will show graph ,bar graph , pie charts of analysis , it will show testimonials of users of this scheme, and articles from specialists

it will open a chat bot , alan.ai is third party api which provides this feature , it uses generative ai , and can be trained on customized data

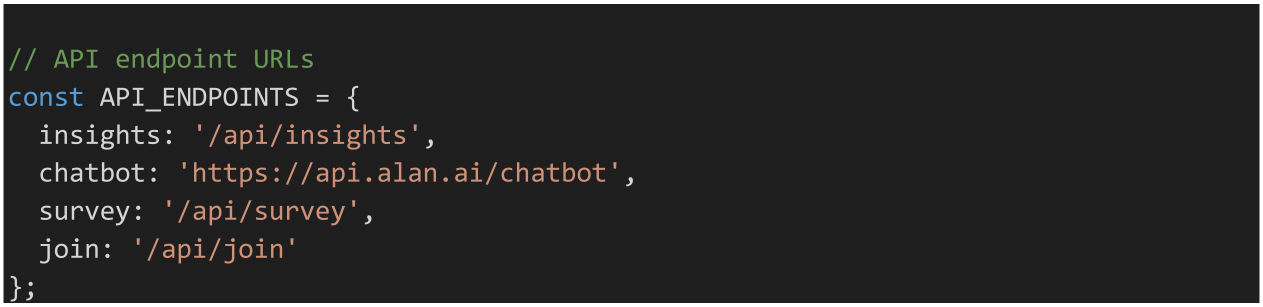
if user is logged in , this will open a form which will ask question and record survey from user and feedback how this scheme can be made better

Here for making the submitting testimonials fast and easy we will give audio recording feature

if user want to get msgs and emails about latest new and reports

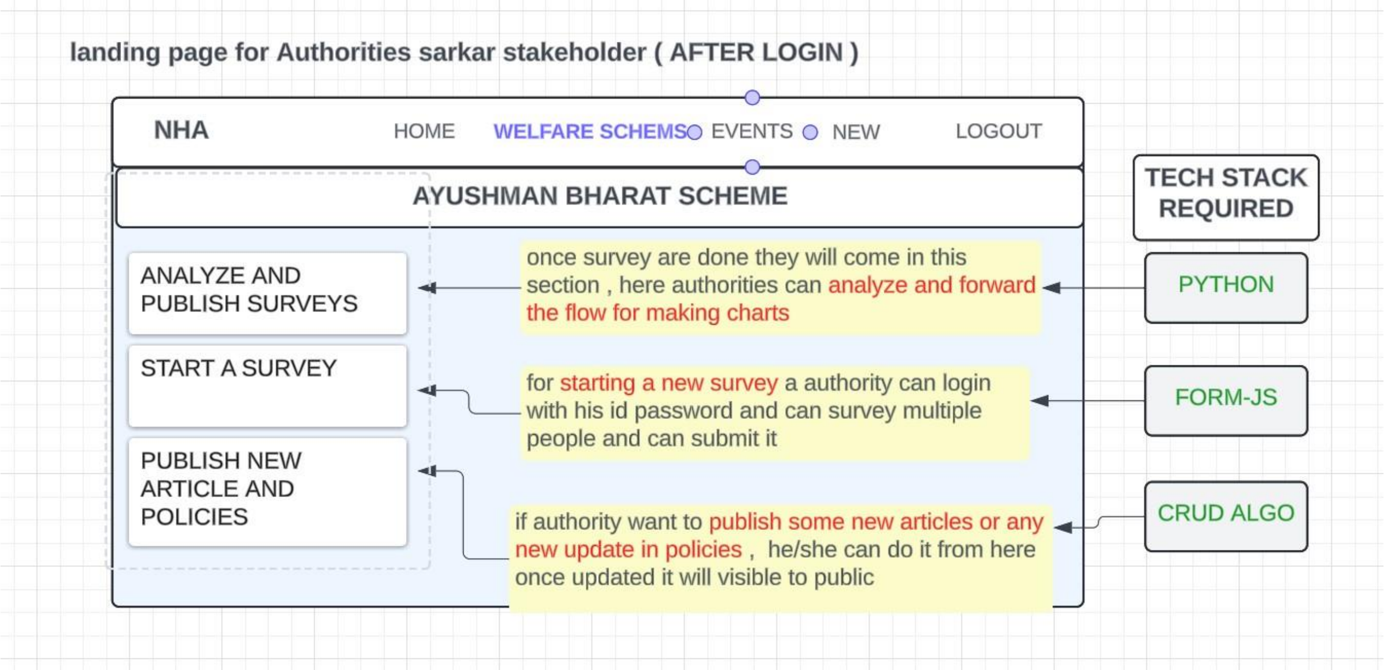
This page will help you to see chart analysis , and other insights about scheme .
Making testimonials giving process easy and fast we will give audio recording option through audio,js library
Here we will use alan.ai to make a chat bot to which user can ask question regarding this scheme .
(REACT , CHART.JS , ALAN.AI , FORM-JS , AUDIO.JS will help to complete this web page)

Required Api Endpoints for this page



```
// API endpoint URLs
const API_ENDPOINTS = {
  insights: '/api/insights',
  chatbot: 'https://api.alan.ai/chatbot',
  survey: '/api/survey',
  join: '/api/join'
};
```

Second type of landing page (when authorities login by their own login id) (Admin Dashboard)


landing page for Authorities sarkar stakeholder (AFTER LOGIN)
NHA
HOME
WELFARE SCHEMS
EVENTS
NEW
LOGOUT
AYUSHMAN BHARAT SCHEME
ANALYZE AND PUBLISH SURVEYS
START A SURVEY
PUBLISH NEW ARTICLE AND POLICIES
TECH STACK REQUIRED
PYTHON
FORM-JS
CRUD ALGO

once survey are done they will come in this section , here authorities can analyze and forward the flow for making charts

for starting a new survey a authority can login with his id password and can survey multiple people and can submit it

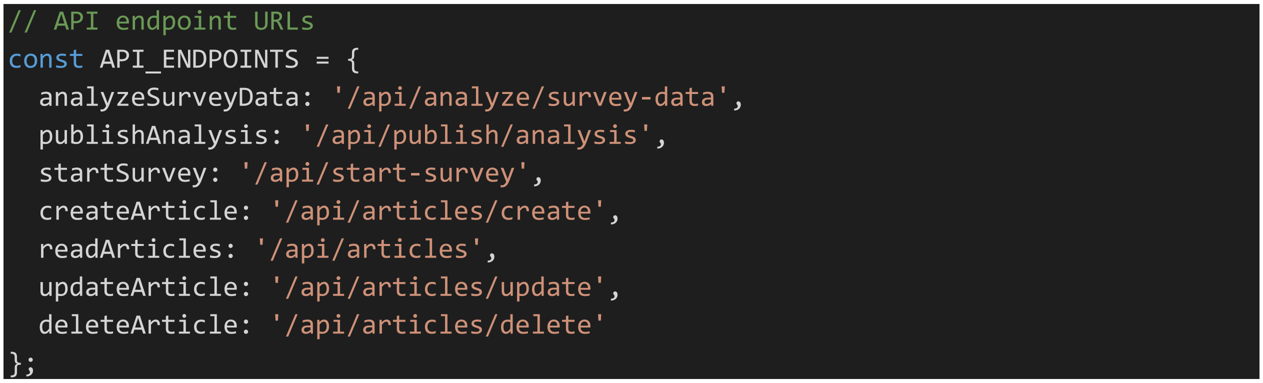
if authority want to publish some new articles or any new update in policies , he/she can do it from here once updated it will visible to public

From this page authority can publish the surveys for public, also a new survey can be started if any authority want to get the survey of some individua , also from here authority can publish new articles and updates in policies ,also here will be dashboard to check state of welfare scheme

The data recorded saved in DB or any csv excel file can be given to feed data to analyzing model and with help of CHART.JS charts will be made and can be published .

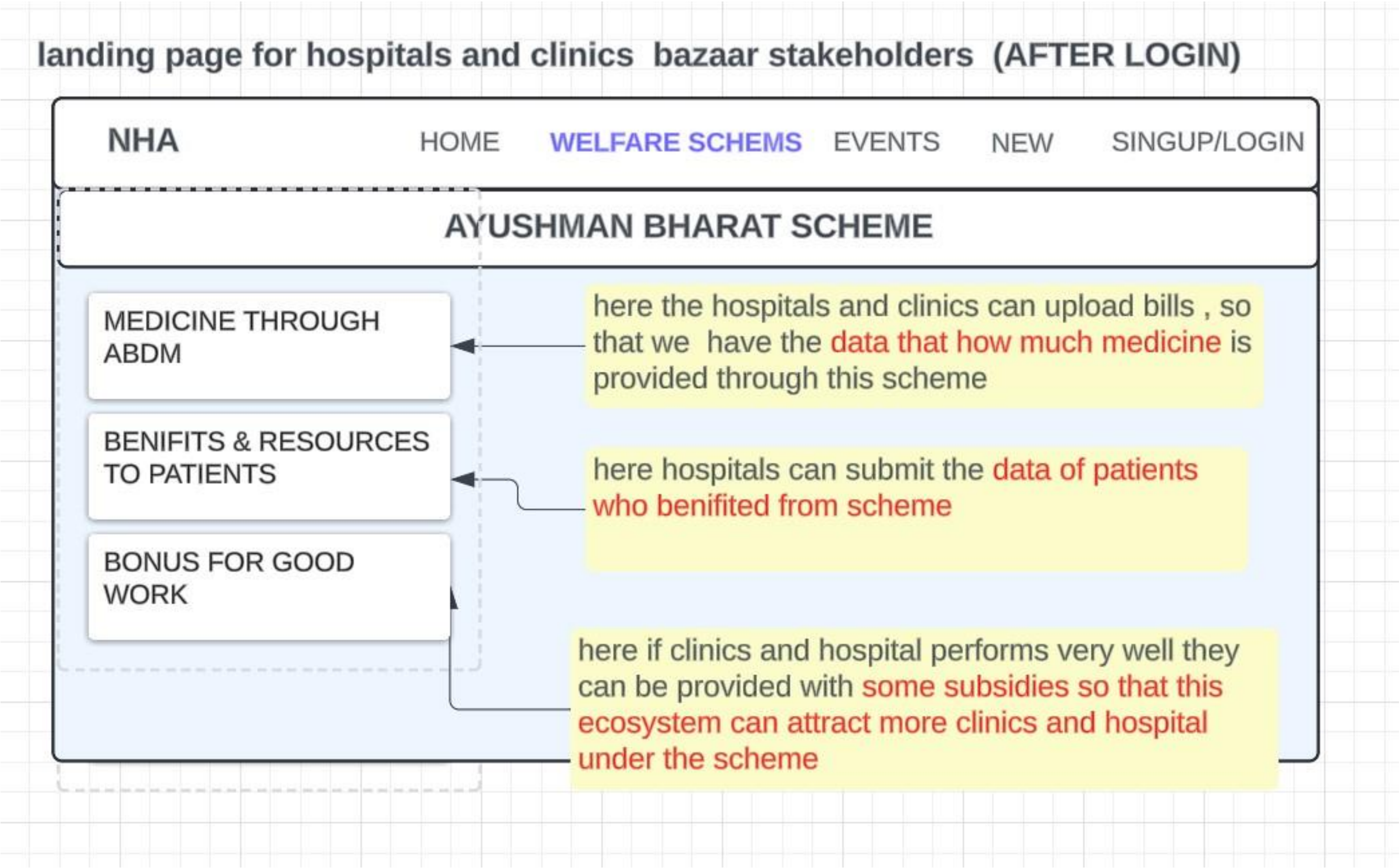
this page will use REACT , PYTHON for data analysis of excel or csv file , FORM-JS for making forms for taking the survey , CRUD algorithm for updates and publishing of new policies

<u>Api end points for this landing page</u>



```
// API endpoint URLs
const API_ENDPOINTS = {
  analyzeSurveyData: '/api/analyze/survey-data',
  publishAnalysis: '/api/publish/analysis',
  startSurvey: '/api/start-survey',
  createArticle: '/api/articles/create',
  readArticles: '/api/articles',
  updateArticle: '/api/articles/update',
  deleteArticle: '/api/articles/delete'
};
```

Third type of landing page (when hospitals and clinics login by their own login id)



landing page for hospitals and clinics bazaar stakeholders (AFTER LOGIN)

here the hospitals and clinics can upload bills , so that we have the data that how much medicine is provided through this scheme

here hospitals can submit the data of patients who benifited from scheme

here if clinics and hospital performs very well they can be provided with some subsidies so that this ecosystem can attract more clinics and hospital under the scheme

Api end points for this landing page

```
// API endpoint URLs

const API_ENDPOINTS = {

  uploadBills: '/api/upload-bills',
  submitPatientData: '/api/submit-patient-data',
  requestSubsidies: '/api/request-subsidies'

};
```

Data structure and Data Pipeline

Designing the underlying data structures for the Ayushman Bharat Digital Mission in MongoDB involves defining collections and documents to store healthcare-related information efficiently. Here's a proposed outline

```
const mongoose = require('mongoose');

// Schema for General Users/Public landing page
const UserSchema = new mongoose.Schema({
    username: String,
    email: String,
    // Other user information
}, { timestamps: true });

const SurveyResponseSchema = new mongoose.Schema({
    userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
    surveyData: { type: Object },
    // Other survey response details
}, { timestamps: true });

// Schema for Admin Dashboard landing page
const ArticleSchema = new mongoose.Schema({
    title: String,
    content: String,
    author: String,
    // Other article details
}, { timestamps: true });

const SurveyAnalysisSchema = new mongoose.Schema({
    // Analysis data
    // Other analysis details
}, { timestamps: true });

// Schema for Hospitals and Clinics landing page
const BillSchema = new mongoose.Schema({
    hospitalId: { type: mongoose.Schema.Types.ObjectId, ref: 'Hospital' },
    patientId: { type: mongoose.Schema.Types.ObjectId, ref: 'Patient' },
    medicine: String,
    quantity: Number,
    date: Date
}, { timestamps: true });



const PatientSchema = new mongoose.Schema({
    name: String,
    age: Number,
    gender: String,
    diagnosis: String,
    treatmentDetails: String
}, { timestamps: true });

const SubsidyRequestSchema = new mongoose.Schema({
    hospitalId: { type: mongoose.Schema.Types.ObjectId, ref: 'Hospital' },
    performanceMetrics: { type: Map, of: Number },
    requestedAmount: Number,
    status: { type: String, default: 'pending' } // Status can be 'pending', 'approved', 'rejected', etc.
}, { timestamps: true });

// Define models
const User = mongoose.model('User', UserSchema);
const SurveyResponse = mongoose.model('SurveyResponse', SurveyResponseSchema);
const Article = mongoose.model('Article', ArticleSchema);
const SurveyAnalysis = mongoose.model('SurveyAnalysis', SurveyAnalysisSchema);
const Bill = mongoose.model('Bill', BillSchema);
const Patient = mongoose.model('Patient', PatientSchema);
const SubsidyRequest = mongoose.model('SubsidyRequest', SubsidyRequestSchema);

module.exports = {
    User,
    SurveyResponse,
    Article,
    SurveyAnalysis,
    Bill,
    Patient,
    SubsidyRequest
};
```

Unit tests (some tests are written for demo purpose)

```
// Import necessary modules and functions
const { User, Article, SurveyResponse, Bill, Patient, SubsidyRequest } =
require('./models');
const { app } = require('./app');
const request = require('supertest');

// Unit tests for models
describe('Models', () => {
  // User model tests
  describe('User Model', () => {
    test('should create a new user', async () => {
      const userData = { username: 'testuser', email: 'test@example.com' };
      const newUser = await User.create(userData);
      expect(newUser.username).toBe(userData.username);
      expect(newUser.email).toBe(userData.email);
    });
    //more tests for CRUD operations, data validation, etc.
  });
});

// Integration tests for API endpoints
describe('API Endpoints', () => {
  // Test user registration endpoint
  describe('User Registration', () => {
    test('should register a new user', async () => {
      const userData = { username: 'testuser', email: 'test@example.com',
password: 'password123' };
      const res = await request(app).post('/api/register').send(userData);
      expect(res.status).toBe(200);

    });

  });


});
```

2. Implementation Details with timelines:

Goal 1: Design the Microsite in Coordination with the SOWR Working Group

a. Milestone 1: Initial Meetings and Requirements Gathering (Weeks 1-2)

- Conduct initial meetings with the SOWR working group to understand project objectives and requirements.
- Gather input on design preferences, features, and functionalities from stakeholders.
- Document the gathered requirements and create a project roadmap.

b. Milestone 2: Wireframing and Mockup Development (Weeks 3-4)

- Develop wireframes and low-fidelity mockups based on the documented requirements.
- Present the initial designs to the SOWR working group for feedback and validation.
- Iterate on the designs based on stakeholder feedback and finalize the layout and structure of the microsite.

c. Milestone 3: High-Fidelity Design and Prototype Creation (Weeks 5-6)

- Refine the approved wireframes into high-fidelity designs incorporating branding elements and visual aesthetics.
- Develop a clickable prototype of the microsite to demonstrate the user flow and interaction design.
- Conduct usability testing with representative users to gather feedback on the prototype and make necessary refinements.

Goal 2: Design the Underlying Data Structures

a. Milestone 4: Data Architecture Planning (Week 7)

- Collaborate with data architects and stakeholders to define the data requirements and architecture for the microsite.
- Determine the types of data to be collected, stored, and managed, considering both structured and unstructured data sources.

b. Milestone 5: Data Model Design (Weeks 8-9)

- Develop data models and schemas based on the defined data requirements and architecture.

- Ensure that the data models are scalable, efficient, and capable of supporting the desired functionalities of the microsite.

c. Milestone 6: Integration and Testing (Week 10)

- Implement the designed data structures into the microsite's backend architecture.
- Conduct integration testing to verify the functionality and performance of the data structures.
- Address any issues or discrepancies identified during testing and optimize the data structures as needed.

Availability

The duration of the coding period is from June to September. Please share your availability in detail

Number of hours available to dedicate to this project per week	40 hours
Do you have any other engagements during this period ? (projects/internships)	NO

Personal Information

About Me:

I am a passionate and driven software developer with four years of experience, currently pursuing my studies at IIT Kharagpur. As the former tech head of Spring Fest which is one of the biggest college fest in Asia, I spearheaded the development of a robust web application that efficiently handled over 10,000 registrations. With a stellar problem-solving ability, evidenced by my 1600+ rating on Codeforces, I thrive on challenges and am dedicated to crafting elegant solutions. When I'm not immersed in code, you can find me honing my strategic skills on the chessboard, where I apply the same analytical mindset that drives my development projects.

What is your motivation to apply for this project? Answer briefly in 3-4 lines.

My motivation to apply for this project is deeply rooted in my four years of hands-on experience across diverse tech stacks such as React, React Native, Django, MongoDB, SQL, Docker, MERN , AWS, and Go. Through my journey, I've honed my problem-solving skills and gained valuable insights into optimizing user experiences. Having completed two industry-level internships, I'm eager to channel my expertise towards this project. Contributing to this transformative initiative aligns with my aspirations to be part of a digital revolution that improves the nation's digital infrastructure, benefiting countless individuals and communities.

Previous experience/open source projects (Optional):

In this section you can mention your relevant work experience/projects (not just limited to open-source). You should mention experiences in this section if any with the relevant tech stack of the project

Project Name	**Project Description**	**Links (if any)**
DENSITY EXCHNAGE INTERN	FANTASY GAME APP (LIKE DREAM 11)	REPO IS PRIVATED BY COMPANY
ENGPHILE INNOVATION LABS INTERN	AUDIO BASED EDUCATION APP	https://github.com/Gaurav05082002/phyeng
SPRING FEST MAIN WEBSITE	HIGH END 3D WEBSITE	Spring Fest 2023 (sf2k23-main-lilac.vercel.app)
SQL EDITOR	ALLOWS TO RUN SQL QUERY AND PROPER SEARCH IN TABLE	https://gaurav-sql-editor.netlify.app/

GSAP LIBRARY ASSIGNMENT	ANIMATED WEBSITE	https://animated-web-eight.vercel.app/
CV BUDDY	OPEN SOURCE CONTRIBUTION	https://github.com/praneeth-rdy/CV-Buddy/pull/10
PULL SHARK	OPEN SOURCE CONTRIBUTION	https://github.com/codemistic/Web-Development/pull/291
DASHBOARD	USING CHART.JS GRAPHS AND ANALYSIS	https://gaurav05082002.github.io/CuvetteDashboard/
QR READER	JS LIBRARY TO MAKE A QR CODE READER TO VERIFY ENTRY	https://github.com/Gaurav05082002/qr-reader

Contribution in C4GT's open community:

In this section answer the questions about your participation in C4GT's open community tickets, provide the screenshot of the leaderboard with your GitHub ID and DPG points earned (if any).

Have you contributed to tickets in C4GT's open community? **(Mandatory to answer)**	Yes
Have you successfully completed C4GT's GitHub Classroom Assignment? **(Mandatory to answer)**	Partial
Enter your DPG points **(Mandatory to answer, Enter 0 if not applicable)**	0 (already had experience of open source & version control)

Screenshot of leaderboard with your GitHub ID **(Mandatory to answer, enter 0 if not applicable)**	0 (already had experience of open source & version control)