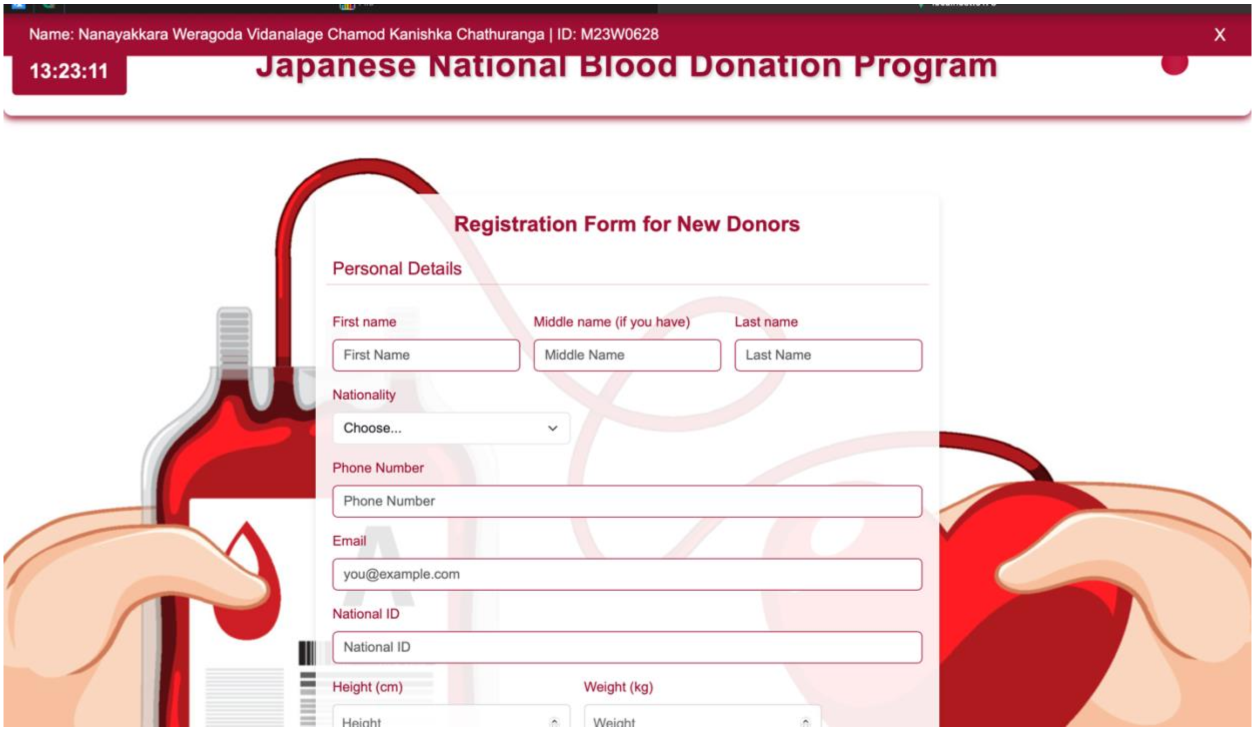
Name: Nanayakkara Weragoda Vidanalage Chamod Kanishka Chathuranga
Student ID : M23W0628

UI of the Web Page



Name: Nanayakkara Weragoda Vidanalage Chamod Kanishka Chathuranga | ID: M23W0628

13:23:11

Japanese National Blood Donation Program

Registration Form for New Donors

Personal Details

First name: First Name

Middle name (if you have): Middle Name

Last name: Last Name

Nationality: Choose...

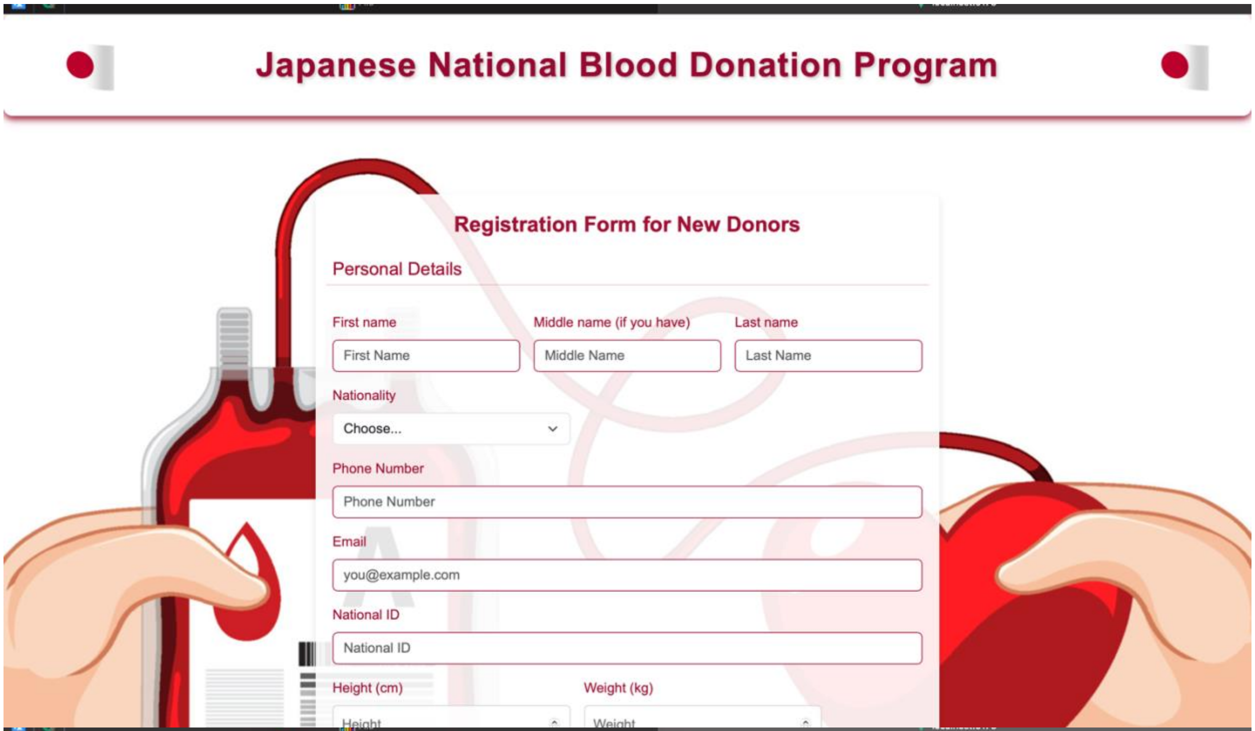
Phone Number: Phone Number

Email: you@example.com

National ID: National ID

Height (cm): Height

Weight (kg): Weight



Japanese National Blood Donation Program

Registration Form for New Donors

Personal Details

First name: First Name

Middle name (if you have): Middle Name

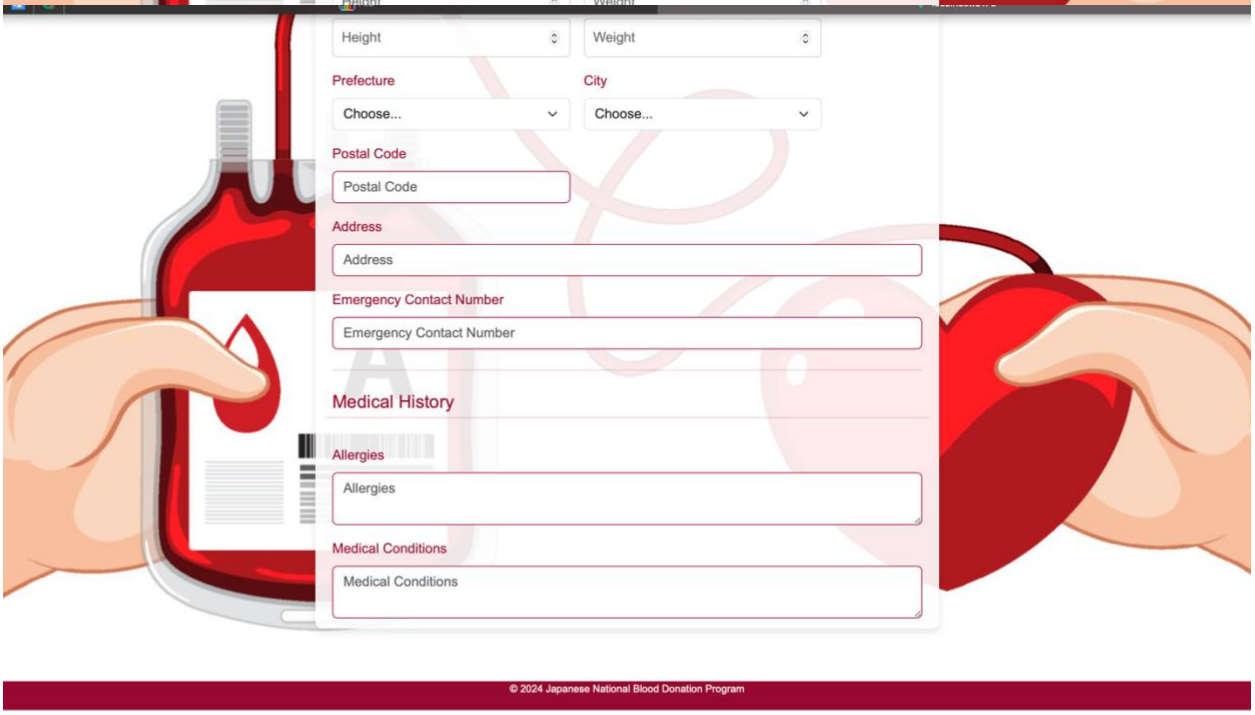
Last name: Last Name

Nationality: Choose...

Phone Number: Phone Number

Email: you@example.com

National ID: National ID



Height (cm): Height

Weight (kg): Weight

Prefecture: Choose...

City: Choose...

Postal Code: Postal Code

Address: Address

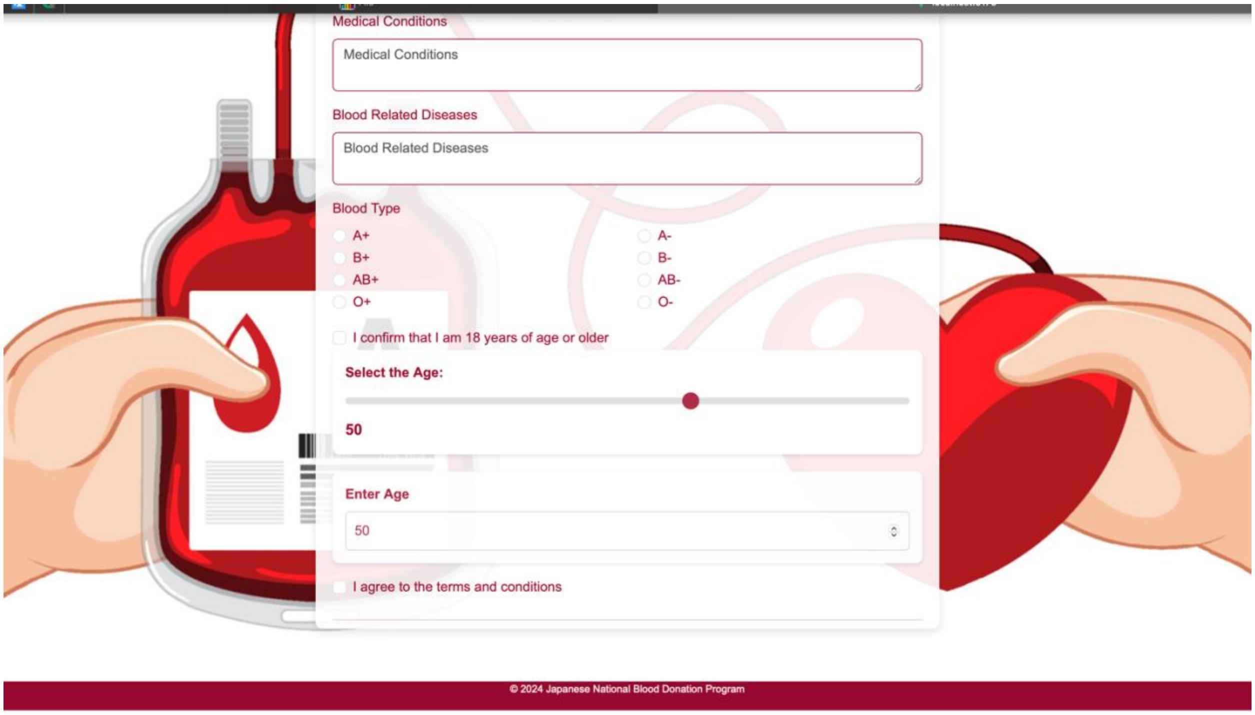
Emergency Contact Number: Emergency Contact Number

Medical History

Allergies: Allergies

Medical Conditions: Medical Conditions

© 2024 Japanese National Blood Donation Program



Medical Conditions

Medical Conditions

Blood Related Diseases

Blood Related Diseases

Blood Type

○ A+	○ A-
○ B+	○ B-
○ AB+	○ AB-
○ O+	○ O-

□ I confirm that I am 18 years of age or older

Select the Age:

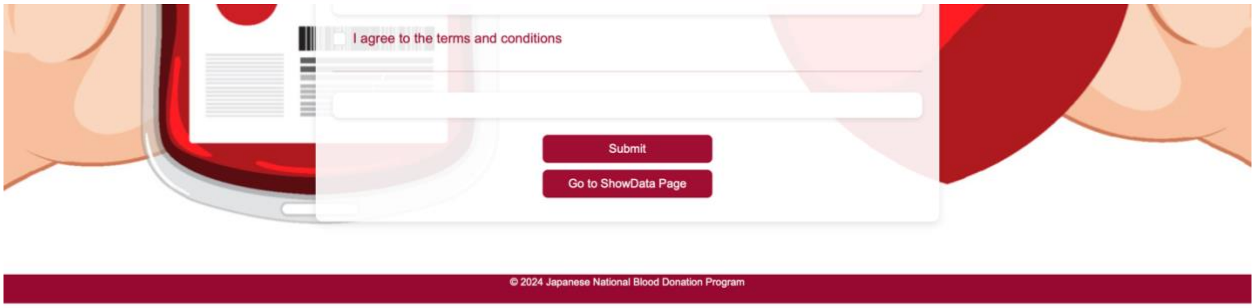
50

Enter Age

50

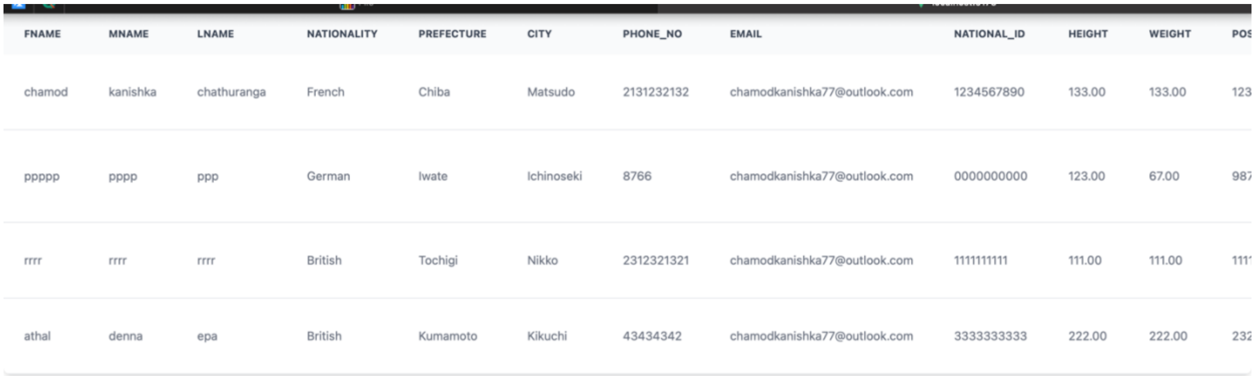
□ I agree to the terms and conditions

© 2024 Japanese National Blood Donation Program

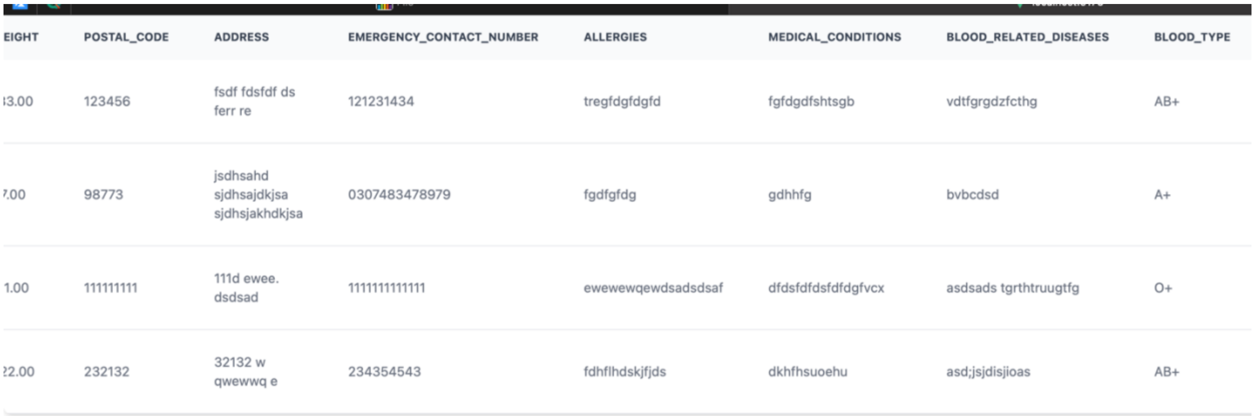


□ I agree to the terms and conditions

© 2024 Japanese National Blood Donation Program



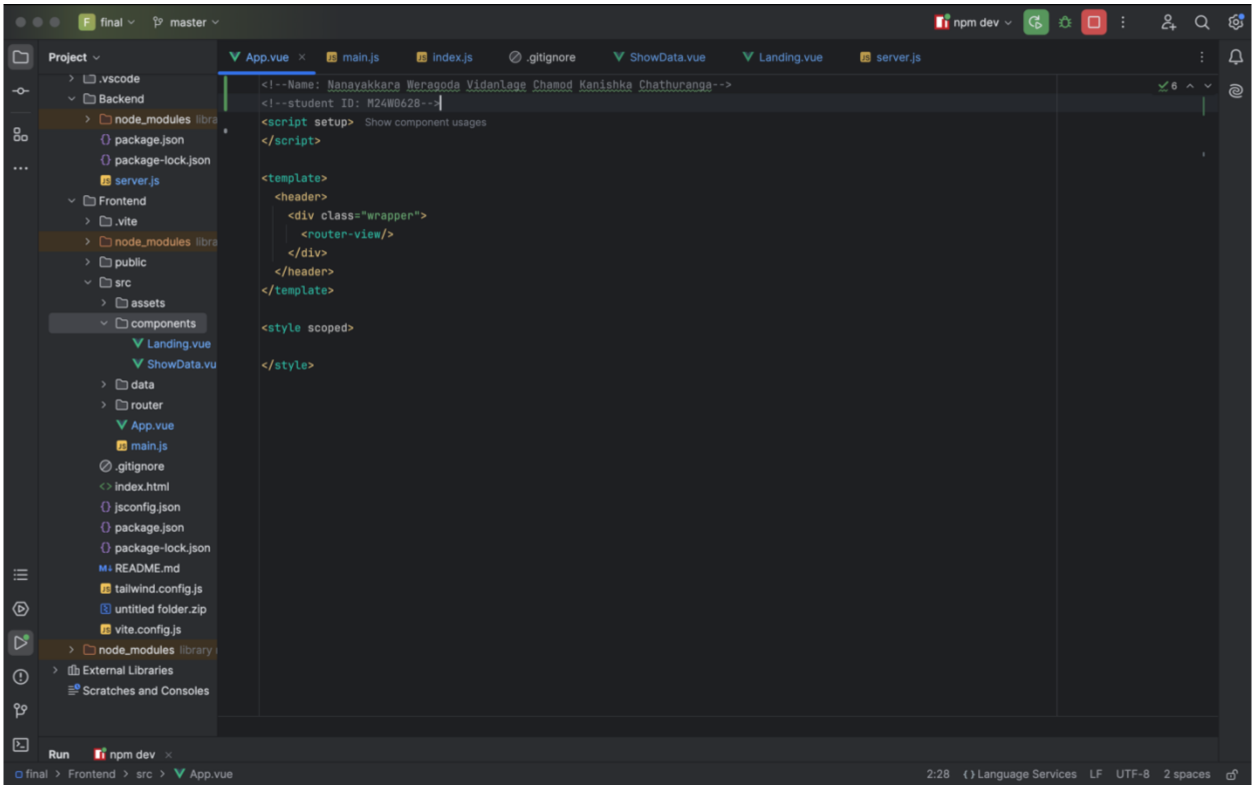
FNAME	MNAME	LNAME	NATIONALITY	PREFECTURE	CITY	PHONE_NO	EMAIL	NATIONAL_ID	HEIGHT	WEIGHT	POS
chamod	kanishka	chathuranga	French	Chiba	Matsudo	2131232132	chamodkanishka77@outlook.com	1234567890	133.00	133.00	123
ppppp	pppp	ppp	German	Iwate	Ichinoseki	8766	chamodkanishka77@outlook.com	0000000000	123.00	67.00	987
rrrr	rrrr	rrrr	British	Tochigi	Nikko	2312321321	chamodkanishka77@outlook.com	1111111111	111.00	111.00	1111
athal	denna	epa	British	Kumamoto	Kikuchi	43434342	chamodkanishka77@outlook.com	3333333333	222.00	222.00	232



EIGHT	POSTAL_CODE	ADDRESS	EMERGENCY_CONTACT_NUMBER	ALLERGIES	MEDICAL_CONDITIONS	BLOOD_RELATED_DISEASES	BLOOD_TYPE
3.00	123456	fsdf fdsfdf ds ferr re	121231434	tregfdgfdgfd	fgfdgdfshtsgb	vdtfgrgdzfcthg	AB+
7.00	98773	jsdhsahd sjdhsajdkjsa sjdhsjakhdkjsa	0307483478979	fgdfgfdg	gdhhfg	bvbcdsd	A+
1.00	111111111	111d ewee. dsdsad	1111111111111	ewewewqewdsadsdsaf	dfdsfdfdsfdfdgfvcx	asdsads tgrthtruugtfg	O+
22.00	232132	32132 w qwewwq e	234354543	fdhflhdskjfjds	dkhfhsuoehu	asd;jsjdisjioas	AB+

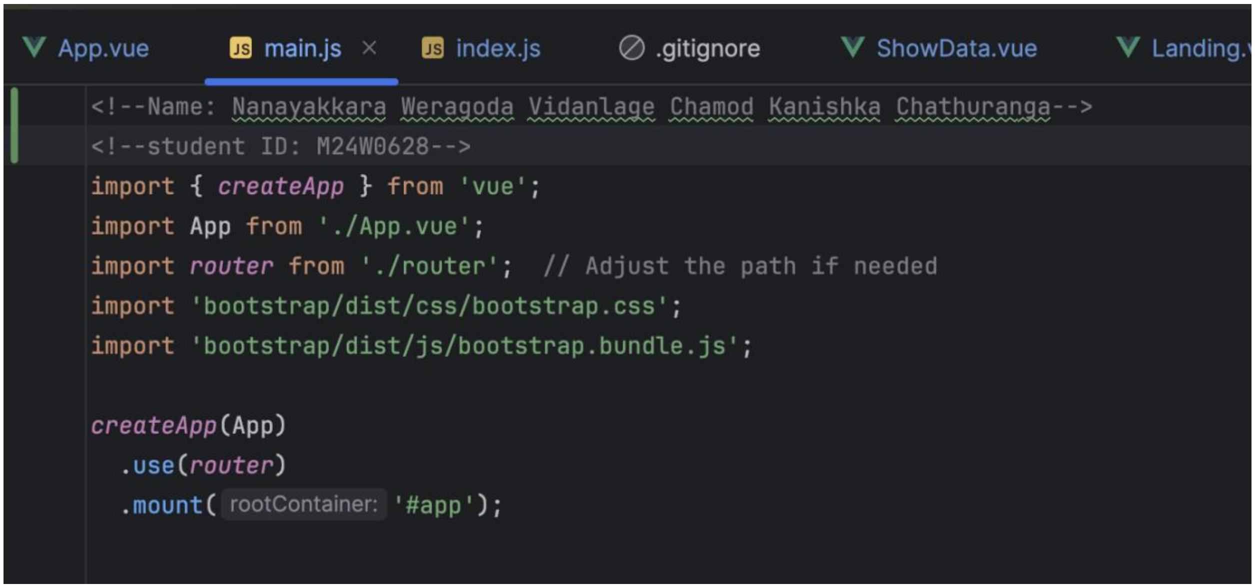
Code of the Frontend/ UI

App.vue



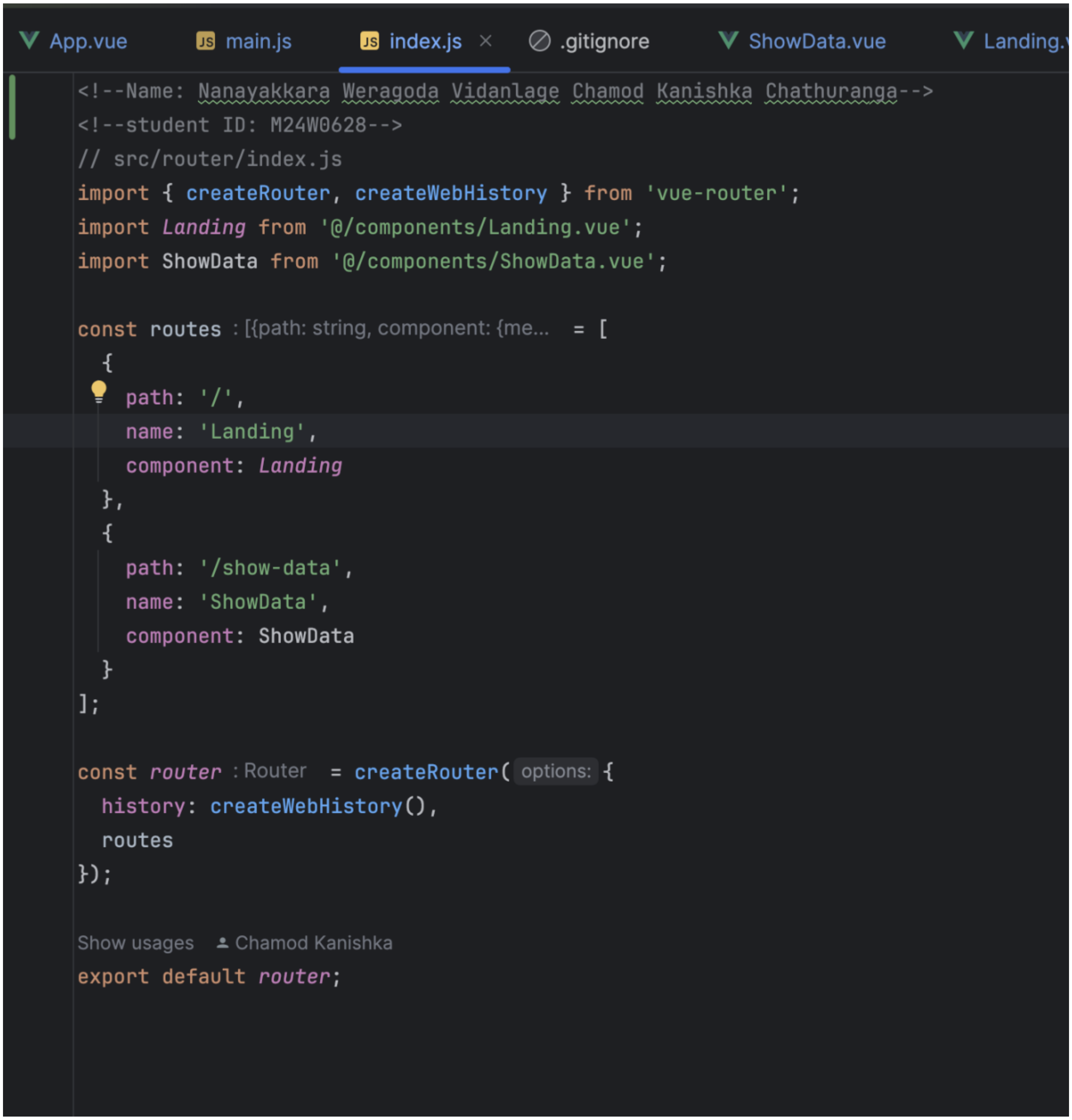
```
<!--Name: Nanayakkara Weragoda Vidanlage Chamod Kanishka Chathuranga-->
<!--student ID: M24W0628-->
<script setup>  Show component usages
</script>

<template>
  <header>
    <div class="wrapper">
      <router-view/>
    </div>
  </header>
</template>

<style scoped>

</style>
```

Main.js



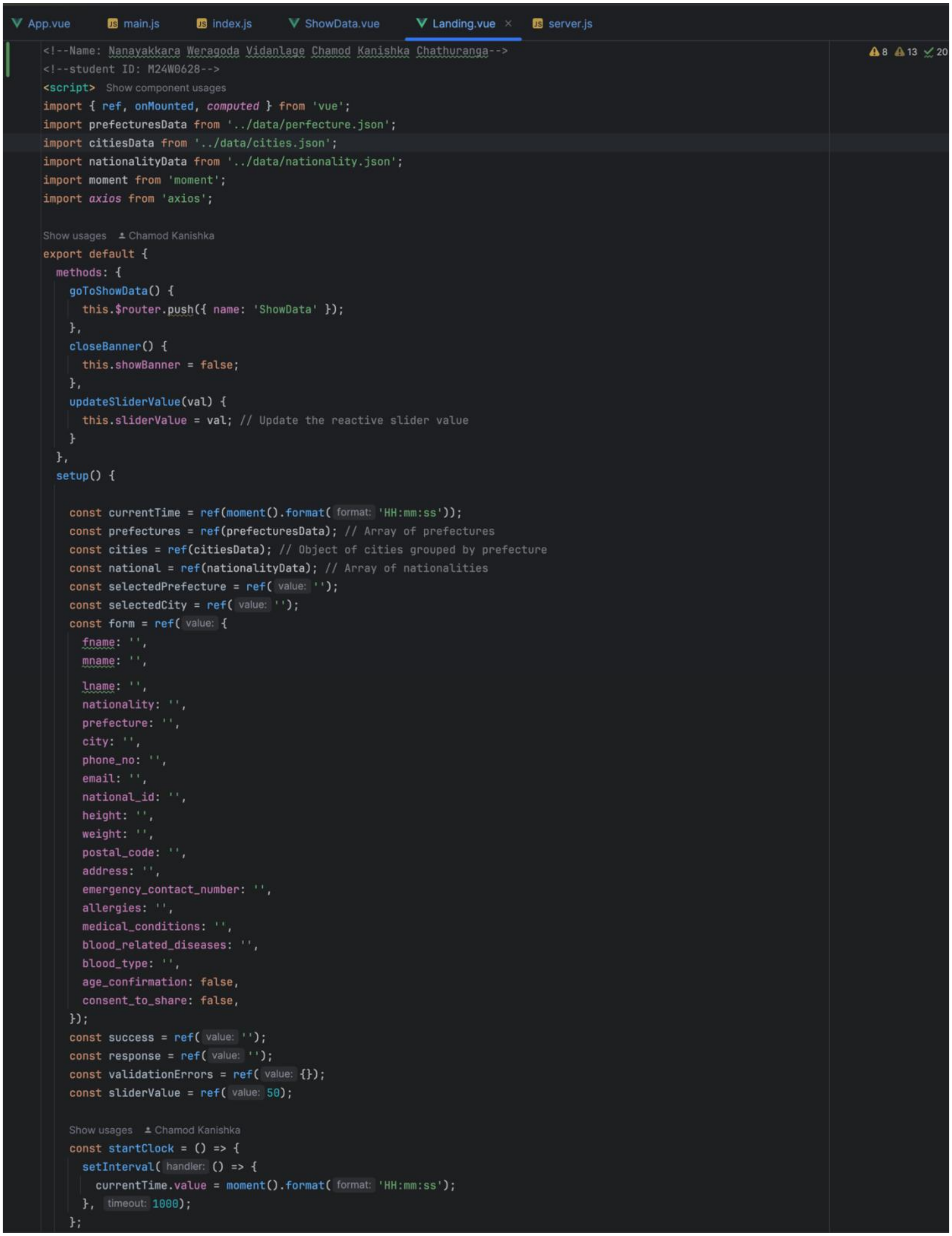
```
<!--Name: Nanayakkara Weragoda Vidanlage Chamod Kanishka Chathuranga-->
<!--student ID: M24W0628-->
import { createApp } from 'vue';
import App from './App.vue';
import router from './router';  // Adjust the path if needed
import 'bootstrap/dist/css/bootstrap.css';
import 'bootstrap/dist/js/bootstrap.bundle.js';

createApp(App)
  .use(router)
  .mount( rootContainer: '#app');
```

Index.js

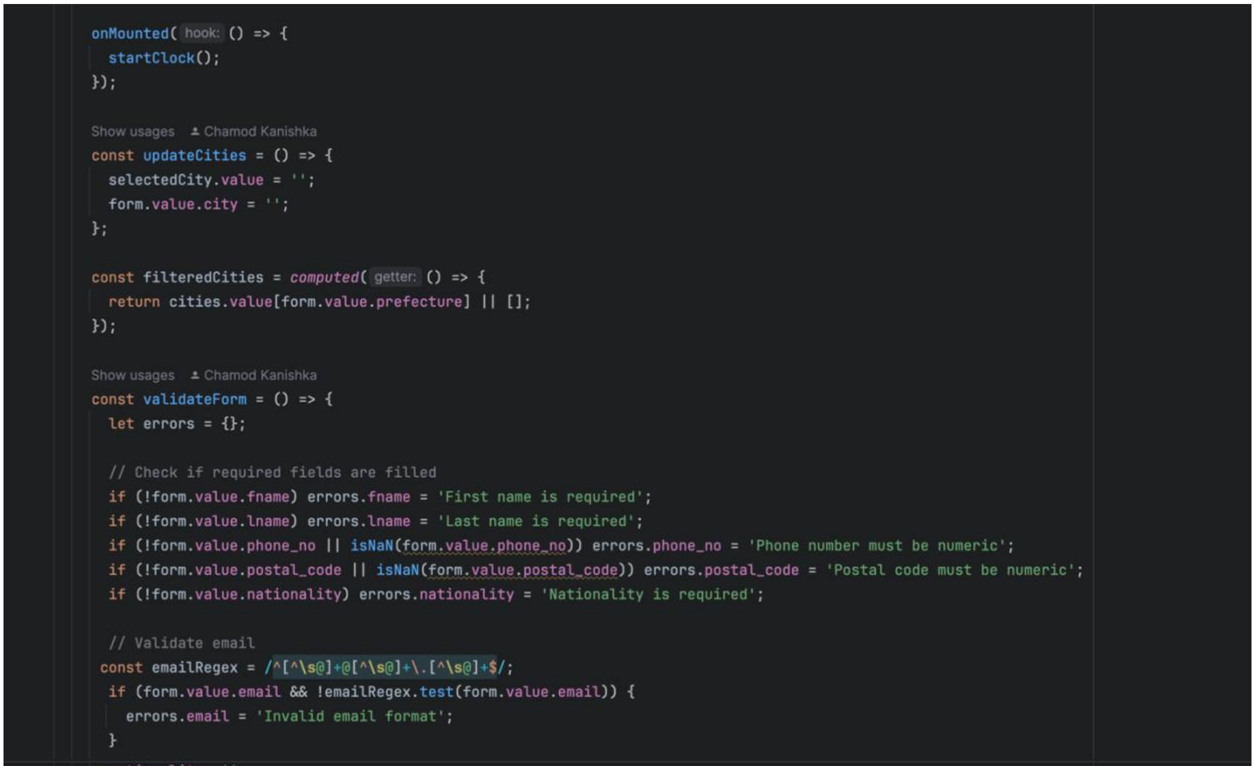


```
<!--Name: Nanayakkara Weragoda Vidanlage Chamod Kanishka Chathuranga-->
<!--student ID: M24W0628-->
// src/router/index.js
import { createRouter, createWebHistory } from 'vue-router';
import Landing from '@/components/Landing.vue';
import ShowData from '@/components/ShowData.vue';

const routes : [{path: string, component: {me...  = [
  {
    path: '/',
    name: 'Landing',
    component: Landing
  },
  {
    path: '/show-data',
    name: 'ShowData',
    component: ShowData
  }
];

const router : Router  = createRouter( options: {
  history: createWebHistory(),
  routes
});

Show usages   Chamod Kanishka
export default router;
```

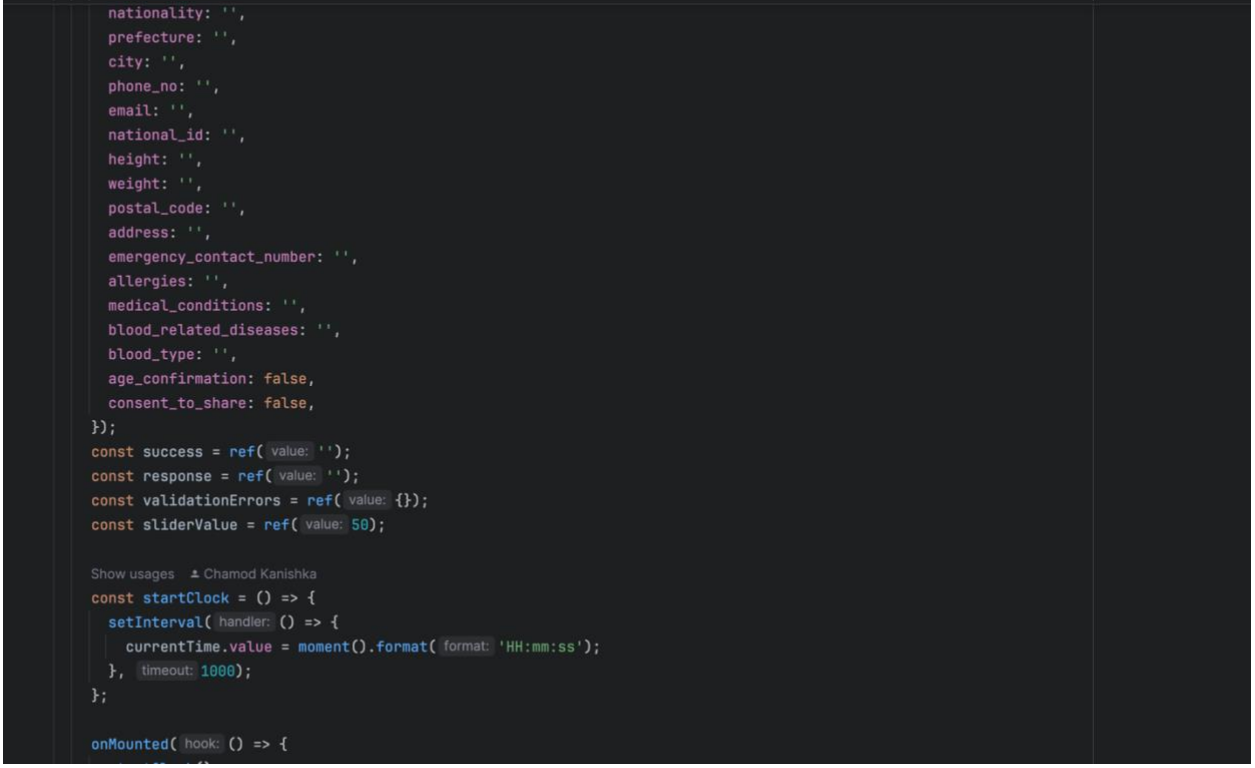
Landing.vue



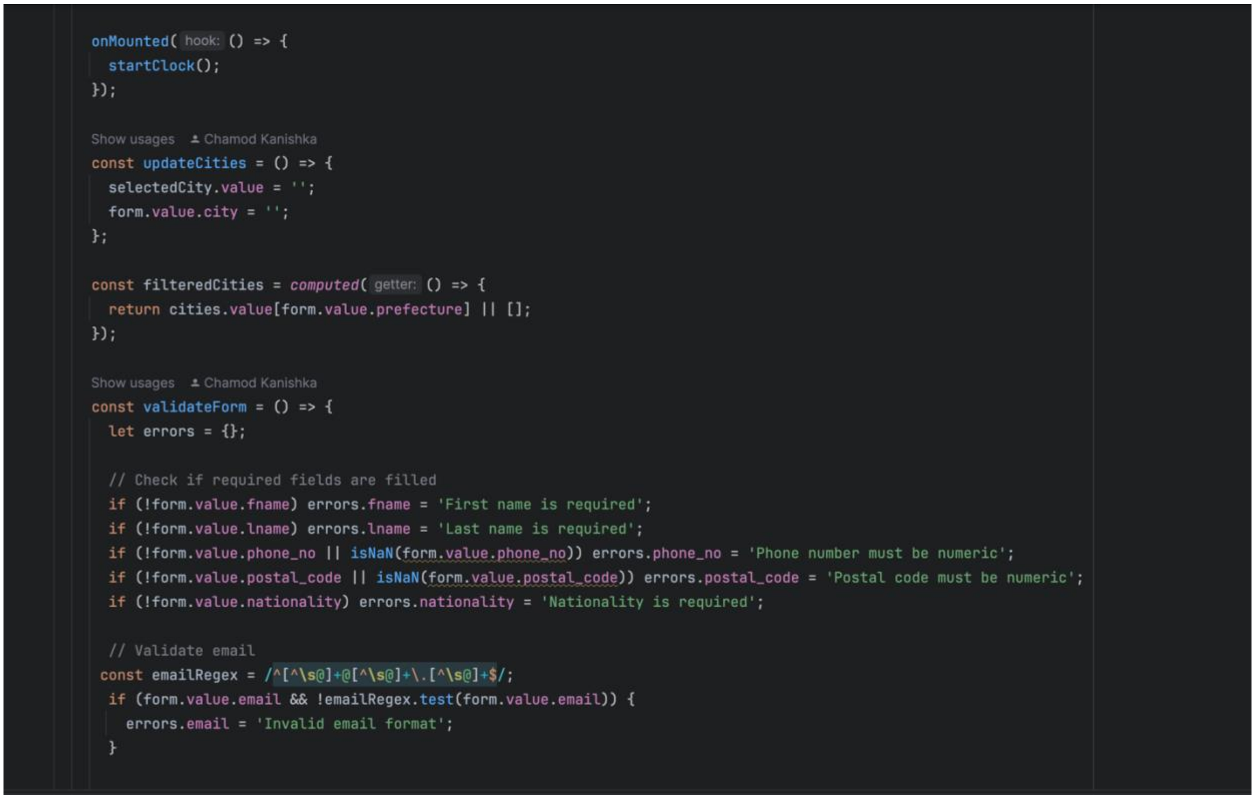
```
<!--Name: Nanayakkara Weragoda Vidanlage Chamod Kanishka Chathuranga-->
<!--student ID: M24W0628-->
<script>  Show component usages
import { ref, onMounted, computed } from 'vue';
import prefecturesData from '../data/perfecture.json';
import citiesData from '../data/cities.json';
import nationalityData from '../data/nationality.json';
import moment from 'moment';
import axios from 'axios';

Show usages   Chamod Kanishka
export default {
  methods: {
    goToShowData() {
      this.$router.push({ name: 'ShowData' });
    },
    closeBanner() {
      this.showBanner = false;
    },
    updateSliderValue(val) {
      this.sliderValue = val; // Update the reactive slider value
    }
  },
  setup() {

    const currentTime = ref(moment().format( format: 'HH:mm:ss'));
    const prefectures = ref(prefecturesData); // Array of prefectures
    const cities = ref(citiesData); // Object of cities grouped by prefecture
    const national = ref(nationalityData); // Array of nationalities
    const selectedPrefecture = ref( value: '');
    const selectedCity = ref( value: '');
    const form = ref( value: {
      fname: '',
      mname: '',
      lname: '',
      nationality: '',
      prefecture: '',
      city: '',
      phone_no: '',
      email: '',
      national_id: '',
      height: '',
      weight: '',
      postal_code: '',
      address: '',
      emergency_contact_number: '',
      allergies: '',
      medical_conditions: '',
      blood_related_diseases: '',
      blood_type: '',
      age_confirmation: false,
      consent_to_share: false,
    });
    const success = ref( value: '');
    const response = ref( value: '');
    const validationErrors = ref( value: {});
    const sliderValue = ref( value: 50);

    Show usages   Chamod Kanishka
    const startClock = () => {
      setInterval( handler: () => {
        currentTime.value = moment().format( format: 'HH:mm:ss');
      },  timeout: 1000);
    };
```



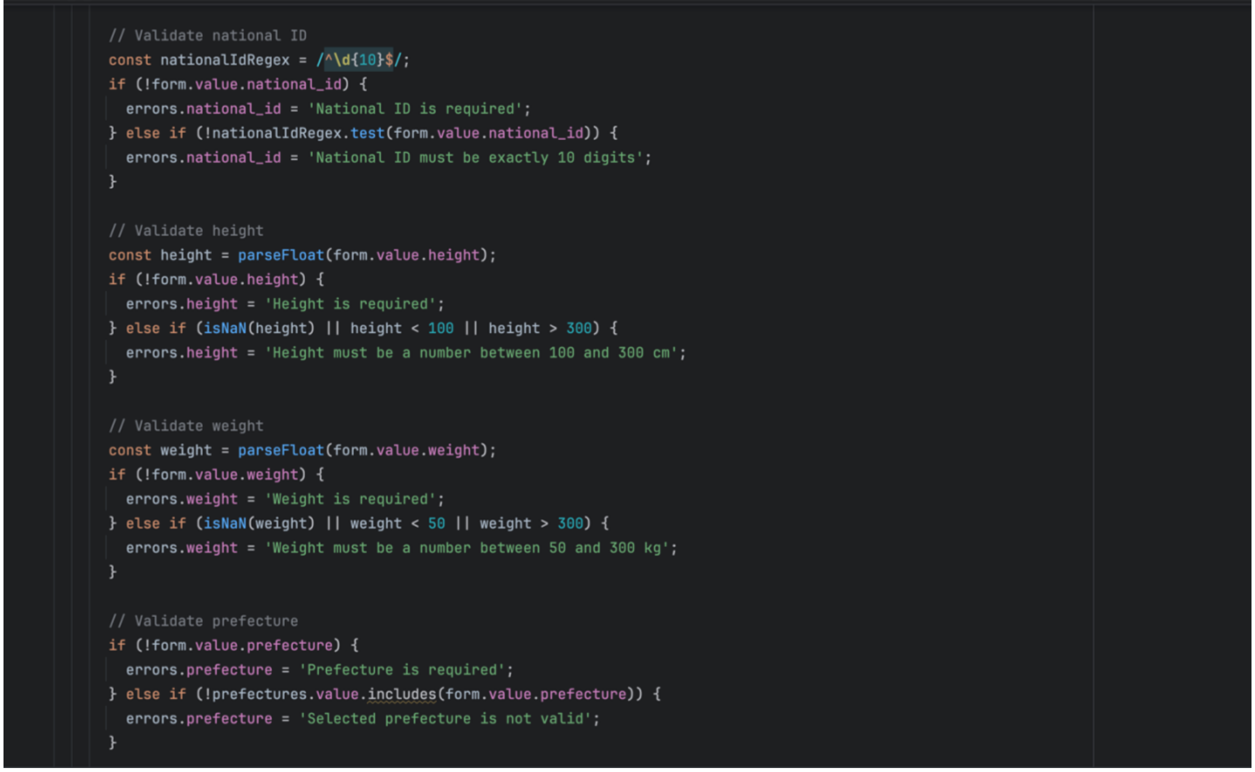
```
onMounted( hook: () => {
  startClock();
});

Show usages   Chamod Kanishka
const updateCities = () => {
  selectedCity.value = '';
  form.value.city = '';
};

const filteredCities = computed( getter: () => {
  return cities.value[form.value.prefecture] || [];
});

Show usages   Chamod Kanishka
const validateForm = () => {
  let errors = {};

  // Check if required fields are filled
  if (!form.value.fname) errors.fname = 'First name is required';
  if (!form.value.lname) errors.lname = 'Last name is required';
  if (!form.value.phone_no || isNaN(form.value.phone_no)) errors.phone_no = 'Phone number must be numeric';
  if (!form.value.postal_code || isNaN(form.value.postal_code)) errors.postal_code = 'Postal code must be numeric';
  if (!form.value.nationality) errors.nationality = 'Nationality is required';

  // Validate email
 const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (form.value.email && !emailRegex.test(form.value.email)) {
    errors.email = 'Invalid email format';
  }
```



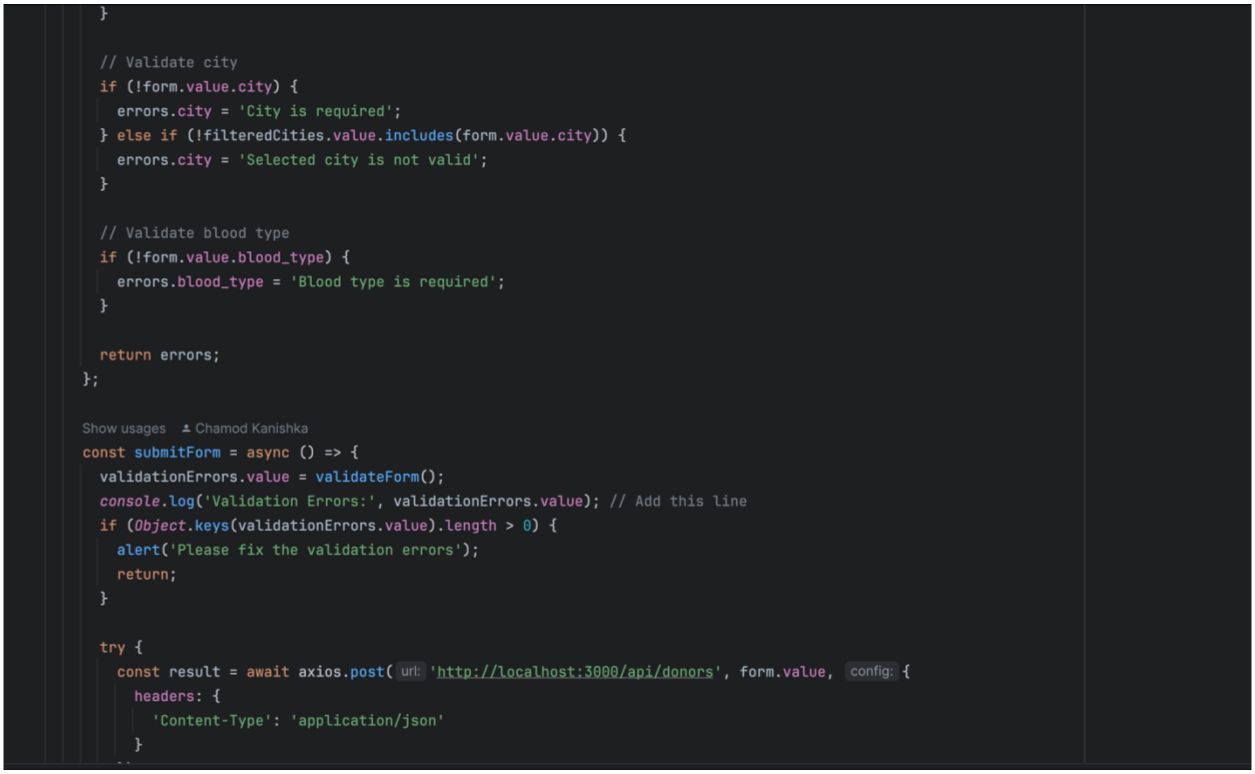
```
  nationality: '',
  prefecture: '',
  city: '',
  phone_no: '',
  email: '',
  national_id: '',
  height: '',
  weight: '',
  postal_code: '',
  address: '',
  emergency_contact_number: '',
  allergies: '',
  medical_conditions: '',
  blood_related_diseases: '',
  blood_type: '',
  age_confirmation: false,
  consent_to_share: false,
});
const success = ref( value: '');
const response = ref( value: '');
const validationErrors = ref( value: {});
const sliderValue = ref( value: 50);

Show usages   Chamod Kanishka
const startClock = () => {
  setInterval( handler: () => {
    currentTime.value = moment().format( format: 'HH:mm:ss');
  },  timeout: 1000);
};

onMounted( hook: () => {
```



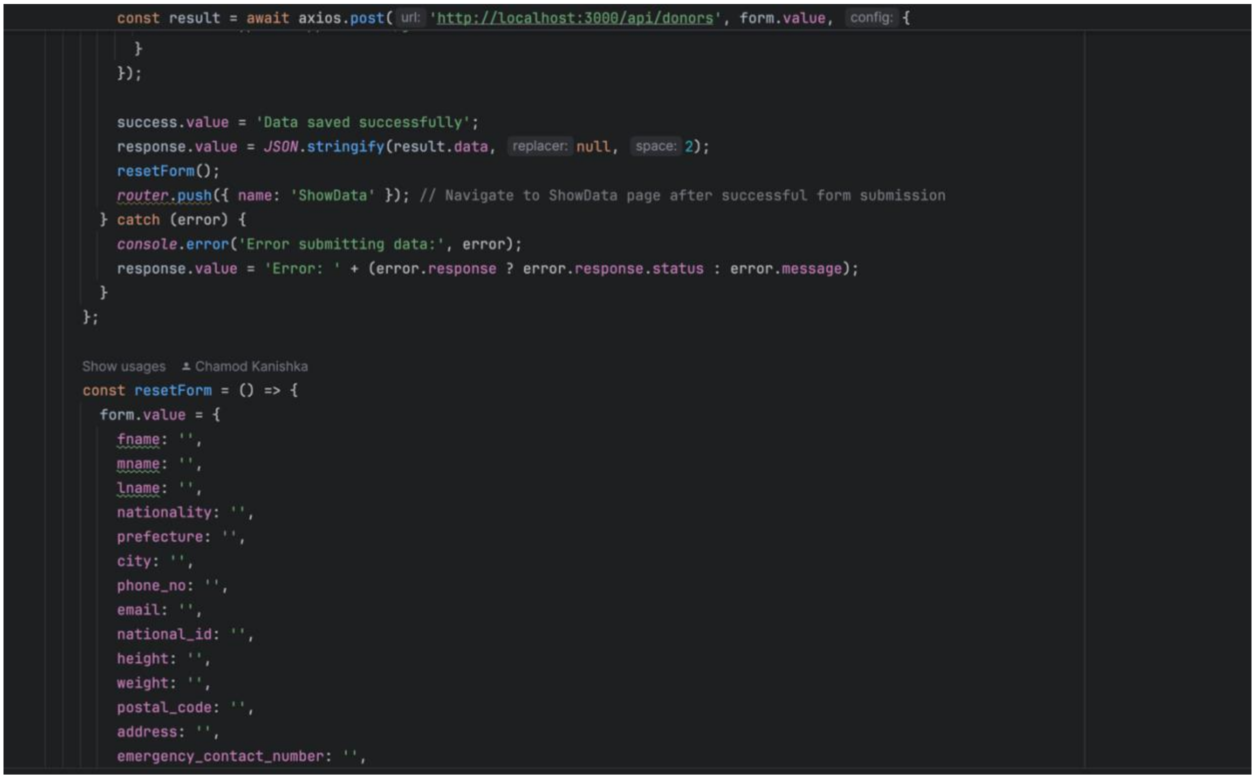
```
onMounted( hook: () => {
  startClock();
});

Show usages   Chamod Kanishka
const updateCities = () => {
  selectedCity.value = '';
  form.value.city = '';
};

const filteredCities = computed( getter: () => {
  return cities.value[form.value.prefecture] || [];
});

Show usages   Chamod Kanishka
const validateForm = () => {
  let errors = {};

  // Check if required fields are filled
  if (!form.value.fname) errors.fname = 'First name is required';
  if (!form.value.lname) errors.lname = 'Last name is required';
  if (!form.value.phone_no || isNaN(form.value.phone_no)) errors.phone_no = 'Phone number must be numeric';
  if (!form.value.postal_code || isNaN(form.value.postal_code)) errors.postal_code = 'Postal code must be numeric';
  if (!form.value.nationality) errors.nationality = 'Nationality is required';

  // Validate email
 const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (form.value.email && !emailRegex.test(form.value.email)) {
    errors.email = 'Invalid email format';
  }
```



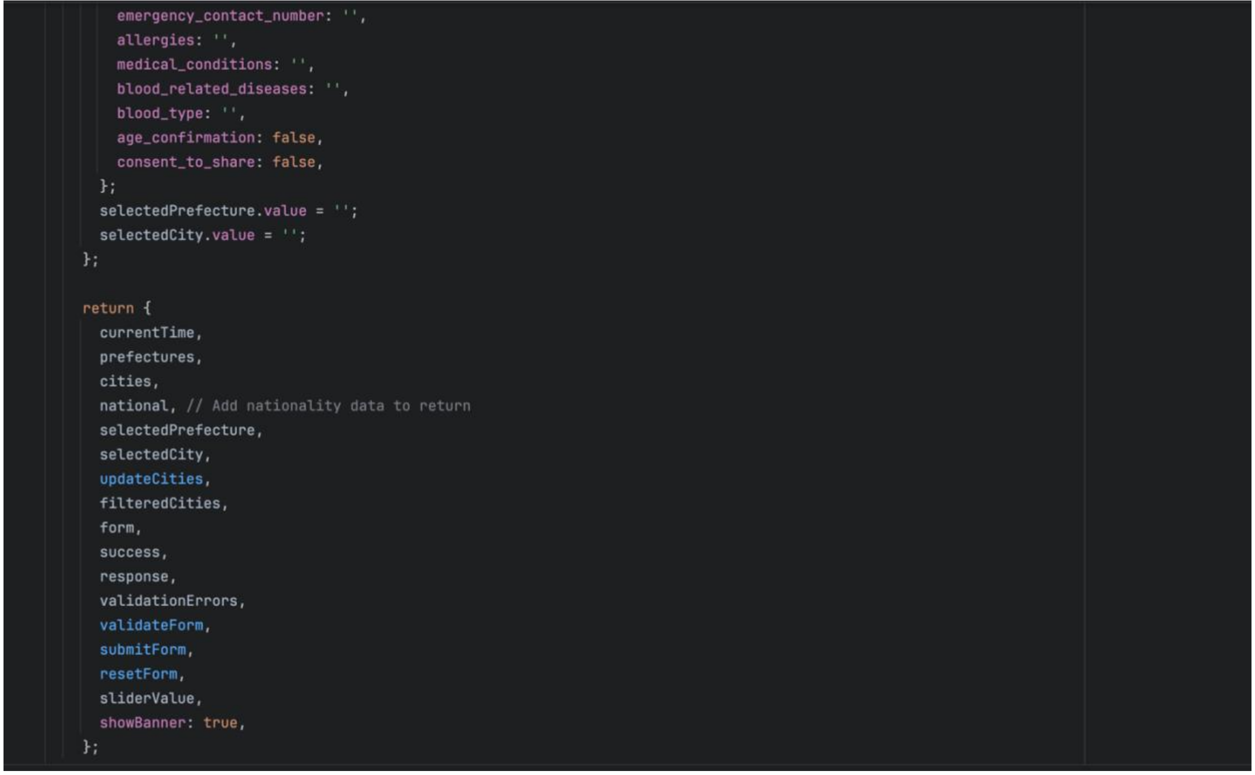
```
  // Validate national ID
  const nationalIdRegex = /^\d{10}$/;
  if (!form.value.national_id) {
    errors.national_id = 'National ID is required';
  } else if (!nationalIdRegex.test(form.value.national_id)) {
    errors.national_id = 'National ID must be exactly 10 digits';
  }

  // Validate height
  const height = parseFloat(form.value.height);
  if (!form.value.height) {
    errors.height = 'Height is required';
  } else if (isNaN(height) || height < 100 || height > 300) {
    errors.height = 'Height must be a number between 100 and 300 cm';
  }

  // Validate weight
  const weight = parseFloat(form.value.weight);
  if (!form.value.weight) {
    errors.weight = 'Weight is required';
  } else if (isNaN(weight) || weight < 50 || weight > 300) {
    errors.weight = 'Weight must be a number between 50 and 300 kg';
  }

  // Validate prefecture
  if (!form.value.prefecture) {
    errors.prefecture = 'Prefecture is required';
  } else if (!prefectures.value.includes(form.value.prefecture)) {
    errors.prefecture = 'Selected prefecture is not valid';
  }
```



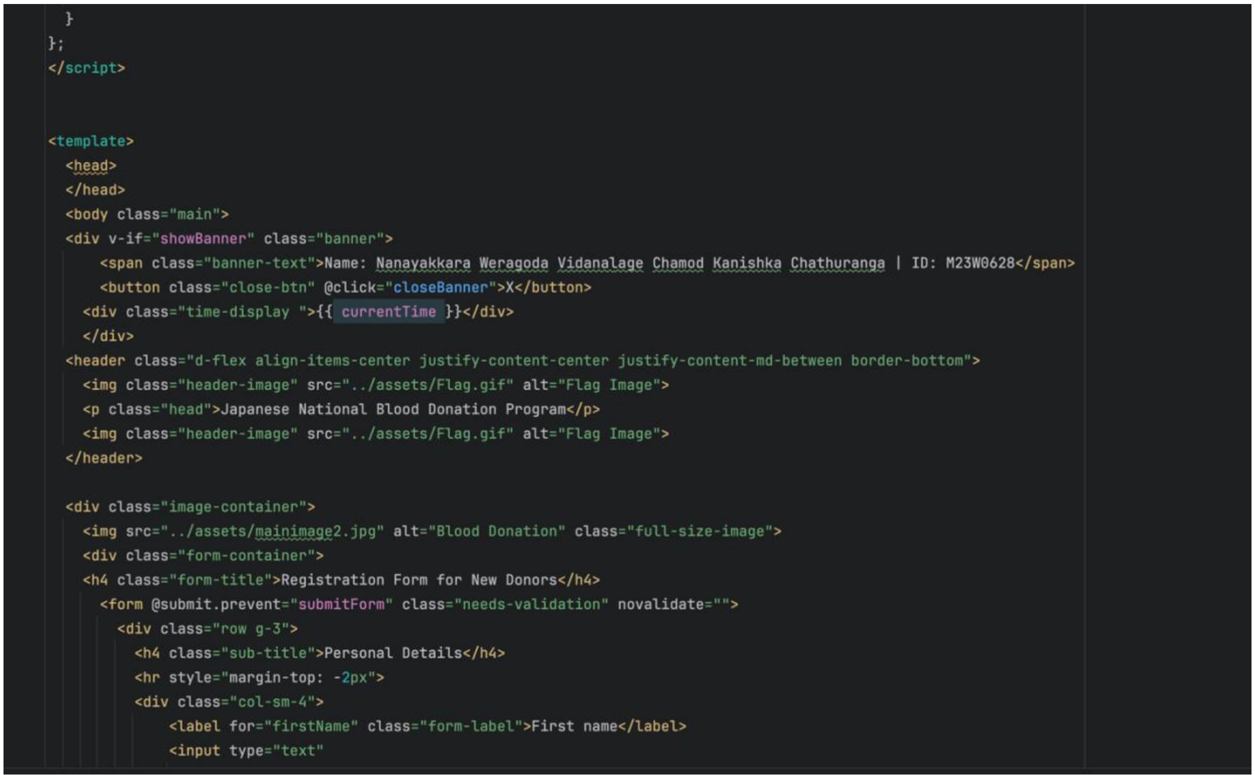
```
  }

  // Validate city
  if (!form.value.city) {
    errors.city = 'City is required';
  } else if (!filteredCities.value.includes(form.value.city)) {
    errors.city = 'Selected city is not valid';
  }

  // Validate blood type
  if (!form.value.blood_type) {
    errors.blood_type = 'Blood type is required';
  }

  return errors;
};

Show usages  Chamod Kanishka
const submitForm = async () => {
  validationErrors.value = validateForm();
  console.log('Validation Errors:', validationErrors.value); // Add this line
  if (Object.keys(validationErrors.value).length > 0) {
    alert('Please fix the validation errors');
    return;
  }

  try {
    const result = await axios.post( url: 'http://localhost:3000/api/donors', form.value,  config: {
      headers: {
        'Content-Type': 'application/json'
      }
```



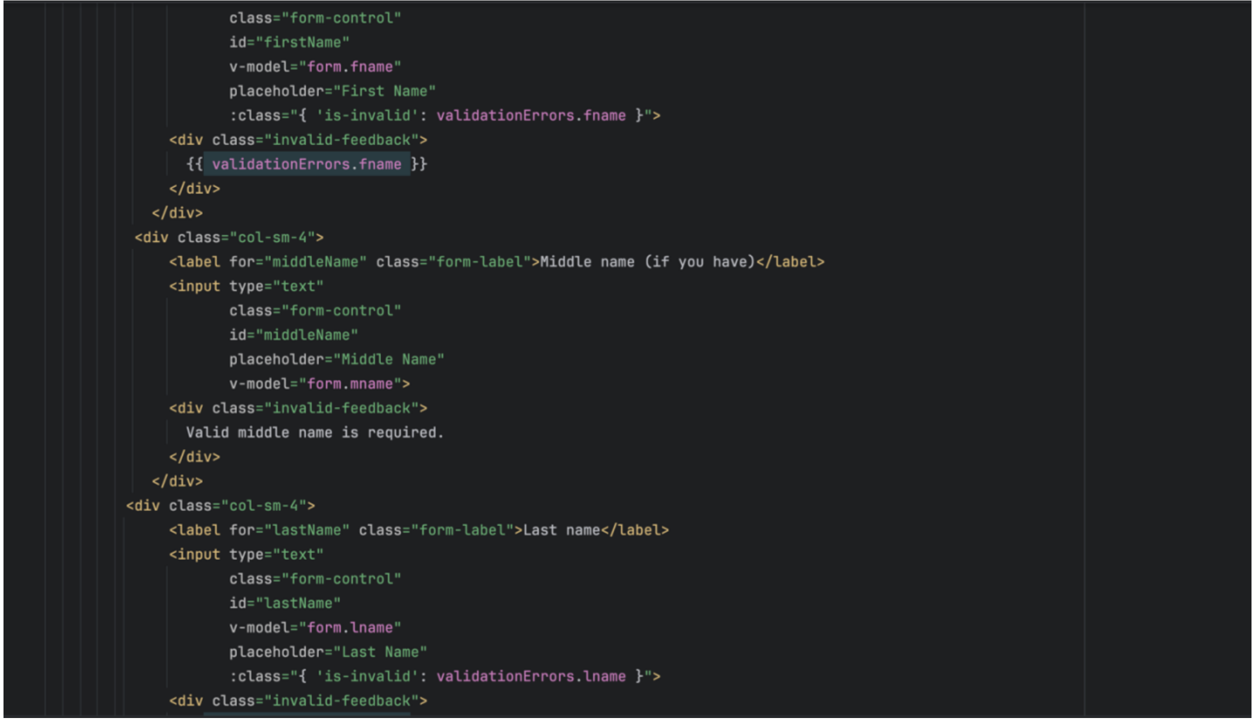
```
      const result = await axios.post( url: 'http://localhost:3000/api/donors', form.value,  config: {
        }
      });

      success.value = 'Data saved successfully';
      response.value = JSON.stringify(result.data,  replacer: null,  space: 2);
      resetForm();
      router.push({ name: 'ShowData' }); // Navigate to ShowData page after successful form submission
    } catch (error) {
      console.error('Error submitting data:', error);
      response.value = 'Error: ' + (error.response ? error.response.status : error.message);
    }
  };

  Show usages   Chamod Kanishka
  const resetForm = () => {
    form.value = {
      fname: '',
      mname: '',
      lname: '',
      nationality: '',
      prefecture: '',
      city: '',
      phone_no: '',
      email: '',
      national_id: '',
      height: '',
      weight: '',
      postal_code: '',
      address: '',
      emergency_contact_number: '',
```



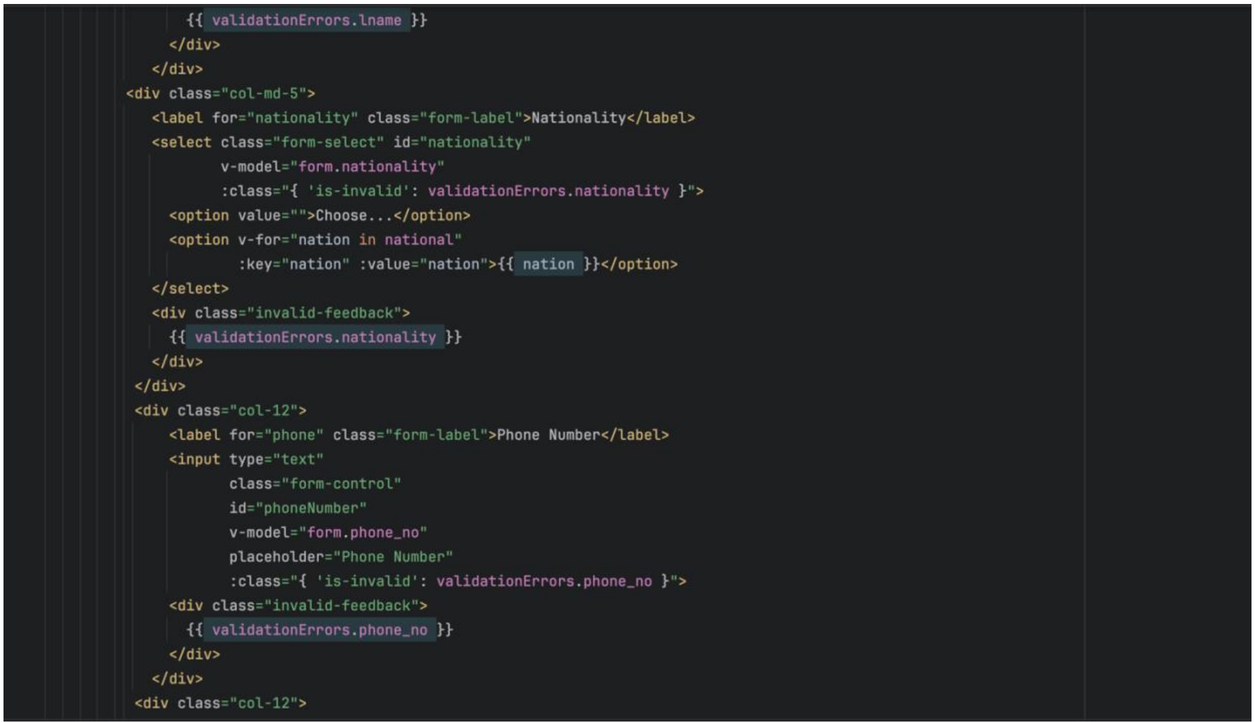
```
      emergency_contact_number: '',
      allergies: '',
      medical_conditions: '',
      blood_related_diseases: '',
      blood_type: '',
      age_confirmation: false,
      consent_to_share: false,
    };
    selectedPrefecture.value = '';
    selectedCity.value = '';
  };

  return {
    currentTime,
    prefectures,
    cities,
    national, // Add nationality data to return
    selectedPrefecture,
    selectedCity,
    updateCities,
    filteredCities,
    form,
    success,
    response,
    validationErrors,
    validateForm,
    submitForm,
    resetForm,
    sliderValue,
    showBanner: true,
  };
```



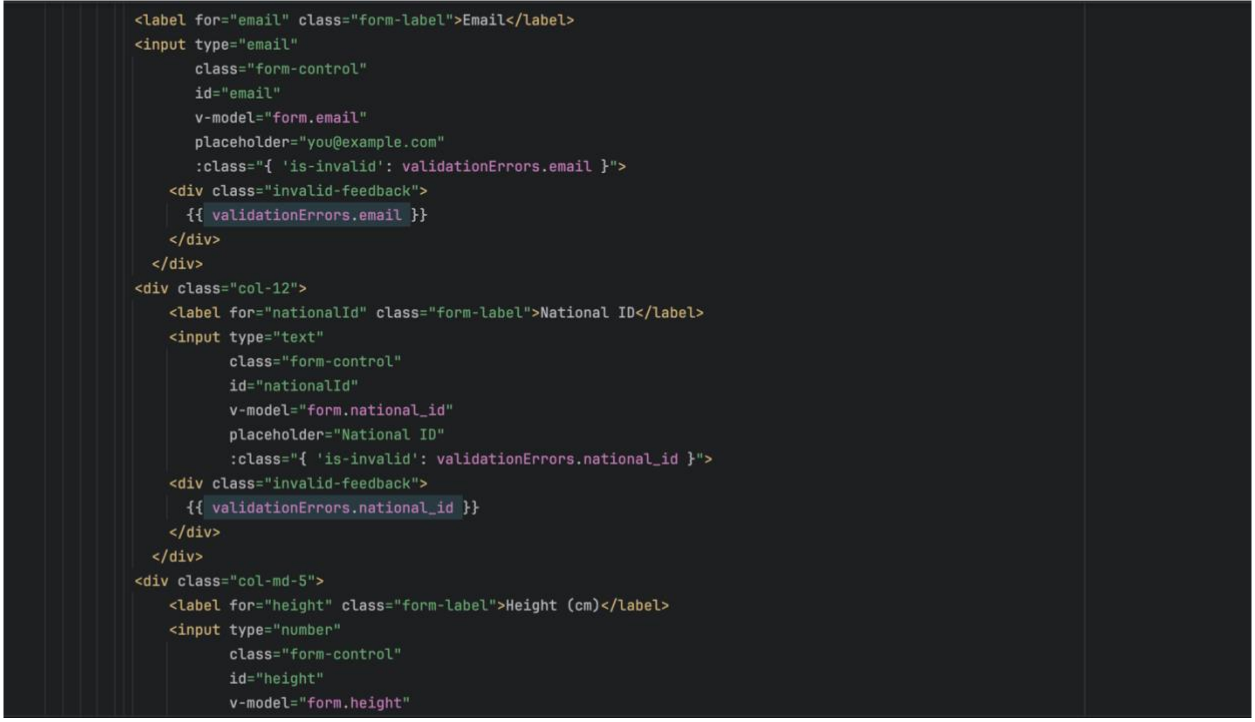
```
  }
};
</script>


<template>
  <head>
  </head>
  <body class="main">
  <div v-if="showBanner" class="banner">
      <span class="banner-text">Name: Nanayakkara Weragoda Vidanalage Chamod Kanishka Chathuranga | ID: M23W0628</span>
      <button class="close-btn" @click="closeBanner">X</button>
    <div class="time-display ">{{ currentTime }}</div>
    </div>
  <header class="d-flex align-items-center justify-content-center justify-content-md-between border-bottom">
    <img class="header-image" src="../assets/Flag.gif" alt="Flag Image">
    <p class="head">Japanese National Blood Donation Program</p>
    <img class="header-image" src="../assets/Flag.gif" alt="Flag Image">
  </header>

  <div class="image-container">
    <img src="../assets/mainimage2.jpg" alt="Blood Donation" class="full-size-image">
    <div class="form-container">
    <h4 class="form-title">Registration Form for New Donors</h4>
      <form @submit.prevent="submitForm" class="needs-validation" novalidate="">
        <div class="row g-3">
          <h4 class="sub-title">Personal Details</h4>
          <hr style="margin-top: -2px">
          <div class="col-sm-4">
              <label for="firstName" class="form-label">First name</label>
              <input type="text"
```



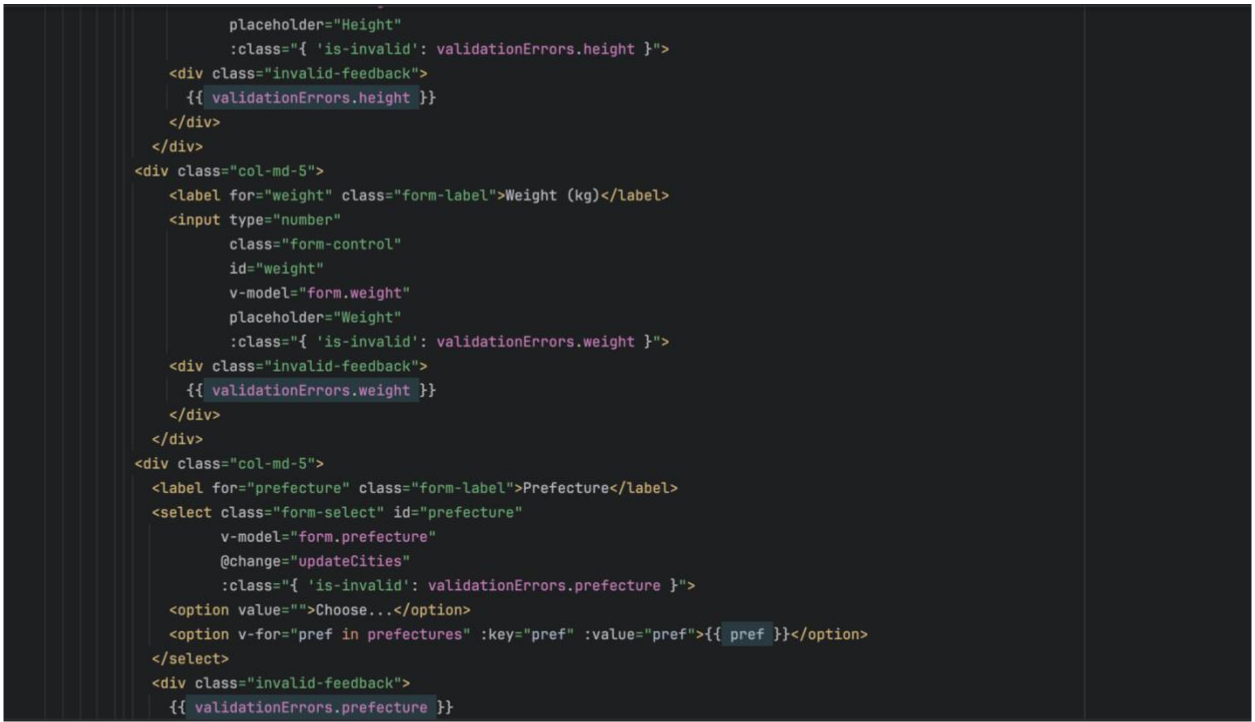
```
                     class="form-control"
                     id="firstName"
                     v-model="form.fname"
                     placeholder="First Name"
                     :class="{ 'is-invalid': validationErrors.fname }">
              <div class="invalid-feedback">
                {{ validationErrors.fname }}
              </div>
            </div>
          <div class="col-sm-4">
              <label for="middleName" class="form-label">Middle name (if you have)</label>
              <input type="text"
                     class="form-control"
                     id="middleName"
                     placeholder="Middle Name"
                     v-model="form.mname">
              <div class="invalid-feedback">
                Valid middle name is required.
              </div>
            </div>
         <div class="col-sm-4">
              <label for="lastName" class="form-label">Last name</label>
              <input type="text"
                     class="form-control"
                     id="lastName"
                     v-model="form.lname"
                     placeholder="Last Name"
                     :class="{ 'is-invalid': validationErrors.lname }">
              <div class="invalid-feedback">
```



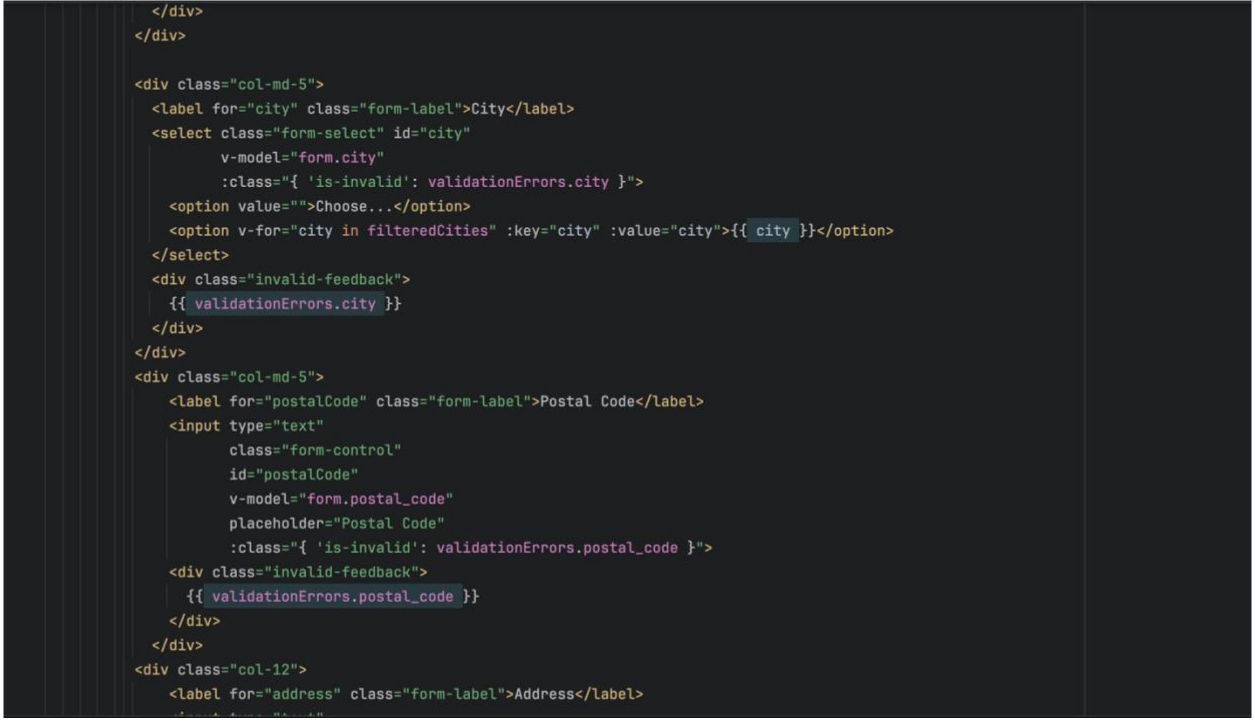
```
       {{ validationErrors.lname }}
     </div>
   </div>
<div class="col-md-5">
   <label for="nationality" class="form-label">Nationality</label>
   <select class="form-select" id="nationality"
           v-model="form.nationality"
           :class="{ 'is-invalid': validationErrors.nationality }">
     <option value="">Choose...</option>
     <option v-for="nation in national"
             :key="nation" :value="nation">{{ nation }}</option>
   </select>
   <div class="invalid-feedback">
     {{ validationErrors.nationality }}
   </div>
 </div>
 <div class="col-12">
     <label for="phone" class="form-label">Phone Number</label>
     <input type="text"
            class="form-control"
            id="phoneNumber"
            v-model="form.phone_no"
            placeholder="Phone Number"
            :class="{ 'is-invalid': validationErrors.phone_no }">
     <div class="invalid-feedback">
       {{ validationErrors.phone_no }}
     </div>
   </div>
 <div class="col-12">
```



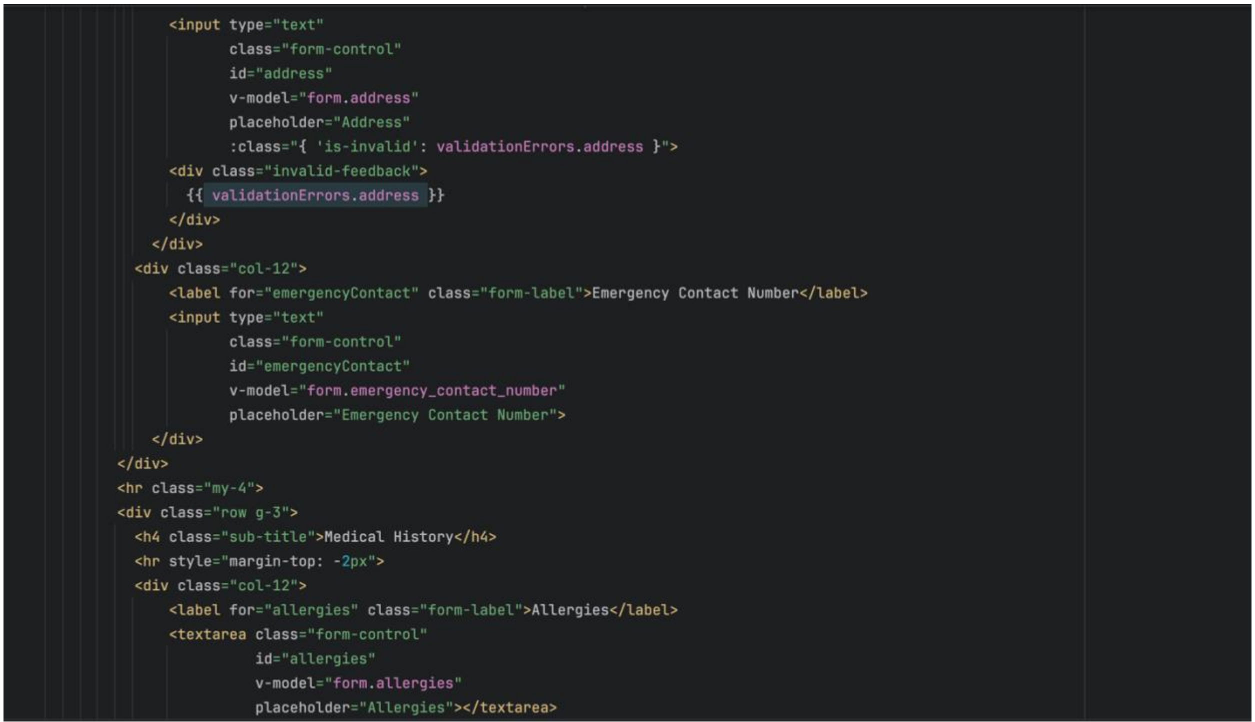
```
 <label for="email" class="form-label">Email</label>
 <input type="email"
        class="form-control"
        id="email"
        v-model="form.email"
        placeholder="you@example.com"
        :class="{ 'is-invalid': validationErrors.email }">
     <div class="invalid-feedback">
       {{ validationErrors.email }}
     </div>
   </div>
 <div class="col-12">
     <label for="nationalId" class="form-label">National ID</label>
     <input type="text"
            class="form-control"
            id="nationalId"
            v-model="form.national_id"
            placeholder="National ID"
            :class="{ 'is-invalid': validationErrors.national_id }">
     <div class="invalid-feedback">
       {{ validationErrors.national_id }}
     </div>
   </div>
 <div class="col-md-5">
     <label for="height" class="form-label">Height (cm)</label>
     <input type="number"
            class="form-control"
            id="height"
            v-model="form.height"
```



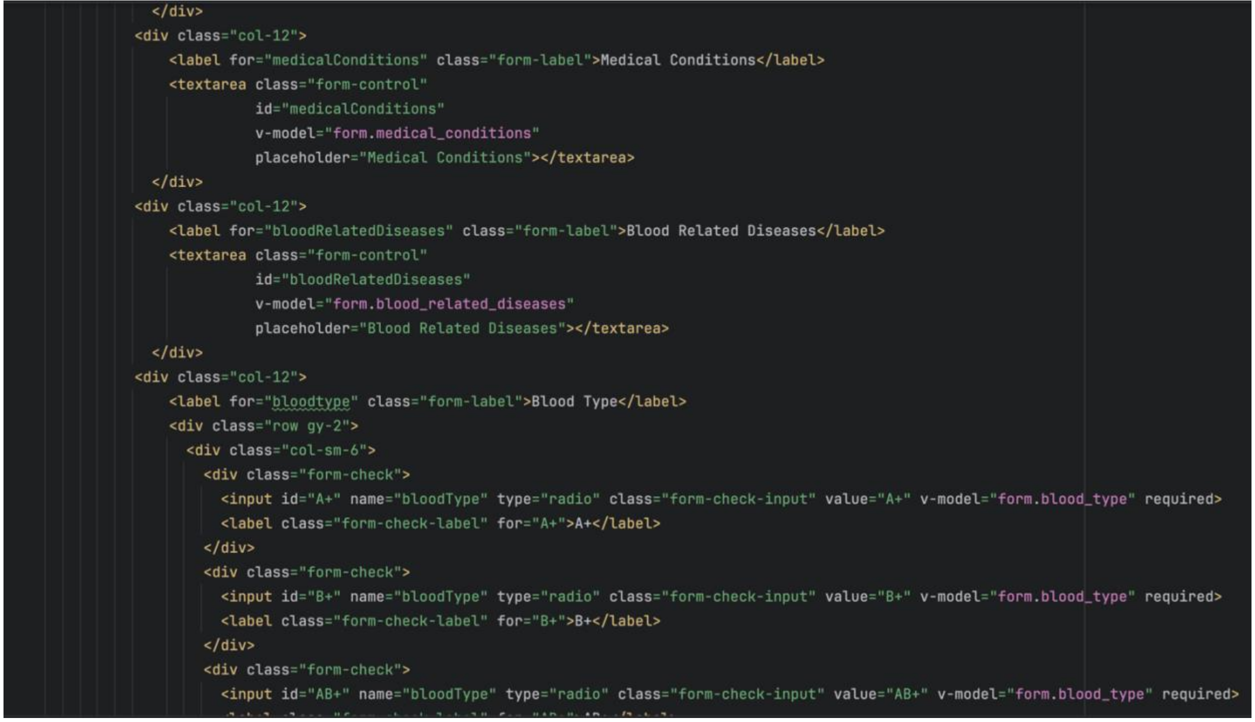
```
           placeholder="Height"
           :class="{ 'is-invalid': validationErrors.height }">
    <div class="invalid-feedback">
      {{ validationErrors.height }}
    </div>
  </div>
<div class="col-md-5">
    <label for="weight" class="form-label">Weight (kg)</label>
    <input type="number"
           class="form-control"
           id="weight"
           v-model="form.weight"
           placeholder="Weight"
           :class="{ 'is-invalid': validationErrors.weight }">
    <div class="invalid-feedback">
      {{ validationErrors.weight }}
    </div>
  </div>
<div class="col-md-5">
  <label for="prefecture" class="form-label">Prefecture</label>
  <select class="form-select" id="prefecture"
          v-model="form.prefecture"
          @change="updateCities"
          :class="{ 'is-invalid': validationErrors.prefecture }">
    <option value="">Choose...</option>
    <option v-for="pref in prefectures" :key="pref" :value="pref">{{ pref }}</option>
  </select>
  <div class="invalid-feedback">
    {{ validationErrors.prefecture }}
```



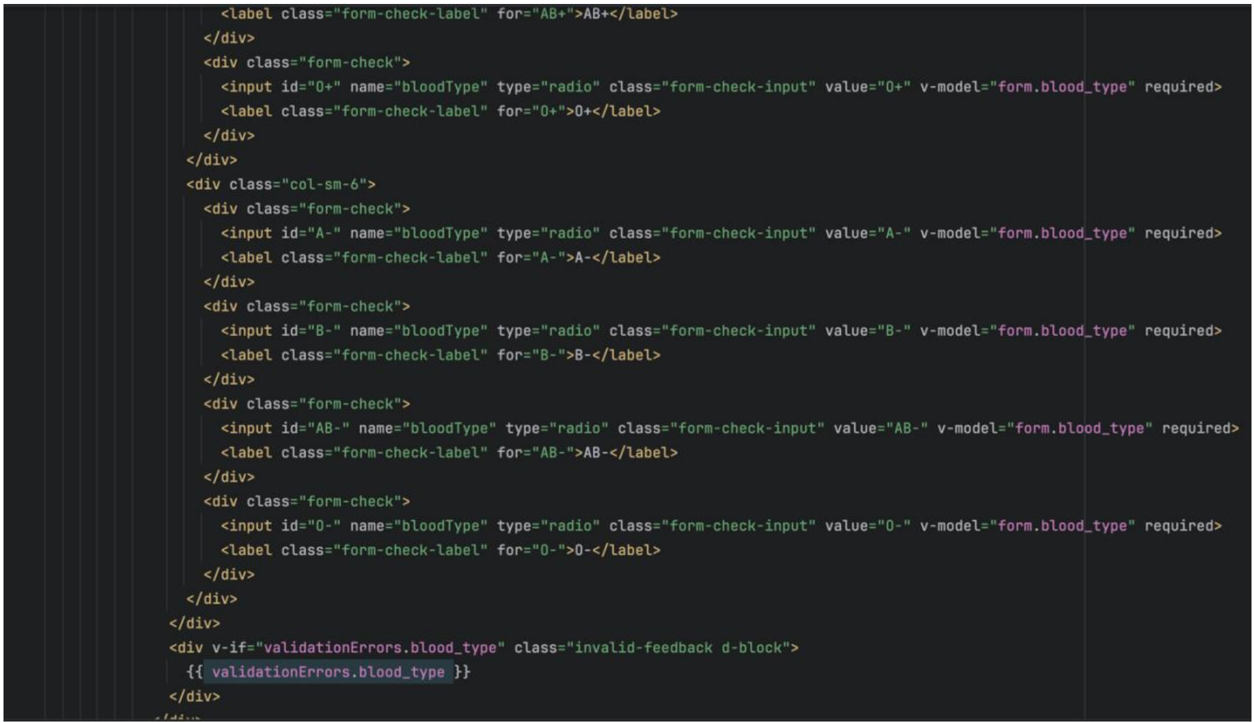
```
  </div>
</div>

<div class="col-md-5">
  <label for="city" class="form-label">City</label>
  <select class="form-select" id="city"
          v-model="form.city"
          :class="{ 'is-invalid': validationErrors.city }">
    <option value="">Choose...</option>
    <option v-for="city in filteredCities" :key="city" :value="city">{{ city }}</option>
  </select>
  <div class="invalid-feedback">
    {{ validationErrors.city }}
  </div>
</div>
<div class="col-md-5">
    <label for="postalCode" class="form-label">Postal Code</label>
    <input type="text"
           class="form-control"
           id="postalCode"
           v-model="form.postal_code"
           placeholder="Postal Code"
           :class="{ 'is-invalid': validationErrors.postal_code }">
    <div class="invalid-feedback">
      {{ validationErrors.postal_code }}
    </div>
  </div>
<div class="col-12">
    <label for="address" class="form-label">Address</label>
```



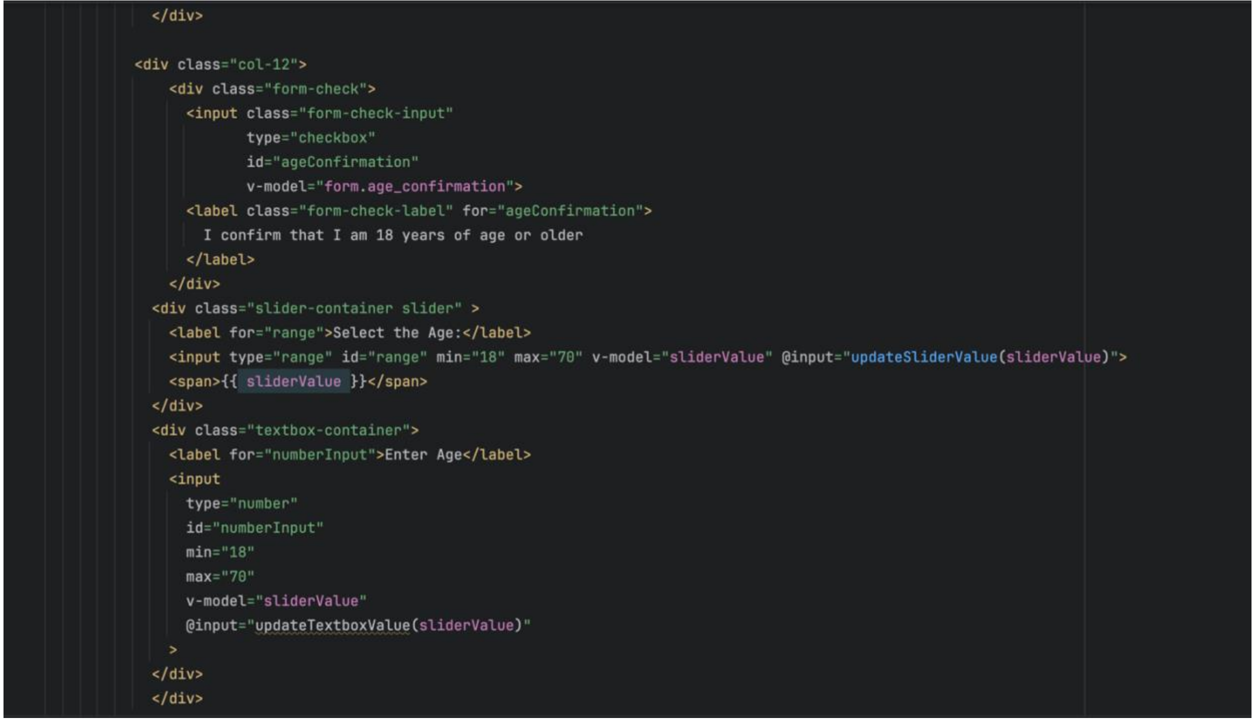
```
      <input type="text"
             class="form-control"
             id="address"
             v-model="form.address"
             placeholder="Address"
             :class="{ 'is-invalid': validationErrors.address }">
      <div class="invalid-feedback">
        {{ validationErrors.address }}
      </div>
    </div>
  <div class="col-12">
      <label for="emergencyContact" class="form-label">Emergency Contact Number</label>
      <input type="text"
             class="form-control"
             id="emergencyContact"
             v-model="form.emergency_contact_number"
             placeholder="Emergency Contact Number">
    </div>
</div>
<hr class="my-4">
<div class="row g-3">
  <h4 class="sub-title">Medical History</h4>
  <hr style="margin-top: -2px">
  <div class="col-12">
      <label for="allergies" class="form-label">Allergies</label>
      <textarea class="form-control"
                id="allergies"
                v-model="form.allergies"
                placeholder="Allergies"></textarea>
```



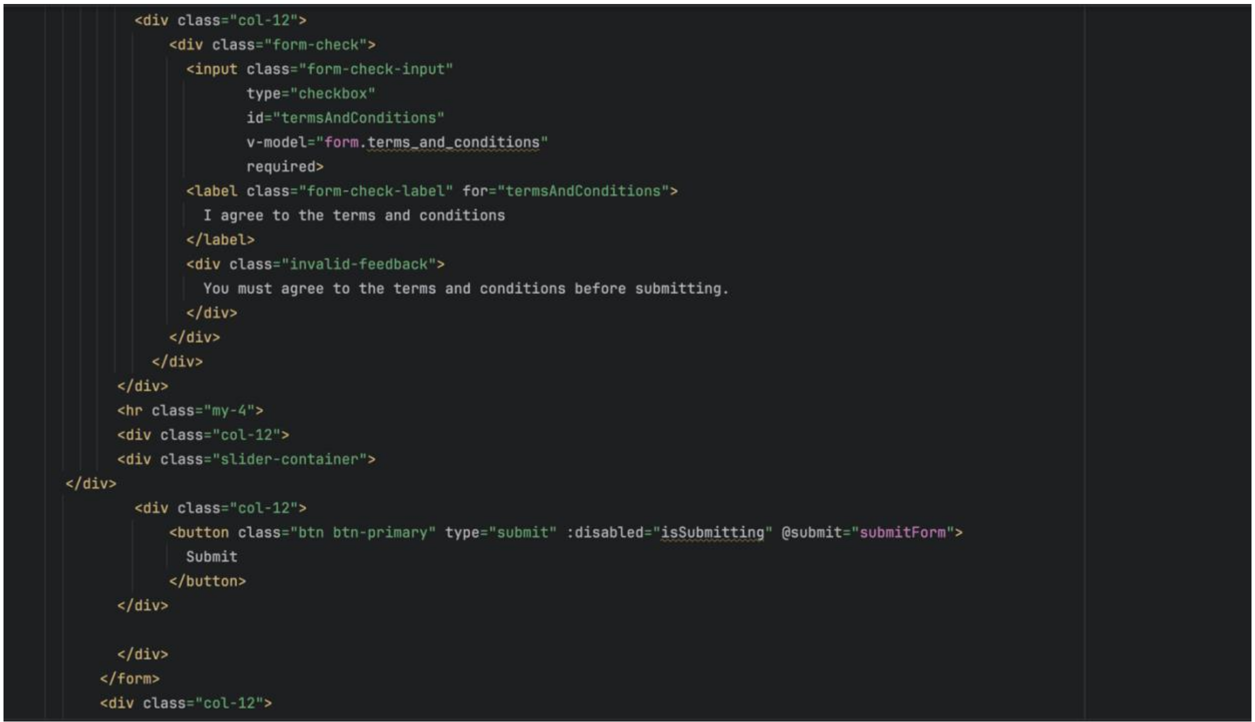
```
    </div>
  <div class="col-12">
      <label for="medicalConditions" class="form-label">Medical Conditions</label>
      <textarea class="form-control"
                id="medicalConditions"
                v-model="form.medical_conditions"
                placeholder="Medical Conditions"></textarea>
    </div>
  <div class="col-12">
      <label for="bloodRelatedDiseases" class="form-label">Blood Related Diseases</label>
      <textarea class="form-control"
                id="bloodRelatedDiseases"
                v-model="form.blood_related_diseases"
                placeholder="Blood Related Diseases"></textarea>
    </div>
  <div class="col-12">
      <label for="bloodtype" class="form-label">Blood Type</label>
      <div class="row gy-2">
        <div class="col-sm-6">
          <div class="form-check">
            <input id="A+" name="bloodType" type="radio" class="form-check-input" value="A+" v-model="form.blood_type" required>
            <label class="form-check-label" for="A+">A+</label>
          </div>
          <div class="form-check">
            <input id="B+" name="bloodType" type="radio" class="form-check-input" value="B+" v-model="form.blood_type" required>
            <label class="form-check-label" for="B+">B+</label>
          </div>
          <div class="form-check">
            <input id="AB+" name="bloodType" type="radio" class="form-check-input" value="AB+" v-model="form.blood_type" required>
            <label class="form-check-label" for="AB+">AB+</label>
```



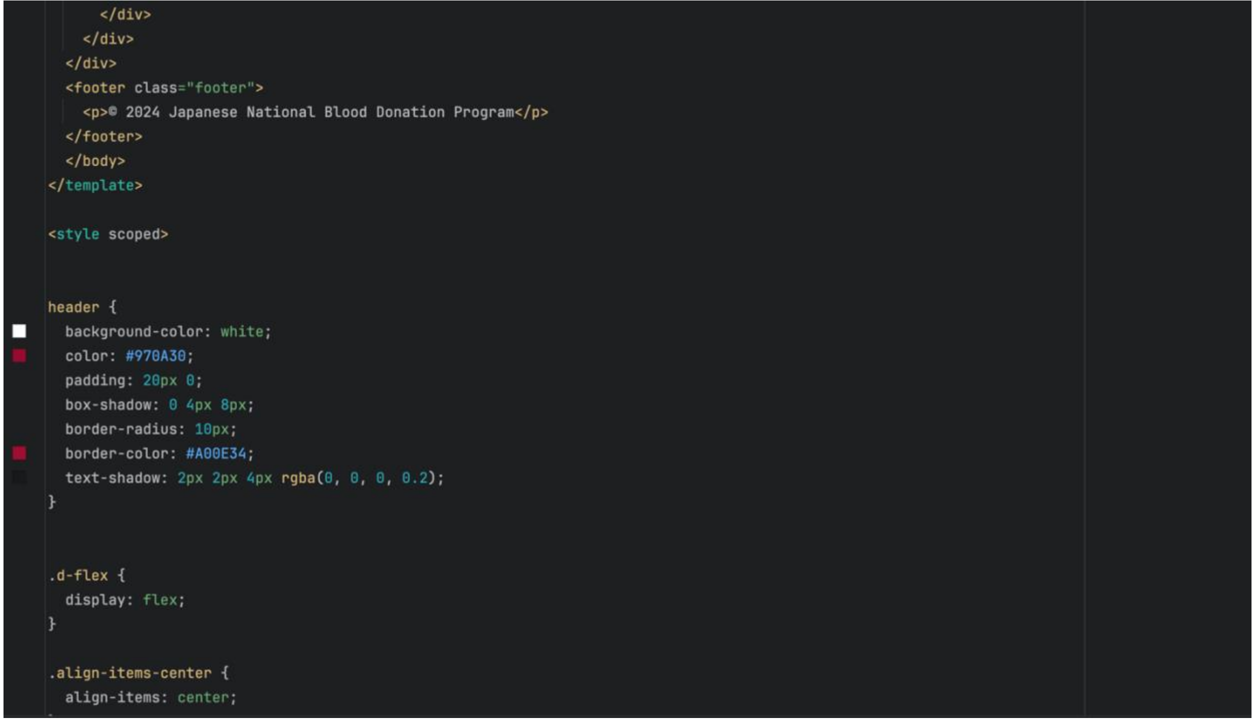
```
          <label class="form-check-label" for="AB+">AB+</label>
        </div>
        <div class="form-check">
          <input id="O+" name="bloodType" type="radio" class="form-check-input" value="O+" v-model="form.blood_type" required>
          <label class="form-check-label" for="O+">O+</label>
        </div>
      </div>
      <div class="col-sm-6">
        <div class="form-check">
          <input id="A-" name="bloodType" type="radio" class="form-check-input" value="A-" v-model="form.blood_type" required>
          <label class="form-check-label" for="A-">A-</label>
        </div>
        <div class="form-check">
          <input id="B-" name="bloodType" type="radio" class="form-check-input" value="B-" v-model="form.blood_type" required>
          <label class="form-check-label" for="B-">B-</label>
        </div>
        <div class="form-check">
          <input id="AB-" name="bloodType" type="radio" class="form-check-input" value="AB-" v-model="form.blood_type" required>
          <label class="form-check-label" for="AB-">AB-</label>
        </div>
        <div class="form-check">
          <input id="O-" name="bloodType" type="radio" class="form-check-input" value="O-" v-model="form.blood_type" required>
          <label class="form-check-label" for="O-">O-</label>
        </div>
      </div>
    </div>
    <div v-if="validationErrors.blood_type" class="invalid-feedback d-block">
      {{ validationErrors.blood_type }}
    </div>
  </div>
```



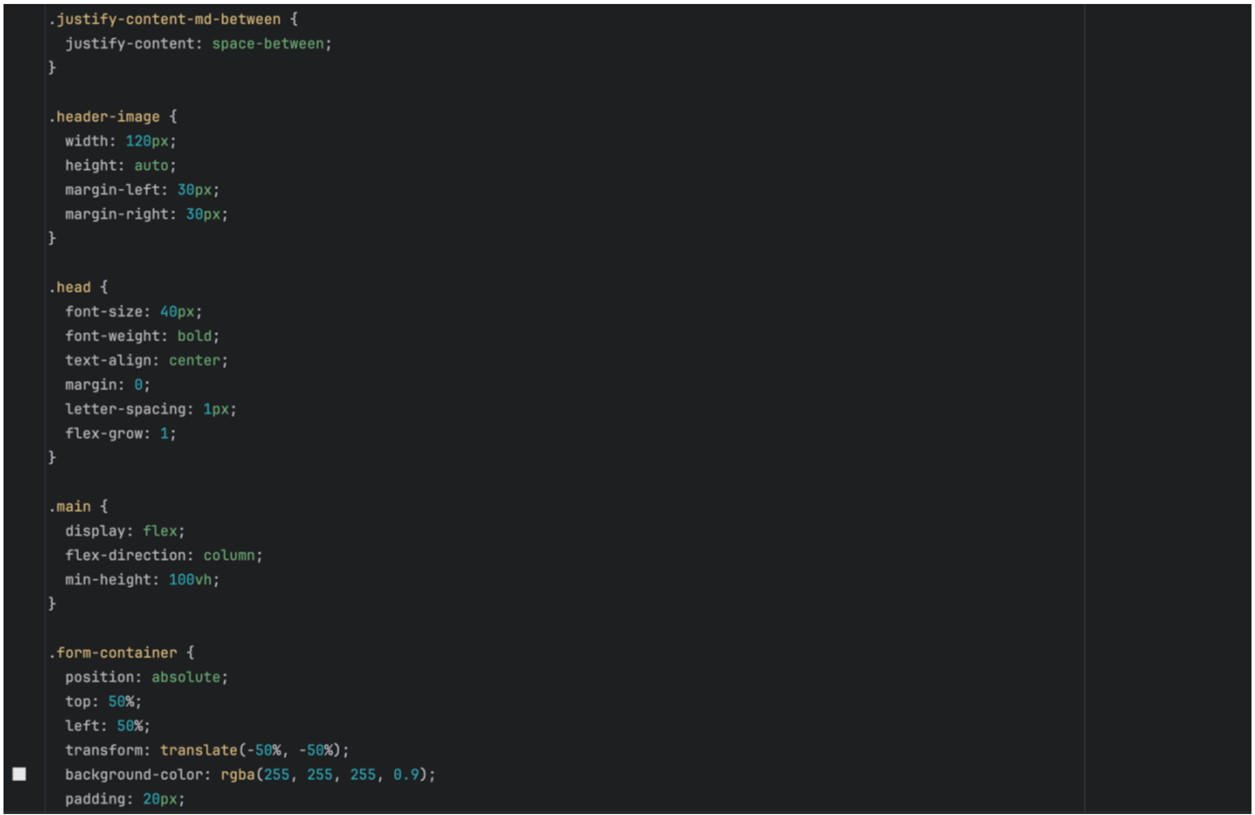
```
  </div>

<div class="col-12">
    <div class="form-check">
      <input class="form-check-input"
             type="checkbox"
             id="ageConfirmation"
             v-model="form.age_confirmation">
      <label class="form-check-label" for="ageConfirmation">
        I confirm that I am 18 years of age or older
      </label>
    </div>
  <div class="slider-container slider" >
    <label for="range">Select the Age:</label>
    <input type="range" id="range" min="18" max="70" v-model="sliderValue" @input="updateSliderValue(sliderValue)">
    <span>{{ sliderValue }}</span>
  </div>
  <div class="textbox-container">
    <label for="numberInput">Enter Age</label>
    <input
      type="number"
      id="numberInput"
      min="18"
      max="70"
      v-model="sliderValue"
      @input="updateTextboxValue(sliderValue)"
    >
  </div>
  </div>
```



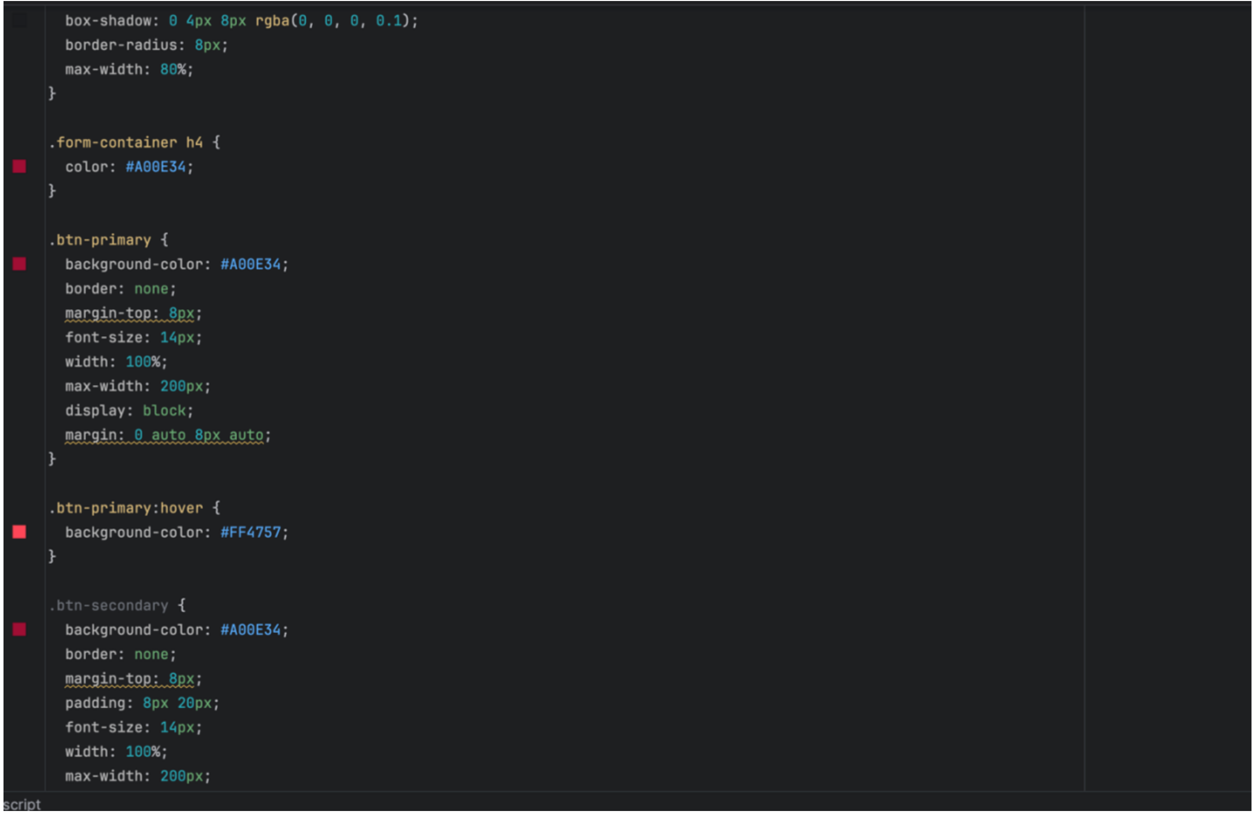
```
          <div class="col-12">
              <div class="form-check">
                <input class="form-check-input"
                       type="checkbox"
                       id="termsAndConditions"
                       v-model="form.terms_and_conditions"
                       required>
                <label class="form-check-label" for="termsAndConditions">
                  I agree to the terms and conditions
                </label>
                <div class="invalid-feedback">
                  You must agree to the terms and conditions before submitting.
                </div>
              </div>
            </div>
        </div>
        <hr class="my-4">
        <div class="col-12">
        <div class="slider-container">
  </div>
          <div class="col-12">
              <button class="btn btn-primary" type="submit" :disabled="isSubmitting" @submit="submitForm">
                Submit
              </button>
        </div>

        </div>
      </form>
      <div class="col-12">
```



```
      </div>
    </div>
  </div>
  <footer class="footer">
    <p>© 2024 Japanese National Blood Donation Program</p>
  </footer>
  </body>
</template>

<style scoped>


header {
  background-color: white;
  color: #970A30;
  padding: 20px 0;
  box-shadow: 0 4px 8px;
  border-radius: 10px;
  border-color: #A00E34;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);
}


.d-flex {
  display: flex;
}

.align-items-center {
  align-items: center;
```

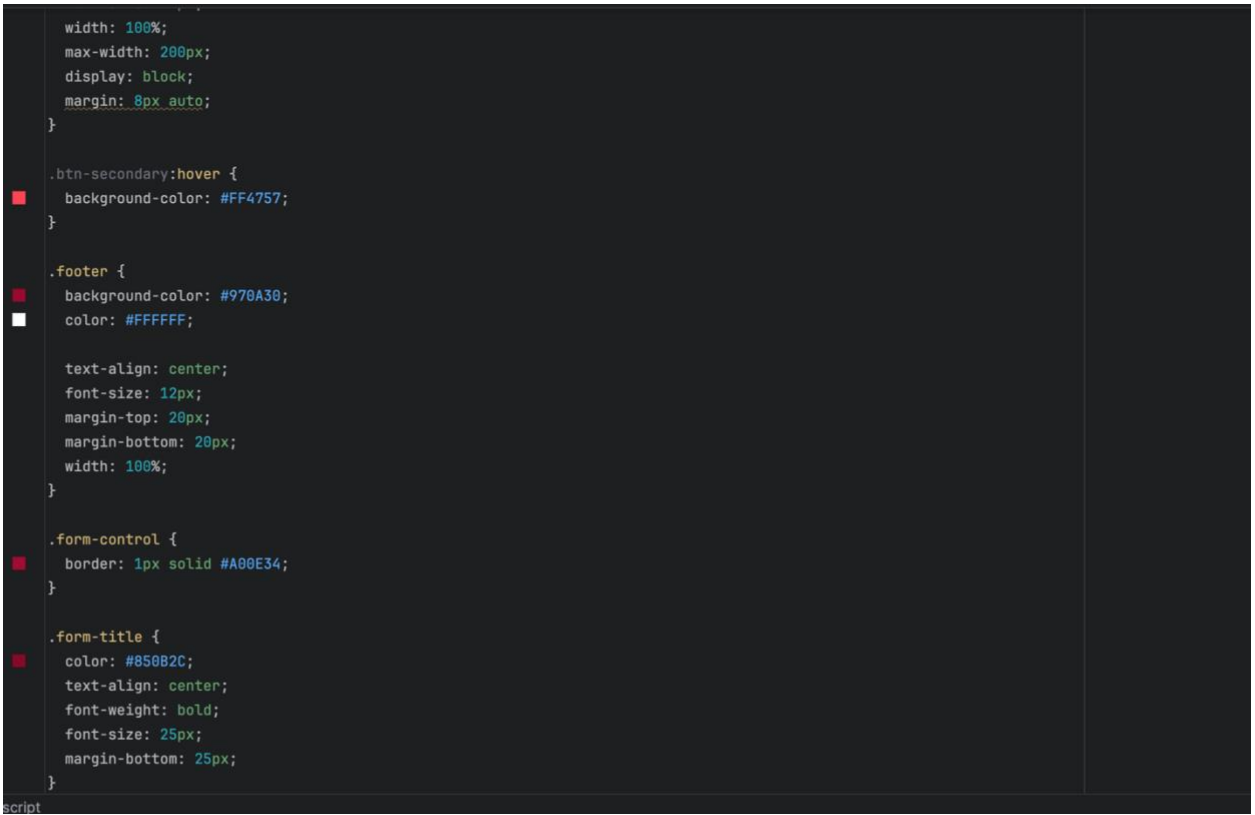


```css
.justify-content-md-between {
  justify-content: space-between;
}

.header-image {
  width: 120px;
  height: auto;
  margin-left: 30px;
  margin-right: 30px;
}

.head {
  font-size: 40px;
  font-weight: bold;
  text-align: center;
  margin: 0;
  letter-spacing: 1px;
  flex-grow: 1;
}

.main {
  display: flex;
  flex-direction: column;
  min-height: 100vh;
}

.form-container {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: rgba(255, 255, 255, 0.9);
  padding: 20px;
```



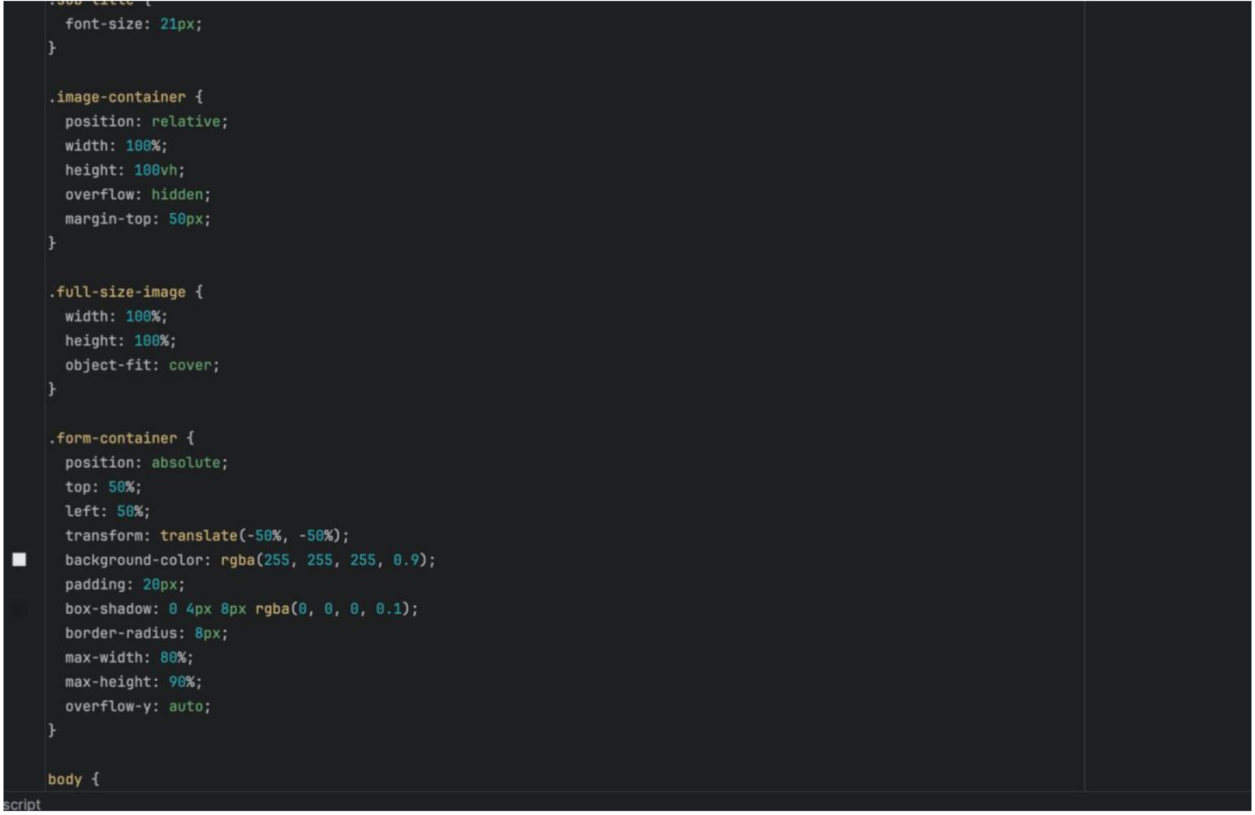
```css
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
  max-width: 80%;
}

.form-container h4 {
  color: #A00E34;
}

.btn-primary {
  background-color: #A00E34;
  border: none;
  margin-top: 8px;
  font-size: 14px;
  width: 100%;
  max-width: 200px;
  display: block;
  margin: 0 auto 8px auto;
}

.btn-primary:hover {
  background-color: #FF4757;
}

.btn-secondary {
  background-color: #A00E34;
  border: none;
  margin-top: 8px;
  padding: 8px 20px;
  font-size: 14px;
  width: 100%;
  max-width: 200px;
```

script



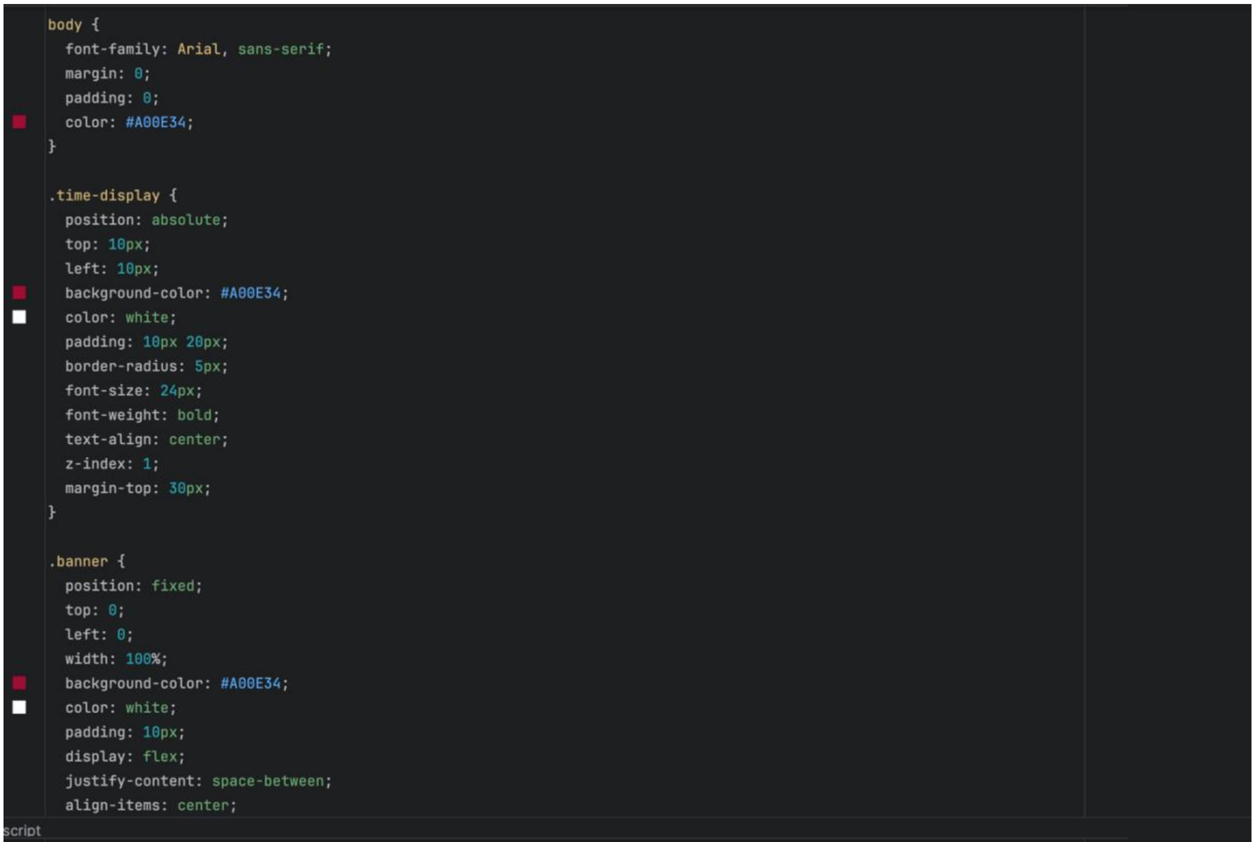
```css
  width: 100%;
  max-width: 200px;
  display: block;
  margin: 8px auto;
}

.btn-secondary:hover {
  background-color: #FF4757;
}

.footer {
  background-color: #970A30;
  color: #FFFFFF;

  text-align: center;
  font-size: 12px;
  margin-top: 20px;
  margin-bottom: 20px;
  width: 100%;
}

.form-control {
  border: 1px solid #A00E34;
}

.form-title {
  color: #850B2C;
  text-align: center;
  font-weight: bold;
  font-size: 25px;
  margin-bottom: 25px;
}
```

script



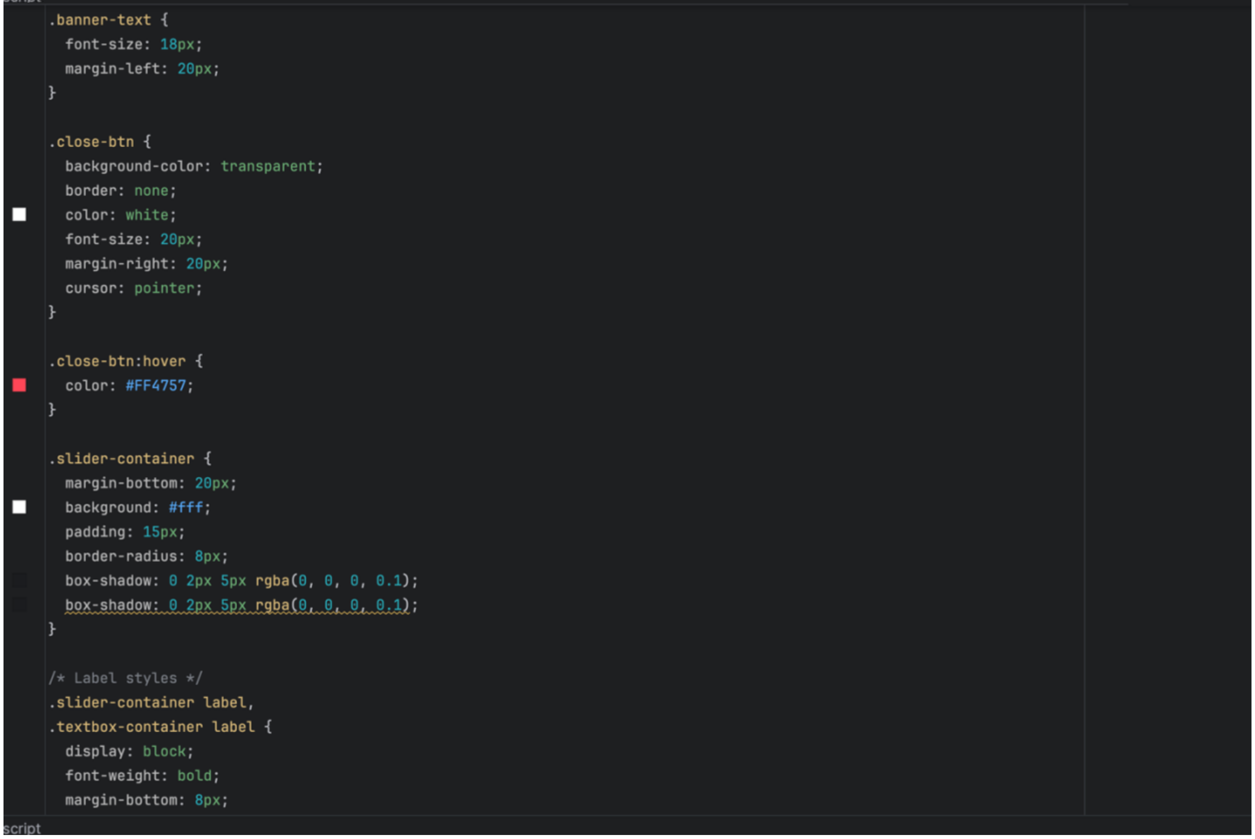
```css
.sub-title {
  font-size: 21px;
}

.image-container {
  position: relative;
  width: 100%;
  height: 100vh;
  overflow: hidden;
  margin-top: 50px;
}

.full-size-image {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

.form-container {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  background-color: rgba(255, 255, 255, 0.9);
  padding: 20px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
  max-width: 80%;
  max-height: 90%;
  overflow-y: auto;
}

body {
```

script



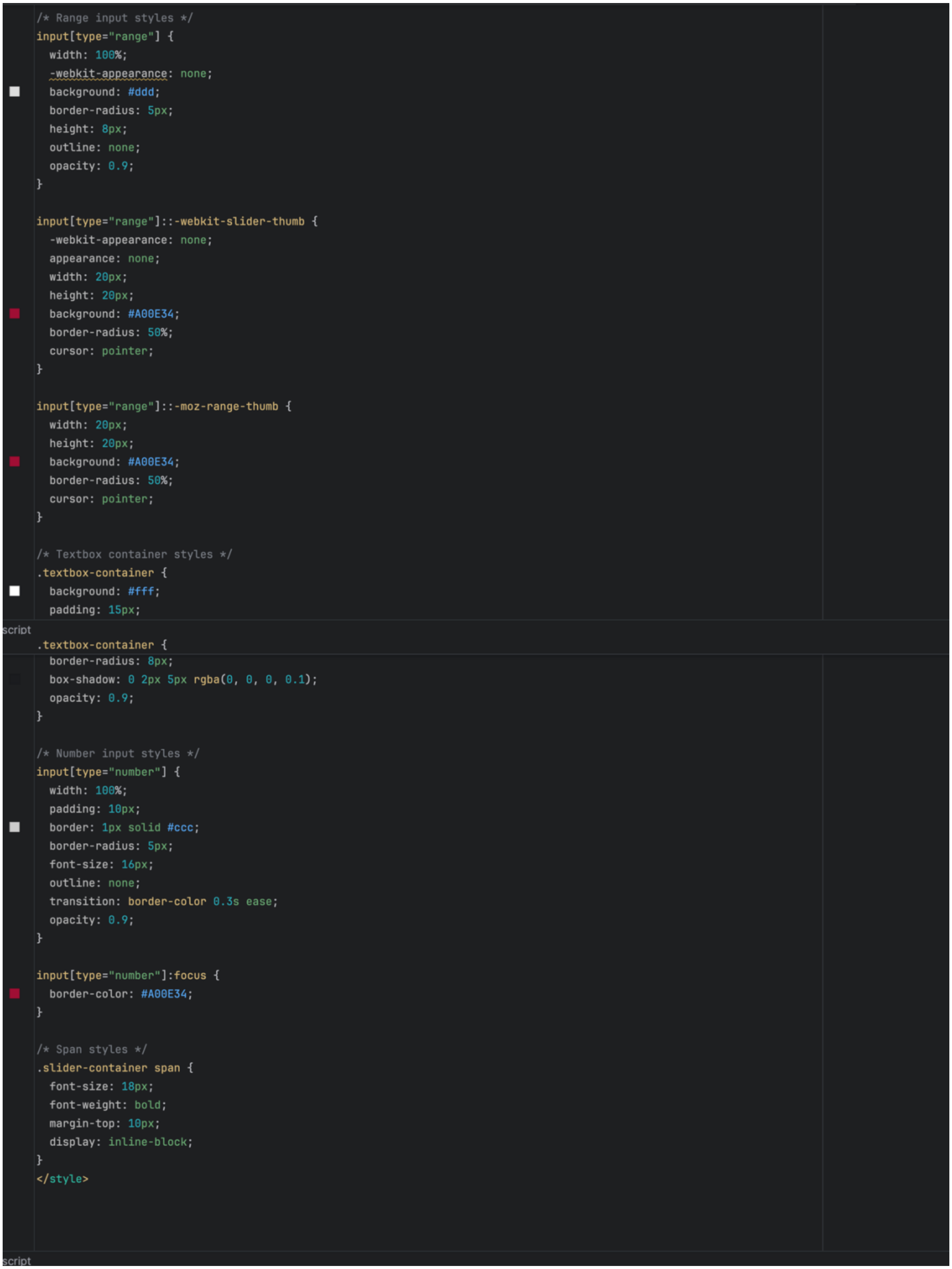
```css
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  color: #A00E34;
}

.time-display {
  position: absolute;
  top: 10px;
  left: 10px;
  background-color: #A00E34;
  color: white;
  padding: 10px 20px;
  border-radius: 5px;
  font-size: 24px;
  font-weight: bold;
  text-align: center;
  z-index: 1;
  margin-top: 30px;
}

.banner {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  background-color: #A00E34;
  color: white;
  padding: 10px;
  display: flex;
  justify-content: space-between;
  align-items: center;
```

script



```css
.banner-text {
  font-size: 18px;
  margin-left: 20px;
}

.close-btn {
  background-color: transparent;
  border: none;
  color: white;
  font-size: 20px;
  margin-right: 20px;
  cursor: pointer;
}

.close-btn:hover {
  color: #FF4757;
}

.slider-container {
  margin-bottom: 20px;
  background: #fff;
  padding: 15px;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

/* Label styles */
.slider-container label,
.textbox-container label {
  display: block;
  font-weight: bold;
  margin-bottom: 8px;
```

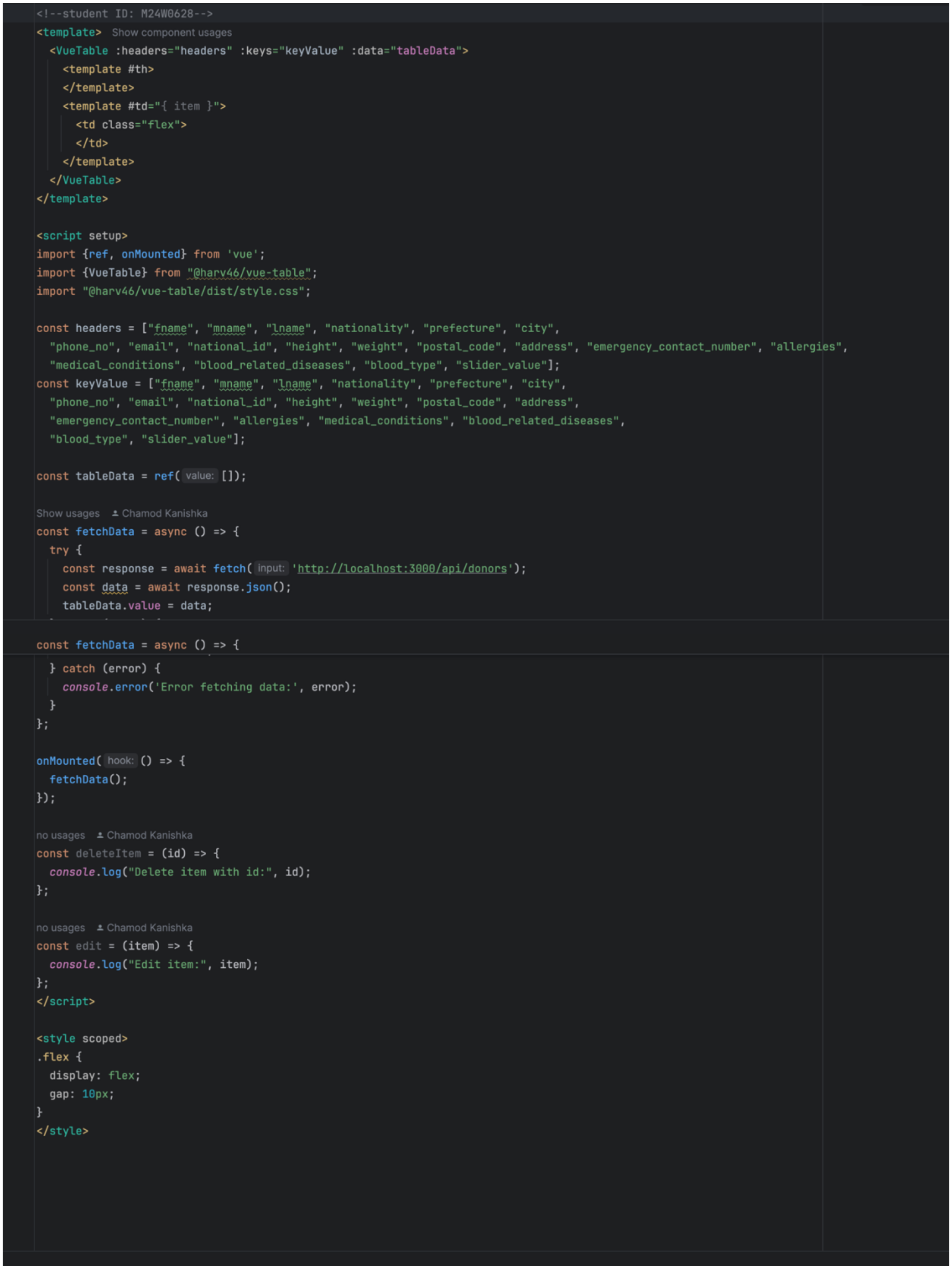
script



```
/* Range input styles */
input[type="range"] {
  width: 100%;
  -webkit-appearance: none;
  background: #ddd;
  border-radius: 5px;
  height: 8px;
  outline: none;
  opacity: 0.9;
}

input[type="range"]::-webkit-slider-thumb {
  -webkit-appearance: none;
  appearance: none;
  width: 20px;
  height: 20px;
  background: #A00E34;
  border-radius: 50%;
  cursor: pointer;
}

input[type="range"]::-moz-range-thumb {
  width: 20px;
  height: 20px;
  background: #A00E34;
  border-radius: 50%;
  cursor: pointer;
}

/* Textbox container styles */
.textbox-container {
  background: #fff;
  padding: 15px;
```

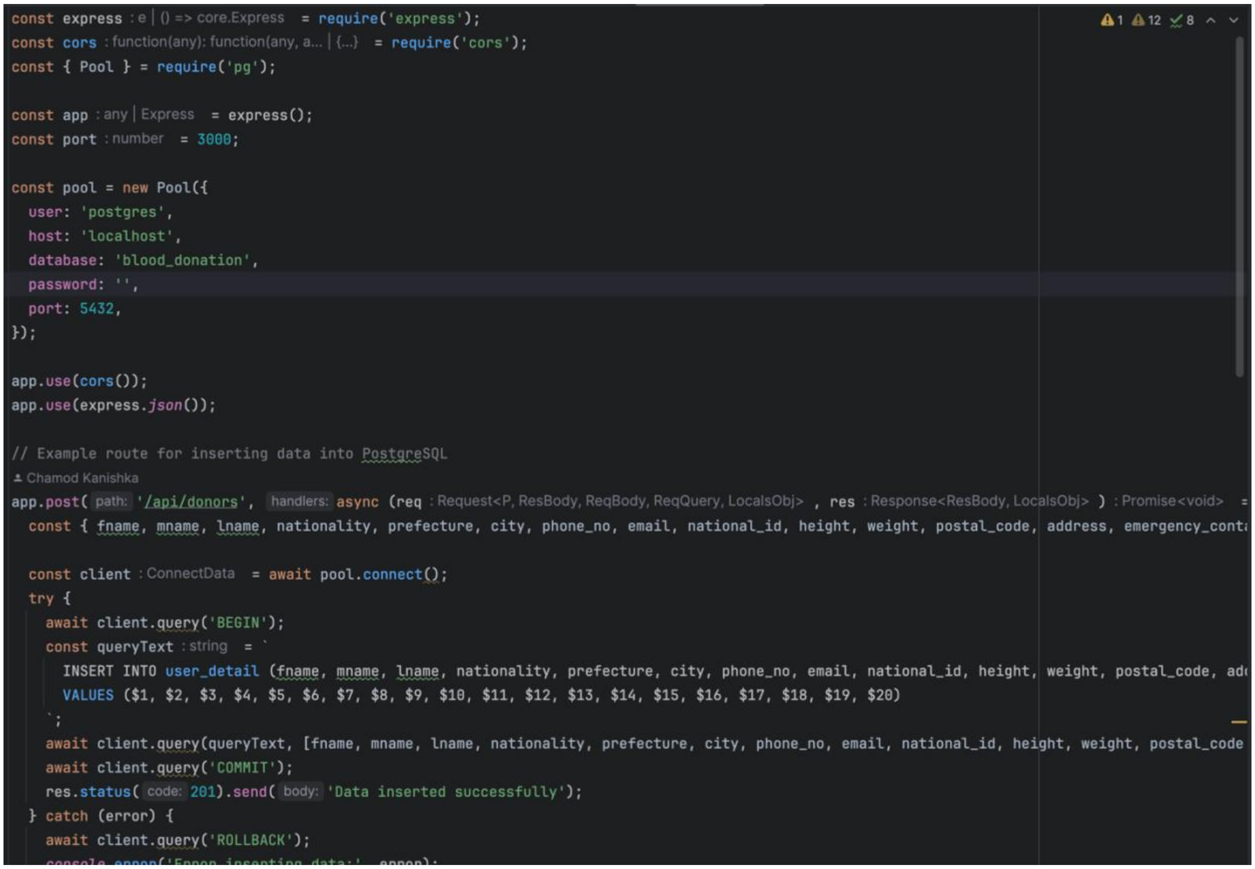
```
.textbox-container {
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
  opacity: 0.9;
}

/* Number input styles */
input[type="number"] {
  width: 100%;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 16px;
  outline: none;
  transition: border-color 0.3s ease;
  opacity: 0.9;
}

input[type="number"]:focus {
  border-color: #A00E34;
}

/* Span styles */
.slider-container span {
  font-size: 18px;
  font-weight: bold;
  margin-top: 10px;
  display: inline-block;
}
</style>
```

ShowData.vue

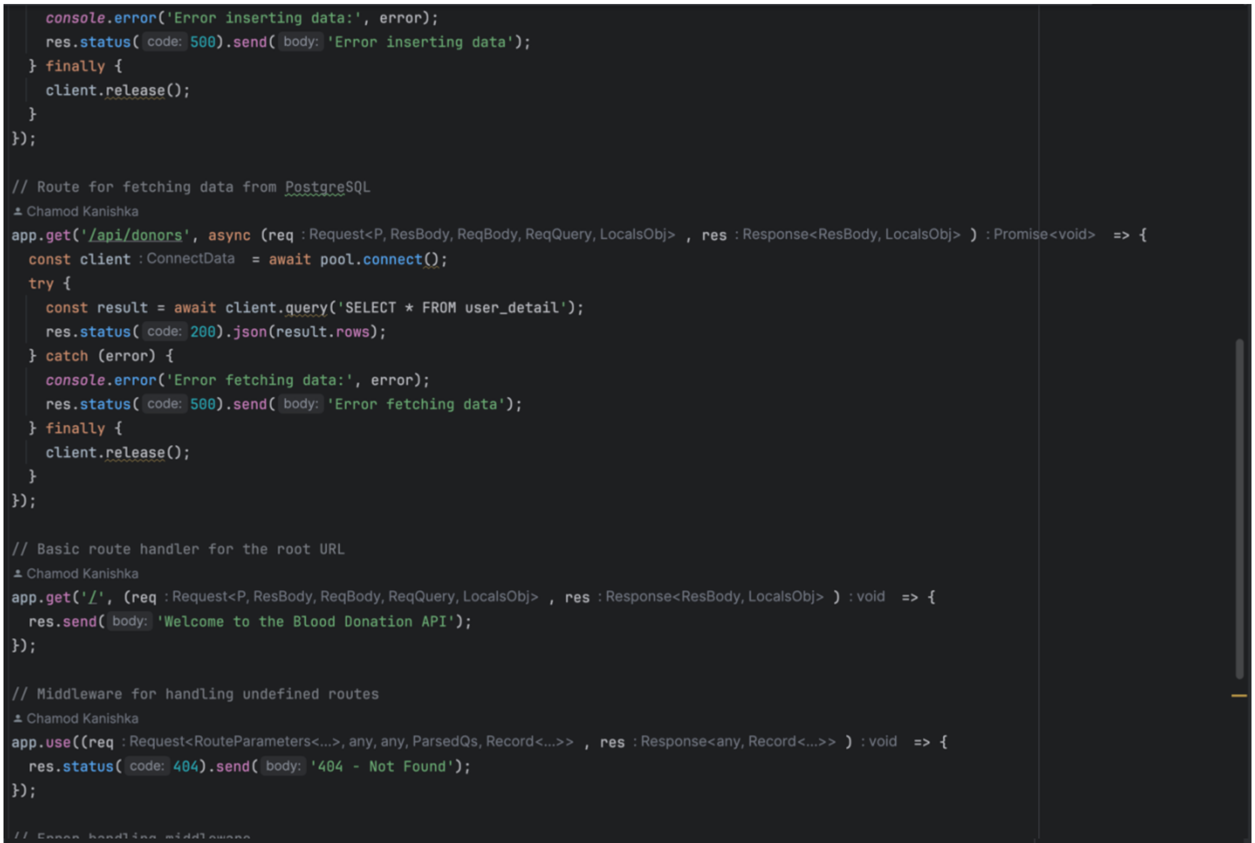


```
<!--student ID: M24W0628-->
<template>  Show component usages
  <VueTable :headers="headers" :keys="keyValue" :data="tableData">
    <template #th>
    </template>
    <template #td="{ item }">
      <td class="flex">
      </td>
    </template>
  </VueTable>
</template>

<script setup>
import {ref, onMounted} from 'vue';
import {VueTable} from "@harv46/vue-table";
import "@harv46/vue-table/dist/style.css";

const headers = ["fname", "mname", "lname", "nationality", "prefecture", "city",
  "phone_no", "email", "national_id", "height", "weight", "postal_code", "address", "emergency_contact_number", "allergies",
  "medical_conditions", "blood_related_diseases", "blood_type", "slider_value"];
const keyValue = ["fname", "mname", "lname", "nationality", "prefecture", "city",
  "phone_no", "email", "national_id", "height", "weight", "postal_code", "address",
  "emergency_contact_number", "allergies", "medical_conditions", "blood_related_diseases",
  "blood_type", "slider_value"];

const tableData = ref( value: []);

Show usages   Chamod Kanishka
const fetchData = async () => {
  try {
    const response = await fetch( input: 'http://localhost:3000/api/donors');
    const data = await response.json();
    tableData.value = data;

const fetchData = async () => {
  } catch (error) {
    console.error('Error fetching data:', error);
  }
};

onMounted( hook: () => {
  fetchData();
});

no usages   Chamod Kanishka
const deleteItem = (id) => {
  console.log("Delete item with id:", id);
};

no usages   Chamod Kanishka
const edit = (item) => {
  console.log("Edit item:", item);
};
</script>

<style scoped>
.flex {
  display: flex;
  gap: 10px;
}
</style>
```

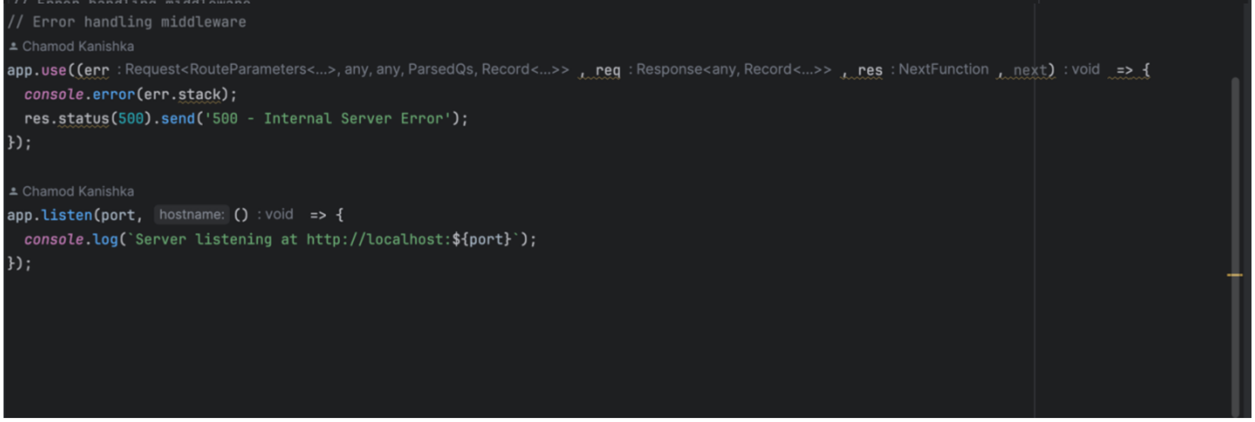
Code Part of Backend/API



```
const express : e | () => core.Express  = require('express');
const cors : function(any): function(any, a... | {...}  = require('cors');
const { Pool } = require('pg');

const app : any | Express  = express();
const port : number  = 3000;

const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'blood_donation',
  password: '',
  port: 5432,
});

app.use(cors());
app.use(express.json());

// Example route for inserting data into PostgreSQL
Chamod Kanishka
app.post( path: '/api/donors',  handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<void>  =
  const { fname, mname, lname, nationality, prefecture, city, phone_no, email, national_id, height, weight, postal_code, address, emergency_conta

  const client : ConnectData  = await pool.connect();
  try {
    await client.query('BEGIN');
    const queryText : string  = `
      INSERT INTO user_detail (fname, mname, lname, nationality, prefecture, city, phone_no, email, national_id, height, weight, postal_code, add
      VALUES ($1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15, $16, $17, $18, $19, $20)
    `;
    await client.query(queryText, [fname, mname, lname, nationality, prefecture, city, phone_no, email, national_id, height, weight, postal_code
    await client.query('COMMIT');
    res.status( code: 201).send( body: 'Data inserted successfully');
  } catch (error) {
    await client.query('ROLLBACK');
    console.error('Error inserting data:', error);
```

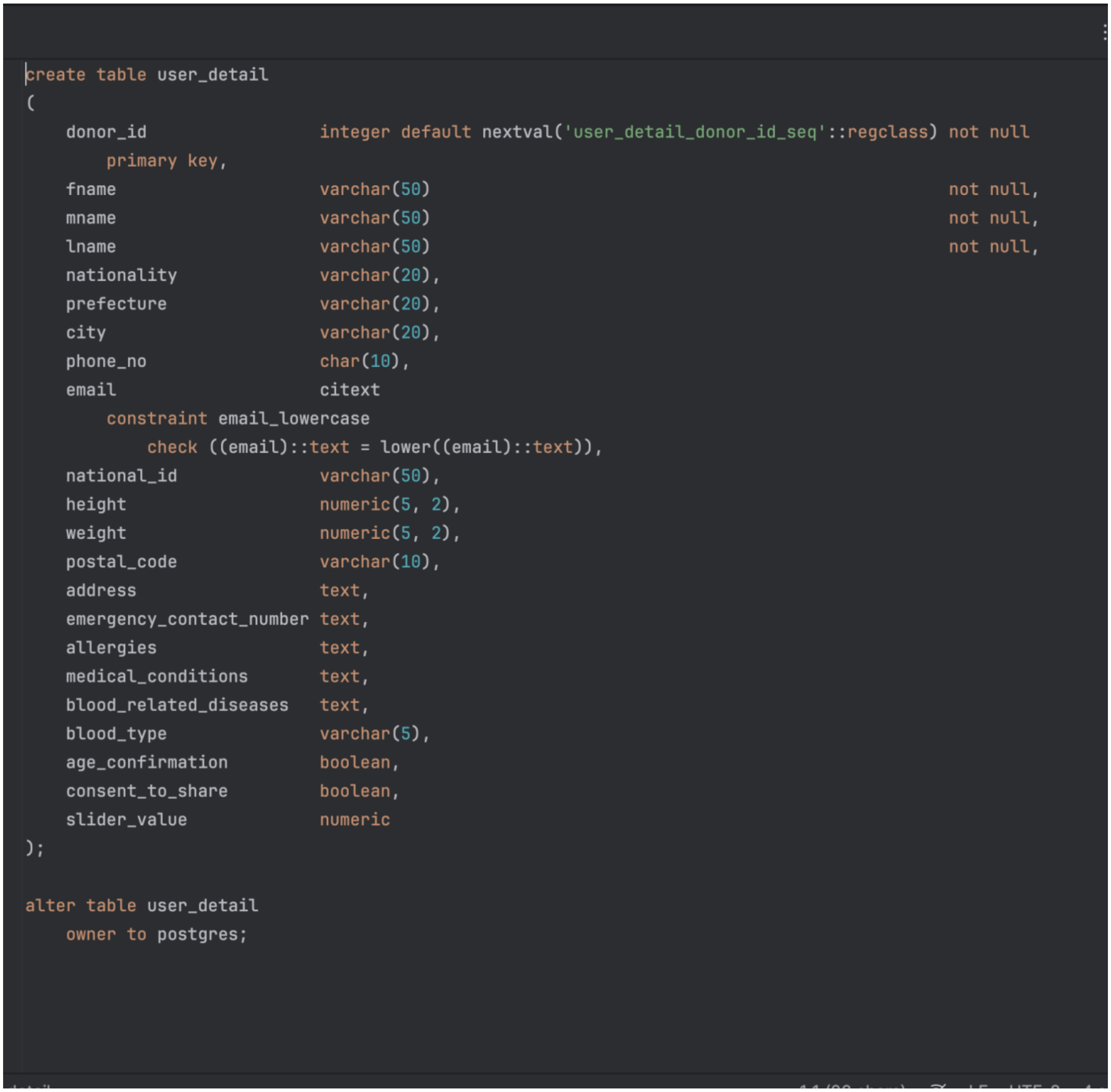


```
    console.error('Error inserting data:', error);
    res.status( code: 500).send( body: 'Error inserting data');
  } finally {
    client.release();
  }
});

// Route for fetching data from PostgreSQL
≗ Chamod Kanishka
app.get('/api/donors', async (req :Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res :Response<ResBody, LocalsObj> ) :Promise<void>  => {
  const client :ConnectData  = await pool.connect();
  try {
    const result = await client.query('SELECT * FROM user_detail');
    res.status( code: 200).json(result.rows);
  } catch (error) {
    console.error('Error fetching data:', error);
    res.status( code: 500).send( body: 'Error fetching data');
  } finally {
    client.release();
  }
});

// Basic route handler for the root URL
≗ Chamod Kanishka
app.get('/', (req :Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res :Response<ResBody, LocalsObj> ) :void  => {
  res.send( body: 'Welcome to the Blood Donation API');
});

// Middleware for handling undefined routes
≗ Chamod Kanishka
app.use((req :Request<RouteParameters<...>, any, any, ParsedQs, Record<...>> , res :Response<any, Record<...>> ) :void  => {
  res.status( code: 404).send( body: '404 - Not Found');
});
```



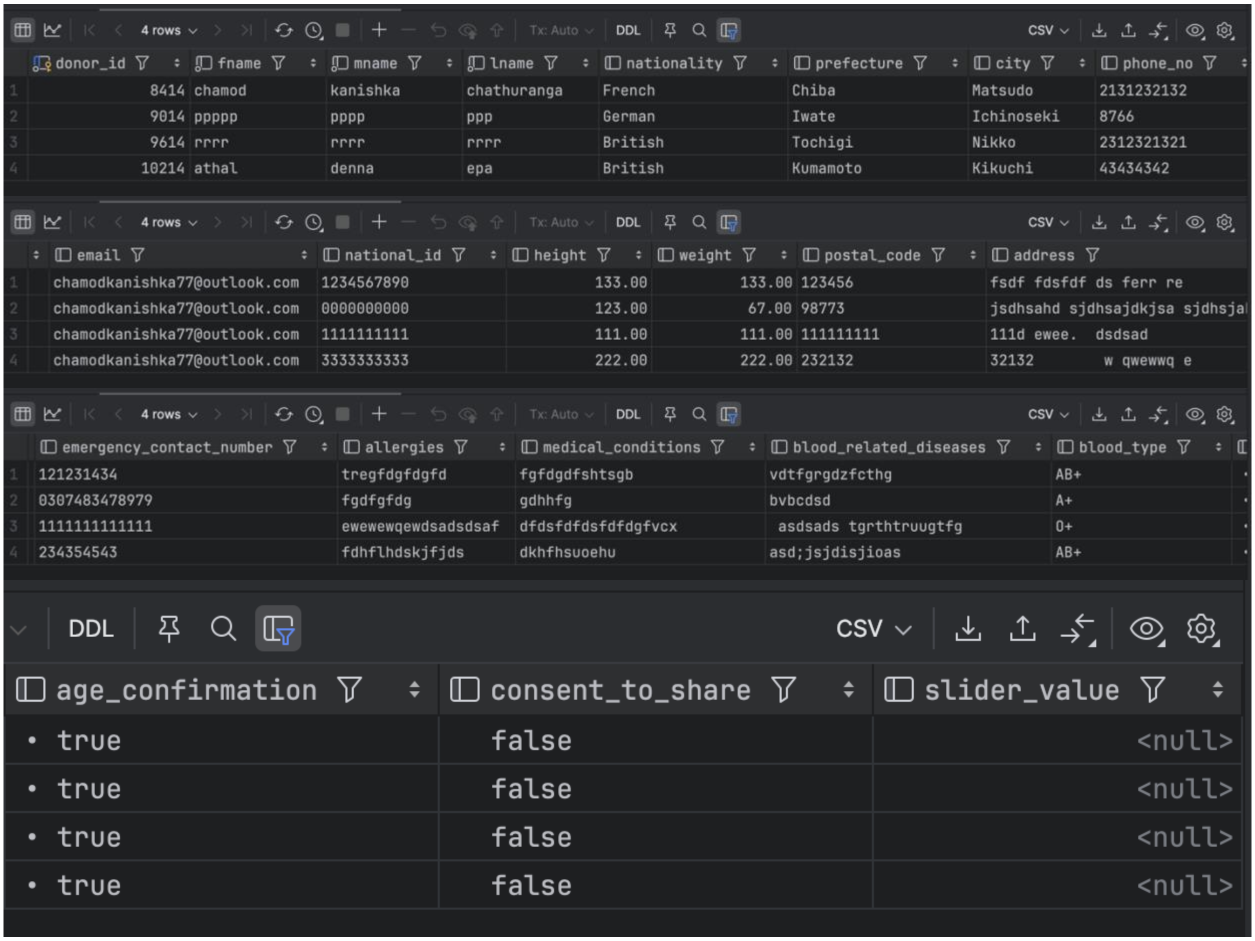
```
// Error handling middleware
≗ Chamod Kanishka
app.use((err :Request<RouteParameters<...>, any, any, ParsedQs, Record<...>> , req :Response<any, Record<...>> , res :NextFunction , next) :void  => {
  console.error(err.stack);
  res.status(500).send('500 - Internal Server Error');
});

≗ Chamod Kanishka
app.listen(port,  hostname: () :void  => {
  console.log(`Server listening at http://localhost:${port}`);
});
```

Code Part of Database (SQL)



```
create table user_detail
(
    donor_id                 integer default nextval('user_detail_donor_id_seq'::regclass) not null
        primary key,
    fname                    varchar(50)                                                   not null,
    mname                    varchar(50)                                                   not null,
    lname                    varchar(50)                                                   not null,
    nationality              varchar(20),
    prefecture               varchar(20),
    city                     varchar(20),
    phone_no                 char(10),
    email                    citext
        constraint email_lowercase
            check ((email)::text = lower((email)::text)),
    national_id              varchar(50),
    height                   numeric(5, 2),
    weight                   numeric(5, 2),
    postal_code              varchar(10),
    address                  text,
    emergency_contact_number text,
    allergies                text,
    medical_conditions       text,
    blood_related_diseases   text,
    blood_type               varchar(5),
    age_confirmation         boolean,
    consent_to_share         boolean,
    slider_value             numeric
);

alter table user_detail
    owner to postgres;
```

Visual of the Table



	donor_id	fname	mname	lname	nationality	prefecture	city	phone_no
1	8414	chamod	kanishka	chathuranga	French	Chiba	Matsudo	2131232132
2	9014	ppppp	pppp	ppp	German	Iwate	Ichinoseki	8766
3	9614	rrrr	rrrr	rrrr	British	Tochigi	Nikko	2312321321
4	10214	athal	denna	epa	British	Kumamoto	Kikuchi	43434342

	email	national_id	height	weight	postal_code	address
1	chamodkanishka77@outlook.com	1234567890	133.00	133.00	123456	fsdf fdsfdf ds ferr re
2	chamodkanishka77@outlook.com	0000000000	123.00	67.00	98773	jsdhsahd sjdhsajdkjsa sjdhsja
3	chamodkanishka77@outlook.com	1111111111	111.00	111.00	111111111	111d ewee. dsdsad
4	chamodkanishka77@outlook.com	3333333333	222.00	222.00	232132	32132 w qwewwq e

	emergency_contact_number	allergies	medical_conditions	blood_related_diseases	blood_type
1	121231434	tregfdgfdgfd	fgfdgdfshtsgb	vdtfgrgdzfcthg	AB+
2	0307483478979	fgdfgfdg	gdhhfg	bvbcdsd	A+
3	1111111111111	ewewewqewdsadsdsaf	dfdsfdfdsfdfdgfvcx	asdsads tgrthtruugtfg	O+
4	234354543	fdhflhdskjfjds	dkhfhsuoehu	asd;jsjdisjioas	AB+

	age_confirmation	consent_to_share	slider_value
•	true	false	<null>
•	true	false	<null>
•	true	false	<null>
•	true	false	<null>

Description

The Japanese National Blood Donation Program uses an advanced and user-friendly online application to improve and expedite the blood donation procedure. The project's main goal is to provide a safe and effective platform where aspiring blood donors may register and provide their personal and health information. This will make donor data administration and collecting easier and more efficient.

Using Vue.js, a thorough registration form forms the basis of this project. Numerous facts are collected on the form, such as blood type and vital medical history in addition to personal information like name, nationality, residence, and contact details. For the blood donation procedure to be safe and effective, it is imperative that comprehensive and precise data be gathered, which is ensured by this method.

Important components of the registration form consist of:

- Real-time Validation: To make sure that all needed fields are filled out accurately, the form has real-time validation. This entails verifying that the data entered is accurate and comprehensive by looking for legitimate phone numbers, postal codes, email formats, and national IDs.
- Prefecture and City Selection: To make it simpler for users to enter their address precisely, the form dynamically changes the list of cities based on the prefecture that is selected.
- Medical History and Blood Type: Users are able to provide comprehensive details on their blood type, allergies, and illnesses connected to their health. Radio buttons are provided in the blood type area to make choosing simple and guarantee correct and speedy input.

- Age and Consent Confirmation: In accordance with legal and ethical requirements, the form has checkboxes for age verification and consent to disclose information.

The project includes front-end programming as well as a solid backend infrastructure that handles form submissions and communicates with a PostgreSQL database using Node.js and Express. This guarantees that the gathered data is properly handled, accessible, and kept securely.

The overall goal of the web application for the Japanese National Blood Donation Program is to guarantee the security and effectiveness of the blood donation procedure while also improving user experience and data accuracy. This initiative advances public health and safety in addition to meeting the blood donation program's operational demands.