

**School of Computer Science and Engineering**

**FALL SEMESTER 2020-21**

**CSE2005**

**Title of the Project**

**COVID Testing Scheduling Portal**

**Team Members**

**Anuraj Agarwal 19BCE2026**

**Pavitra Goyal 19BCE0373**

**Samarthya Jha 19BCB0011**

**Shatayu Mittal 19BCE0446**

**Submitted to**

**Dr. Geraldine Bessie Amali, SCOPE**

**Table of Content**

**Abstract**

In this paper, we developed a COVID Testing Scheduling Portal which not only helps any organization to prioritize the patient appointment list on the basis of symptom severities but also on the basis of age and past medical history of the concerned patient. This will decrease the waiting time for patients which are more severely infected than others, thus increasing the efficiency of the concerned organization to oversee all the arrangements.

One of the biggest reasons that online appointment scheduling is getting popular day by day is that it helps the patient to make the appointment to their doctor, clinic or hospital in an easier way. It makes it through the computer, access a website or software and makes an appointment, then to go to the hospital, wait in a line for a number of hours, just to make an appointment with the doctor for the next week or next month.

Due to the large amount of testing applications and a limited number of tests available each day, people with severe COVID symptoms sometimes have to wait for over 3 days to get their test results after they've given their swab. The present system also causes non symptomatic patients to get the results the very next day which is not that necessary, relatively.

We developed a fully responsive portal which will take information from the patient applying for an appointment for Covid-19 test, regarding his age, symptoms like fever, cold and breathlessness, and past medical history. The designed algorithm will assign the priority based on these parameters and thus assign scheduled timing for testing and provide an appointment key which is unique for every patient as it is linked to their unique Aadhar card number.

Through our project, patients can also involve in the health decisions that they have to make. They can make an appointment to another doctor other than theirs, by nothing more than a click. Thus, modernizing with the technology in the health sector of our society and tackling the problems faced in the pandemic.

**INTRODUCTION**

COVID-19, commonly referred to as novel coronavirus has been identified to have its roots in Wuhan, a small city of China. First casualty caused due to the virus has been identified in November 2019. Ever since, the world tally of infected individuals has seen an exorbitant rise. Initial speculations state that the virus is known to be present in animals such as bats and pangolins, which in turn got transmitted into humans. Studies are being conducted at various national and international levels.

It is quite clear now that the impact of Covid-19 lies way beyond influenza-like conditions, even for the majority who are not much affected by this dreaded disease. It is also evident that this disease affects many more organs beyond the lungs and the respiratory tract including the heart, kidney, brain, and the gastrointestinal tract. What more do we know and how can we be alert about these conditions, particularly since some of that impact can be long-term?

The most common thing that they are facing – the smallest problem – is that they continue to have a low-grade fever for a much longer time than we expected. They would have fever for about four to five weeks, and without any other cause being found. That really troubles the patients and their relatives.

Severe shortages of testing supplies and long waits for test results (frequently 7 days or longer) made it difficult to monitor patient and staff health. Hospitals reported lacking testing kits altogether, or lacking components of testing kits. Also, the long waits for results contributed to longer patient stays in the hospital.

In this paper, we developed a COVID Testing Scheduling Portal which not only helps any organisation to prioritize the patient appointment list on the basis of symptom severities but also on the basis of age and past medical history of the concerned patient. This will decrease the waiting time for patients which are more severely infected than others, thus increasing the efficiency of the concerned organisation to oversee all the arrangements.

One of the biggest reasons that online appointment scheduling is getting popular day by day is that it helps the patient to make the appointment to their doctor, clinic or hospital in an easier way. It makes it through the computer, access a website or software and makes an appointment, then to go to the hospital, wait in a line for a number of hours, just to make an appointment with the doctor for the next week or next month.

**HARDWARE AND SOFTWARE REQUIREMENT**

**Hardware required:**

Just to install and run a web browser.

Example: For chrome

Processor: Windows Pentium 4

Process Mac: Intel

Min RAM: 128 MB

Min Storage: 100 MB

Windows: Windows XP / SP2

OS X: OS X 10.5.6

Linux: Ubuntu 10.04

**Software required:**

A web browser

Nodejs

ExpressJs

MongoDB

**EXISTING APPROACH**

Due to the coronavirus pandemic, patients suffering from critical diseases like cancer and those requiring surgeries are delaying hospital visits and thereby endangering their health and even risking their lives, according to some doctors.

A number of medical practitioners said they have even lost patients who could have survived or their situation salvaged had they reported to hospitals in time. Doctors say patients are even delaying simple surgeries like gall bladder stone removal which could lead to major issues in the long run.

Due to the large amount of testing applications and a limited number of tests available each day, people with severe COVID symptoms sometimes have to wait for over 3 days to get their test results after they've given their swab. The present system also causes non symptomatic patients to get the results the very next day which is not that necessary, relatively.

The existing schedule uses individual-block fixed-interval scheduling. The session typically starts around a fixed time and patients are booked in at intervals, that are usually of twenty minutes, with a break for lunch and a shorter session starting after lunch. Without jumping into the analysis of the schedule, it is worth mentioning that the duration of each appointment varies largely. This makes it very challenging to produce a good schedule that keeps both patient waiting times and surgeon idle times low.

Another challenge in the scheduling is to predict when the same-day diagnostic examination report will be ready. To manage this uncertainty, patients that have an examination are only given the reservation time of the examination and are not given the time of the following surgeon appointment. A patient with an exam is instructed to go to the clinic waiting room after the examination and there they wait indefinitely until the surgeon has the report and an available time to see them. In the data it can be seen that the diagnostic examination is generally in the morning and the following appointment time reserved for patients tends to be several hours after the exam, but in reality, the patient is ready much sooner, and may wait for a long time to be seen.

**PROPOSED MODEL**

Our designed algorithm scheduler reduces the time interval between Covid-19 test and organise every test by assigning them a 30 minutes interval which increases the flexibility for the doctors and clinic staff members as well as the patients. Due to fixed time gap, it maximizes the number of Covid-19 tests per day.

The proposed schedule uses a priority-based algorithm to not only prioritize the patient appointment list on the basis of symptom severities but also on the basis of age and their past medical history. This will decrease the waiting time for patients which are more severely infected than others, thus increasing the efficiency of the concerned organisation to oversee all the arrangements. It helps the clinic treat Covid-19 infected patients much faster and systematically.

We developed fully responsive portal which will take information, from the patient applying for appointment for Covid-19 test, regarding his age, symptoms like fever, cold and breathlessness, and past medical history. The designed algorithm will assign the priority based on these parameters and thus assign scheduled timing for testing and provide a appointment key which is unique for every patient as it is linked to their unique Aadhar card number.

At the end of each day, the order of the applicants is assigned by assigning priority to each applicant depending on their Covid-19 severity. The higher priority goes to the one with the high severity.

**METHODOLOGY**

The first step will be to assign a particular score based on severity of each of the three properties resulting in deaths:

|  |  |
| --- | --- |
| **Attributes** | **Score** |
| Age | 30% |
| Symptoms | 40% |
| Past Medical History | 30% |

The Age Score:

Infants and elders are more susceptible to succumb to COVID rather than a teenager or an adult due to either underdevelopment of their immune system or its degeneration. Hence, factors are assigned to specific brackets of age and the score of age is calculated as follows:

|  |  |  |
| --- | --- | --- |
| Age Bracket | Factor Assigned | Score Calculation |
| 0-13 | 3 | 3 X (14-Age) |
| 14-50 | 1 | 1 |
| Above 50 | 2 | 2 X (Age - 50) |

The Symptom Score:

The person can be non-symptomatic, have moderate or severe symptoms. Also, it is observed that breathlessness is the major factor of death due to respiratory failure followed by fever and then cough. Hence, the factors assigned for the various symptoms are as follows:

|  |  |
| --- | --- |
| Symptom | Factor |
| Breathlessness | 50% |
| Fever | 30% |
| Cough | 20% |

The severity is assigned the factors as follows:

|  |  |
| --- | --- |
| Severity | Score Assigned |
| None | 0 |
| Moderate | 100 |
| Severe | 200 |

The final symptom score is calculated as follows:

Score = 0.5 X Severity score of breathlessness + 0.3 X Severity score of fever + 0.2 X Severity score of coughs.

The Past Medical History Score: **T**

As per reports by WHO, cardiovascular diseases which have caused patients to undergo a heart transplant or similar pulmonary diseases are far more severe and damage-causing in the COVID scenario than gastroenteric diseases which include kidney issues. Neurological diseases are not very severe and hence are assigned a smaller factor.

|  |  |
| --- | --- |
| Disease | Factor |
| Cardio | 50% |
| Gastro | 30% |
| Neuro | 20% |

**Past Medical History Score:**

The presence or absence of past medical issues are given the score as follows:

|  |  |
| --- | --- |
| History | Score Assigned |
| Present | 200 |
| Absent | 0 |

The final past medical history score is calculated by:

Score = 0.5 X (cardio) + 0.3 X (gastro) + 0.2 X (neuro)

**THE FINAL SCORE:**

The final score of each applicant is calculated as:

Final Score = 0.4 X (Symptom Score) + 0.3 X (Age Score) + (0.3 X Past Medical History Score).

At the end of each day, the order of the applicants is assigned by assigning priority to each applicant depending on the final score. The higher priority goes to the one with the greater final score. If two applicants get the same final score, then priority is given to the one who has lesser number of days from the date their symptoms started to show.

Each applicant is given an ID through which they can view their timings.

**MODULE WISE DESCRIPTION**

1. Landing Module

At the landing page we have provided options for contact, appointment booking, status booking, motto and many more features that are a pleasure to view.

This part of the project is handled by:

* Anuraj Agarwal 19BCE2026 (Front-end Development)
* Shatayu Mittal 19BCE0446 (Front-end Development)
* Pavitra Goyal 19BCE0373 (Back-end Development)

2. Applicant Detail Module

This Part of the projects highlights our system of inputs and storing data and here the visiting patient is asked to fill a form if he/she is willing to.

Upon successful submission of the form the patient is provided with a special key which is also his key to view the appointment status.

The input is as follows:

Age  
Symptoms (no / moderate / severe):  
- Fever  
- Cough  
- Breathlessness

Past Medical History (yes/no):  
- Cardiovascular  
- Neurological  
- Gastroenteric

PERSONAL INFO:

Name; Phone no; Aadhar No (to check duplicity);

This part of the module is handled by:

* Anuraj Agarwal 19BCE2026 (Front-end Development)
* Pavitra Goyal 19BCE0373 (Back-end Development)
* Samarthya Jha 19BCB0011 (Input Design)

3. Know your appointment module

This module deals with the patient data and tells the patient about his appointment status and timings at which the testing facility will be available.

This is the heart of our project as this is where our developed algorithm will play its role.

When the module is invoked the algorithm runs on the whole of the available patient data and then the score is calculated and the patient is given the time, which is also subject to change.

This module is handled by:

* Anuraj Agarwal 19BCE2026 (Front-end Development)
* Pavitra Goyal 19BCE0373 (Algorithm Development and Back-end Development)
* Samarthya Jha 19BCB0011 (Algorithm Design and Work Flow)

GIST:

FRONT-END:

* Anuraj Agarwal 19BCE2026
* Shatayu Mittal 19BCE0446

BACK-END:

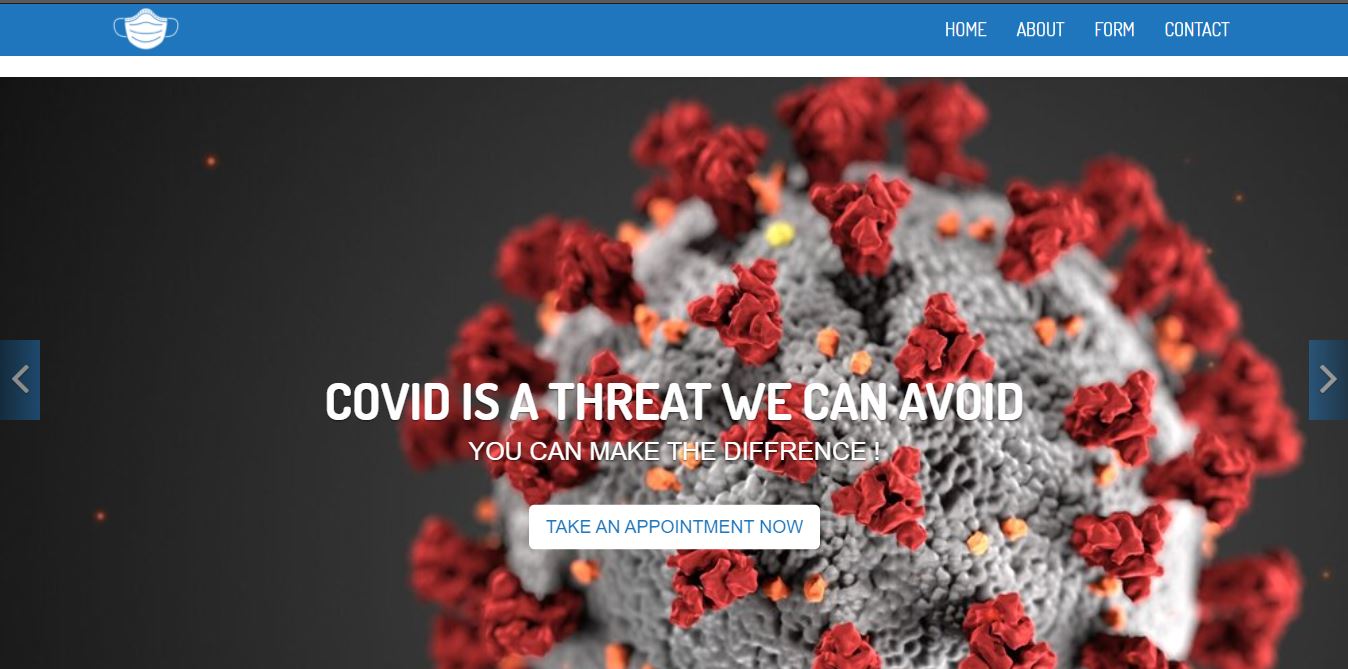
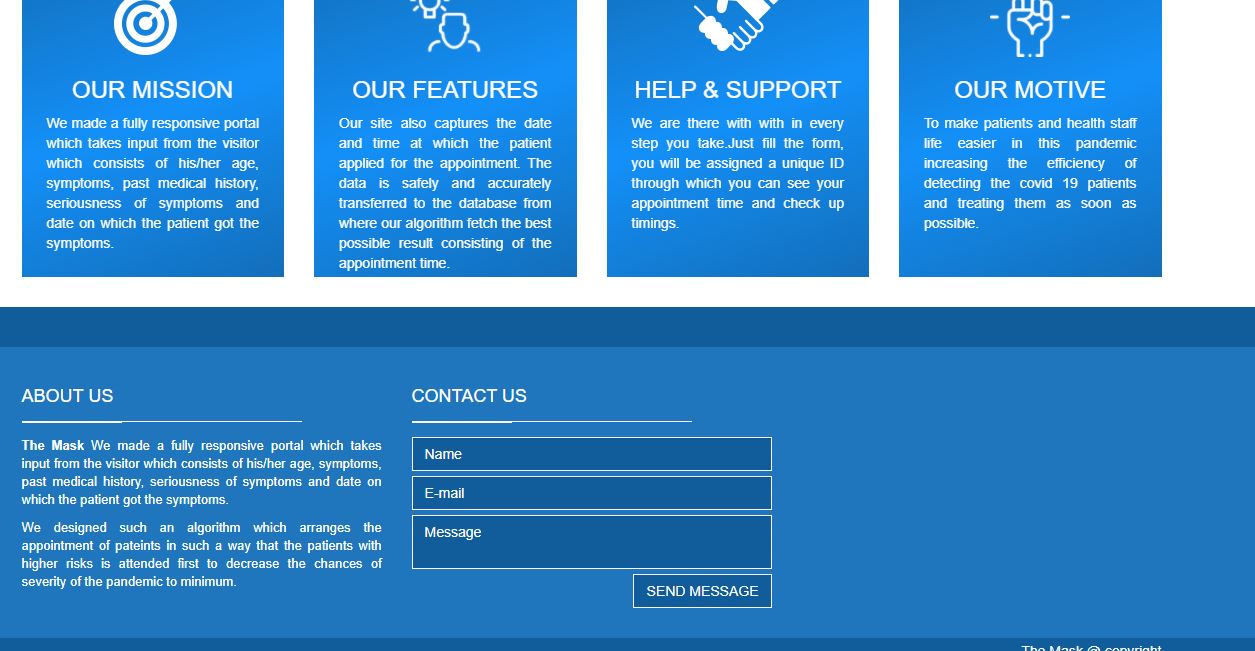
* Pavitra Goyal 19BCE0373
* Samarthya Jha 19BCB0011

TESTING AND DATA FEEDING:

* Anuraj Agarwal 19BCE2026
* Pavitra Goyal 19BCE0373
* Samarthya Jha 19BCB0011
* Shatayu Mittal 19BCE0446

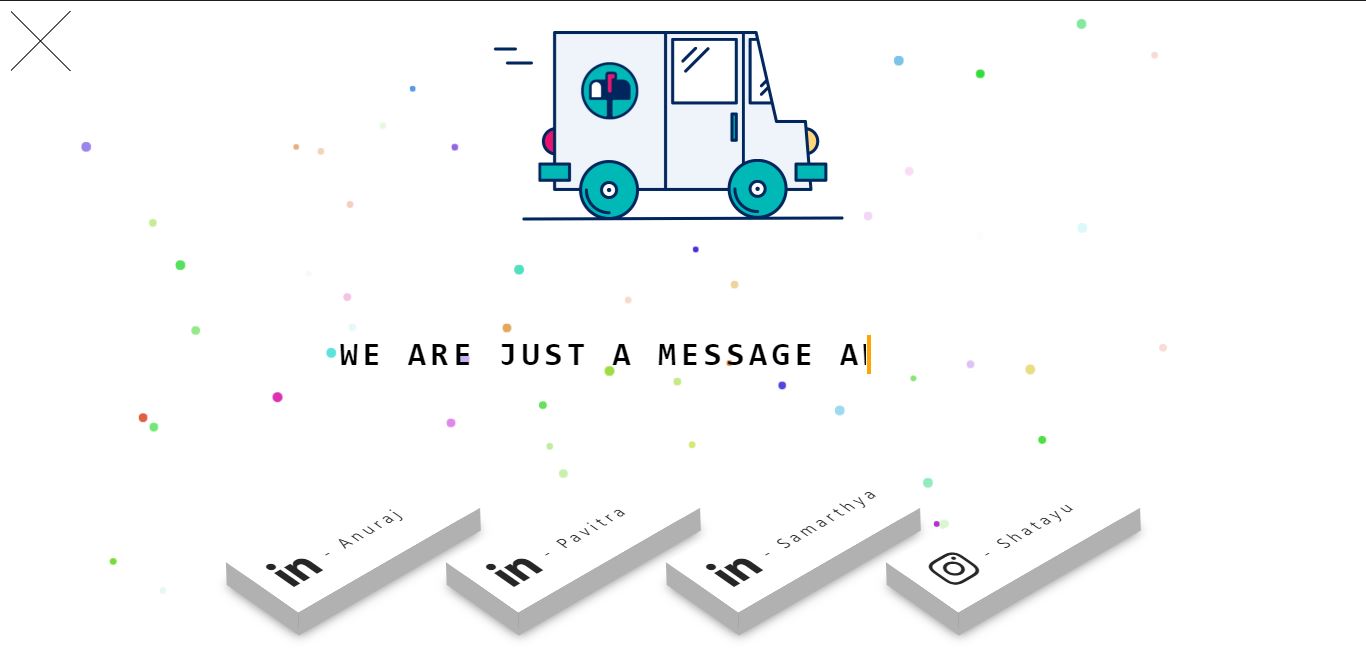
IMPLEMENTATION:

1. LANDING PAGE:

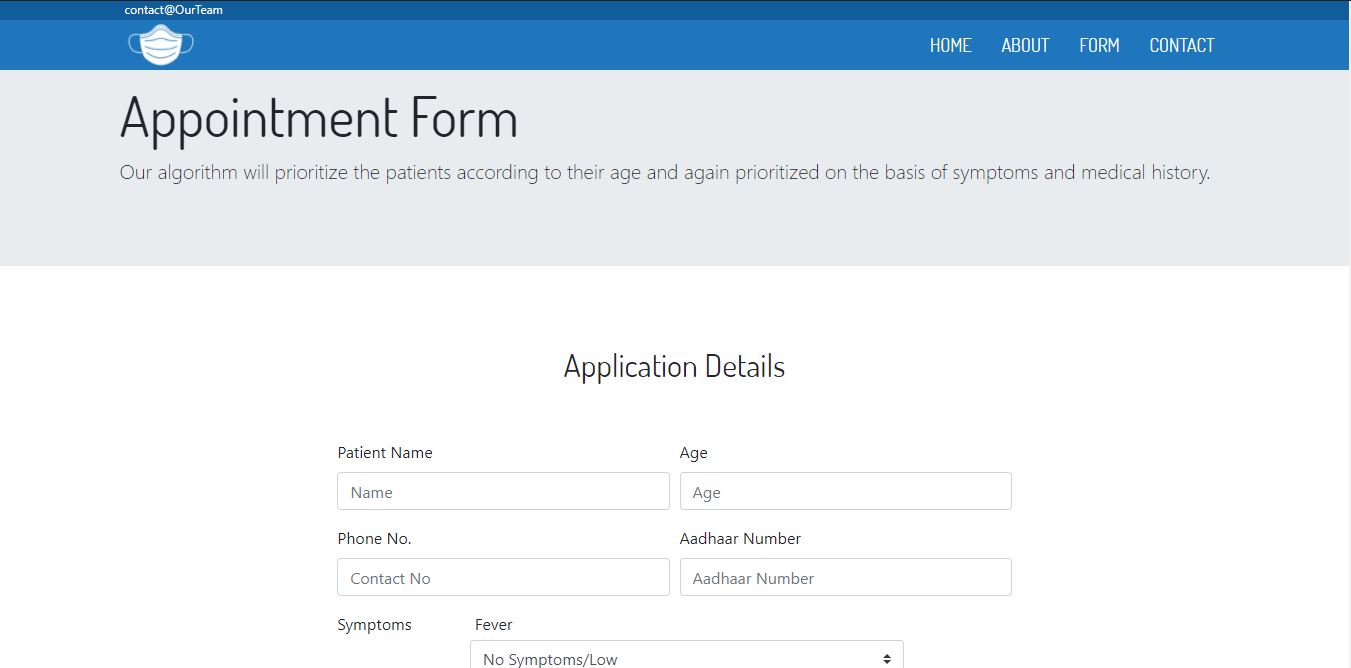
The Navigation Bar contains the firm icon and the HOME button with which the user can anytime reach the home page, ABOUT button which is the developers info page, a FORM dropdown menu which contains: 1. Book an appointment and 2. Know your appointment link, a CONTACT link with which the applicant can contact the developers in case of any issues. The page also contains a centre placed button which upon a click redirects to the form filling page.

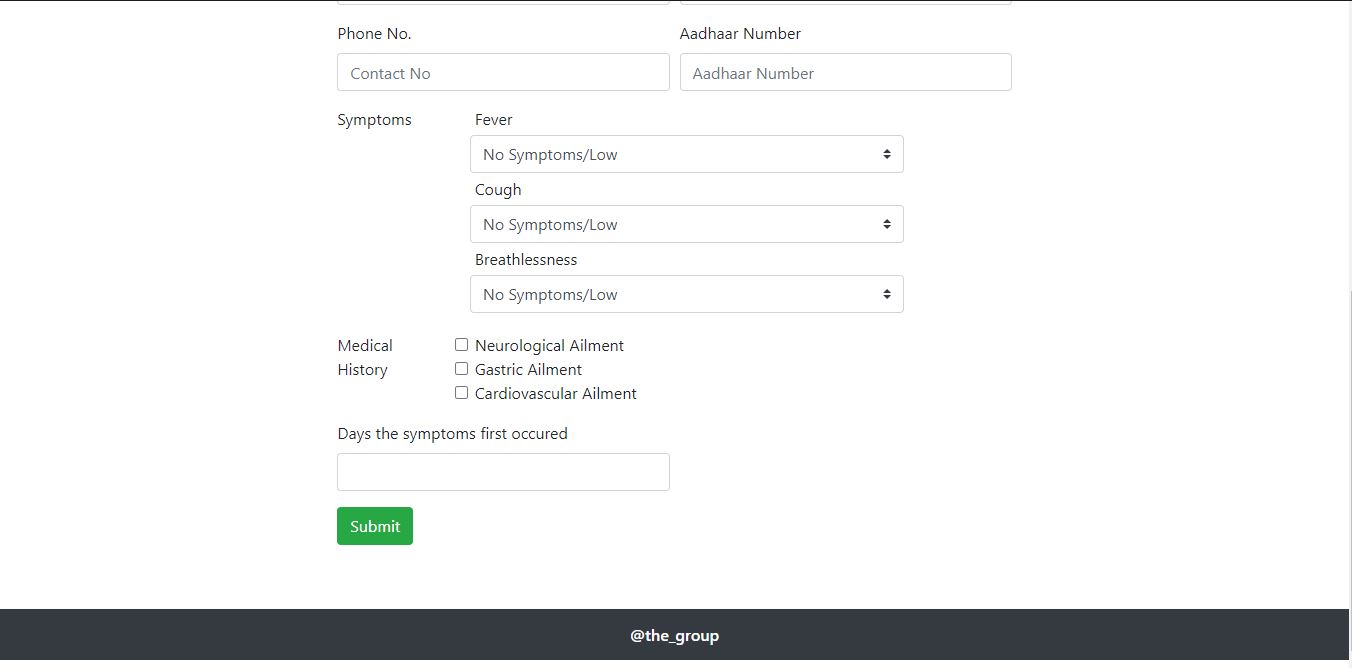
2. CONTACT PAGE:



The eye-catching UI of the page renders four 3-d rectangles which are hyperlinks to the developer’s business contact handles.

3. APPOINTMENT PAGE (FORM):

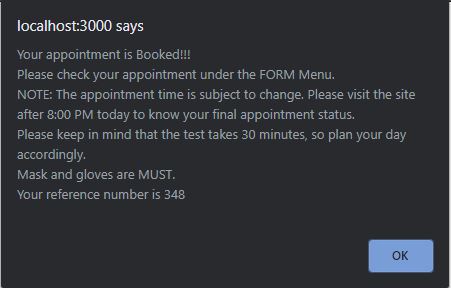




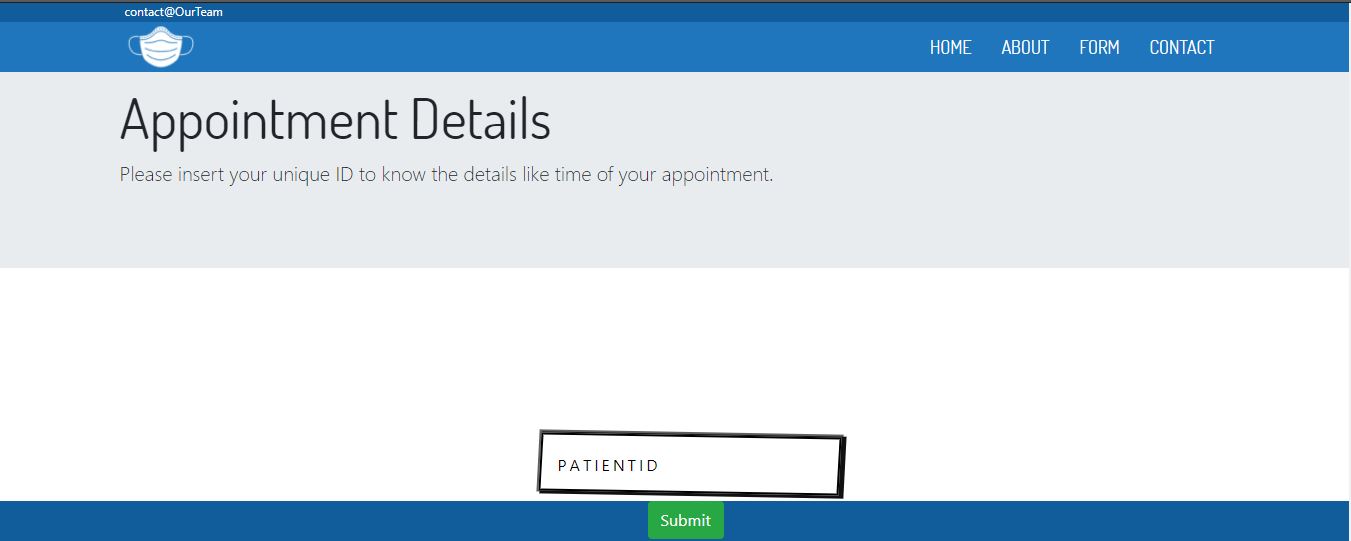
The above form fetches the medical details of the patient along with some important personal details such as Name, UID and Phone number. Once the submit button is clicked the form fields are validated and check for NULL values and upon successful validation the data in the form of a JSON object is sent to the database and duplicity of UID is checked and then a success message is displayed with an reference number if the UID is unique in the whole database otherwise an error message is displayed if any of the specified conditions turn out to be true

1. The Booking portal is not online yet

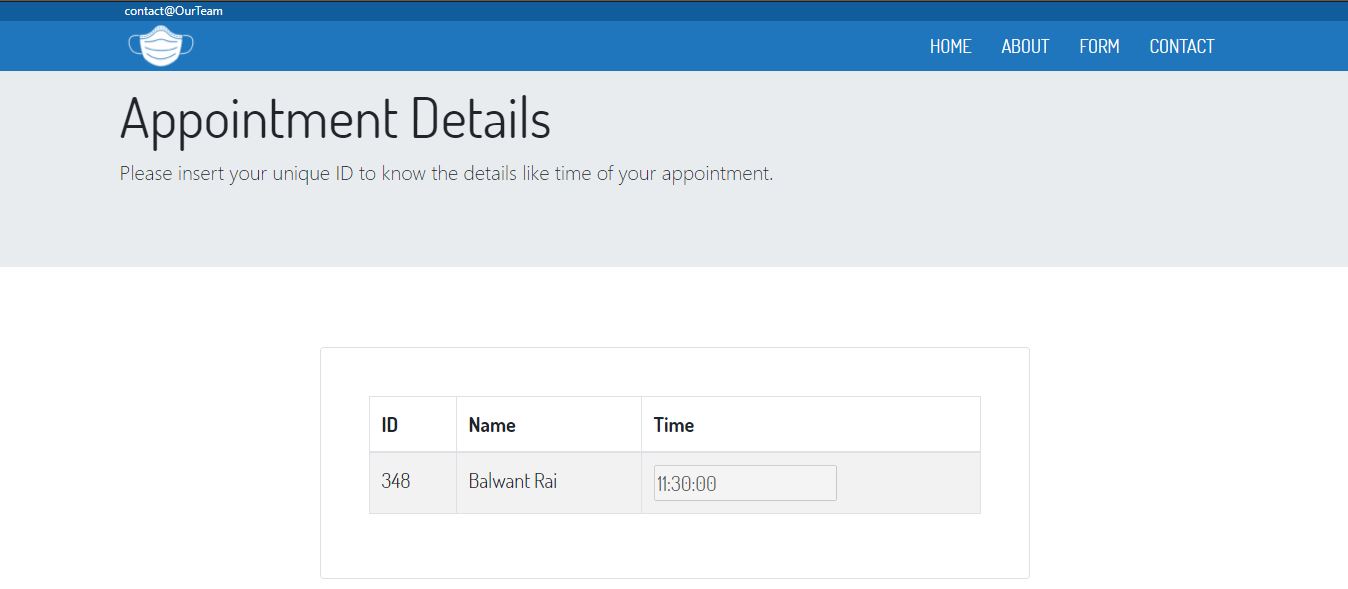
2. Dual Booking Attempt

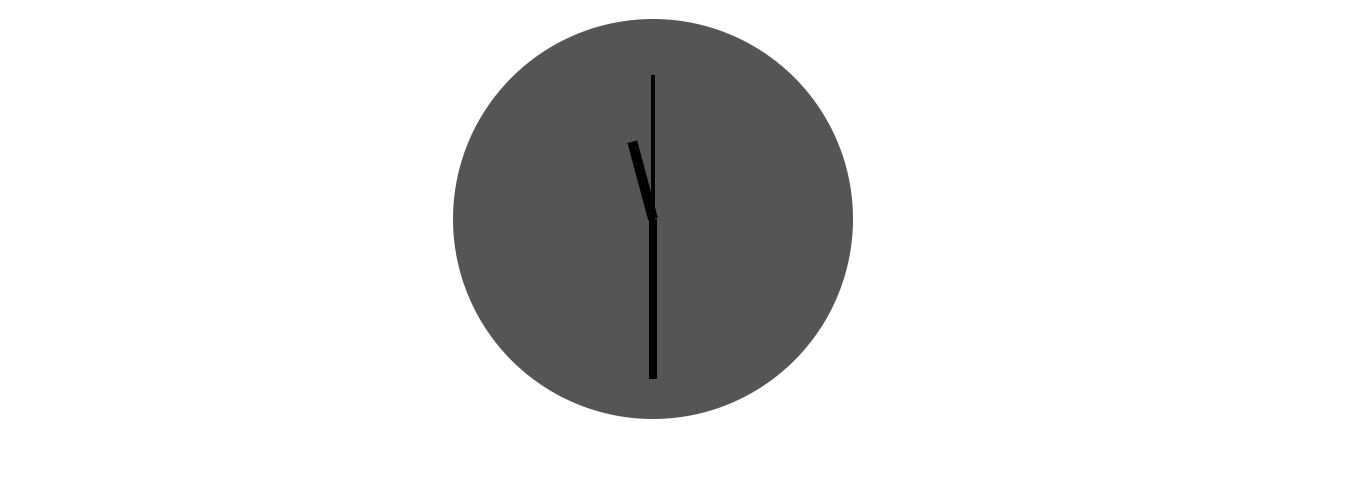


4. TIME SLOT DEATILS:



The know your appointment link brings the user to the above page and it contains a form field which expects the unique reference number assigned to the patient upon successful filling of the form. After the validation of the reference number in the database, the page displayed below is rendered else an error message is displayed.





**RESULT AND DISCUSSION**

Our portal has a user-friendly interface specifically designed to assist the patients to fill their details for the appointment of Covid-19 test and receive desired information about the appointment time. We kept account of the patient’s privacy by providing unique appointment key and implementing abstraction of other patient’s appointment details by only notifying the information of the concerned patient.

This project will help to create and maintain a database for the clinic staff reducing paper work to a bare minimum and thus reducing labour in managing those documents. It also provides flexibility of the database to the clinic staff to manage the appointments more efficiently. It prepares a fully accurate schedule from the information stored in the database based on the severity of the Covid-19 infection so that the most severe patients get treated first. This increases the efficiency of work flow in the clinic.

Our project runs on an algorithm which only allows a fixed number of patient’s appointment for Covid-19 test to get submitted which reduces the risk of any further spread of the infection and thus keeping account of social distancing. Our algorithm provides different time slots to each patient which will also reduce the risk for a non-infected patient having common symptoms of getting in contact of a severe infected patient, thus minimizing the risk of increase in the number of Covid-19 cases.

**CONCLUSION**

Using appointment rules, optimised using priority scheduling algorithm and round robin scheduling algorithm, new schedules for the clinic were produced with a focus on fairly reducing patient waiting times. A system to continually optimise future clinic schedules is demonstrated and tested. Optimising future schedules using symptoms, past medical history and age, the schedules are shown to perform well with some room for improvement.

Late-starts, high variance in appointment durations and the transition arrivals are the main factors causing long waiting times. Performing priority scheduling on basis of different factors like symptoms, past medical history and age, the schedule is improved.

In recent, we aim to include more factors as well as better scheduling algorithms to get more precise results and increase the efficiency of clinic. Better algorithm will also help us to manage large number of patients on every single day without any system overload. We also aim to develop a more user-friendly interface such that any age group can easily access the portal and understand their way through it.

We also plan to introduce a new administration portal for managing more than one test clinics which can be registered by giving their authorised medical licence. We can suggest the nearest test clinic according to patient’s residential address or landmark to save time and reducing the risk of contact with others keeping social distancing in mind.

In future, our project can be used not only for Covid-19 test but also for general or severe health conditions. The aim can be to generalise our project so that it can cover a large spectrum of different patients and their check-ups and appointments, maximizing efficiency of the whole hospital by providing an organised database for the same.

Through our project, patients can also involve in the health decisions that they have to make. They can make an appointment to another doctor other than theirs, by nothing more than a click. Thus, modernizing with the technology in the health sector of our society and tackling the problems faced in the pandemic.

**REFERENCES**

* <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports>
* <https://www.who.int/health-topics/coronavirus>
* <https://www.icmr.gov.in/pdf/covid/faqs/FAQs_English.pdf>
* Bailey, N. (1952). A study of queues and appointment systems in hospital out-patient departments, with special reference to waiting-times. Journal of the Royal Statistical Society. Series B (Methodological)
* Blanco White, M., & Pike, M. (1964). Appointment systems in outpatients' clinics and the effect of patients' unpunctuality. Medical Care

**APPENDIX:**

* BACKEND CODE:

|  |
| --- |
| var express=require("express");  var app=express();  var bodyparser=require("body-parser");  app.use(express.static("cssresources"));  app.use(bodyparser.urlencoded({extended:true}));  app.set("view engine","ejs");  // ====================>  // **Connecting to Database!!!**  const mongoose=require("mongoose");  const { stringify } = require("querystring");  const { Console } = require("console");  mongoose.connect('mongodb://localhost/demo\_db\_OS', {  useNewUrlParser: true,  useUnifiedTopology: true  })  .then(() => console.log('Connected to DB!'))  .catch(error => console.log(error.message));  //**Creating a user schema**  var userschema=new mongoose.Schema({  name: String,  age: Number,  Phone\_no: String,  Aadhaar\_no: Number,  Severity\_cough: Number,  Severity\_fever: Number,  Severity\_breathlessness: Number,  neurological\_history: Number,  cardiovascular\_history: Number,  gastric\_history: Number,  reference\_number: Number,  Booking\_time: String  });  var User=mongoose.model("User",userschema);  // =================!  // ===============>  // **Routes:::**  app.get("/",function(request,response){  var tt="";  response.render("landing",{tt});  });  app.get("/developerinfo",function(request,response){  response.render("developers");  });  //**This route will open the form filling form**  app.get("/openform",function(request,response){  var currtime=new Date().toLocaleTimeString();  // console.log(currtime);  var currhr=parseInt(currtime[0]+currtime[1]);  var currmin=currtime[2]+currtime[3];  var currmerr=currtime[currtime.length-2]+currtime[currtime.length-1];  currmin=parseInt(currmin);  User.countDocuments({},function(err,counter){  if (err){  console.log(err);  }  else{  if (counter<=19 && (currhr>=4 && currhr<=11 && currmerr=="PM")){  response.render("form");  }  else{  var tt="";  if (counter>=20){  tt="The Appointment queue is full!\nPlease try in some other centres or try again tommorow";  }  else if (currhr<=4 || currmerr=='AM'){  tt="Booking starts at 4:00 PM";  }  else if (currhr>=10){  tt="You are late!!! Booking ended at 7:00 PM.\nPlease try in some other centres or try again tommorow.";  }  response.render("landing",{tt});  }    }  })  });  //This route will render the contacts page  app.get("/contact",function(request,response){  response.render("contact");  });  //This route will render the form where the applicant can check the status of the //appointment  app.get("/check",function(request,response){  response.render("check\_status");  });  //**This is a post route that parses the information submitted by the user and send the** //**data to the database.**  app.post("/appointment",function(request,response){  // console.log(request.body);  var degree\_fever=0;  var degree\_cough=0;  var degree\_breathlessness=0;  var iscardio=0;  var isgastric=0;  var isneuro=0;  degree\_fever=request.body.selected\_degree\_fever;  degree\_cough=request.body.selected\_degree\_cough;  degree\_breathlessness=request.body.selected\_degree\_breathlessness;  let check\_neuro=request.body['history\_brain'];  if (check\_neuro!=undefined){  isneuro=1;  }  let check\_cardio=request.body['history\_cardio'];  if (check\_cardio!=undefined){  iscardio=1;  }  let check\_gastric=request.body['history\_gastric'];  if (check\_gastric!=undefined){  isgastric=1;  }  var newuser={  name: request.body.Patient\_name,  age: request.body.Patient\_age,  Phone\_no: request.body.Patient\_phone,  Aadhaar\_no:request.body.UID,  Severity\_cough: degree\_cough,  Severity\_fever: degree\_fever,  Severity\_breathlessness: degree\_breathlessness,  neurological\_history: isneuro,  cardiovascular\_history: iscardio,  gastric\_history: isgastric,  reference\_number: Math.floor((Math.random() \* 1000) + 1),  Booking\_time: new Date().toLocaleTimeString()  };  var tt="";  var refno=newuser.reference\_number;  User.find({Phone\_no:newuser.Aadhaar\_no},function(err,samedata){  if (err){  console.log("You have already booked an appointment!!!");  }  else{  if (samedata.length>=1){  tt="You have already booked an appointment!!!";  response.render("landing",{tt});  }  else{  console.log(newuser);  User.create(newuser,function(err,user){  if (err){  tt="Sorry for the inconvenience caused.Please try again !!!"  }  else{  tt="Your appointment is Booked!!! \nPlease check your appointment under the FORM Menu.\nNOTE: The appointment time is subject to change. Please visit the site after 8:00 PM today to know your final appointment status.\nPlease keep in mind that the test takes 30 minutes, so plan your day accordingly.\nMask and gloves are MUST.\nYour reference number is "+refno;  response.render("landing",{tt});  }  })  }  }  })  })  //**This route will render the time slot on the basis of the score calculated.**  app.get("/finaloutput",function(request,response){  var qq=parseInt(request.query.tellme);  User.find({reference\_number:qq},function(err,records){  if (err){  console.log(err);  }  else{  if (records.length==0){  var tt="Invalid Reference Id!.\n Please try again.";  response.render("landing",{tt})  }  else{  User.find({},function(err,allrecords){  if (err){  console.log(err);  }  else{  console.log(allrecords);  response.render("check\_status2",{data:allrecords,prefno:qq});  }  });  }  }  });  });  //**Code for starting the server**  app.listen(process.env.PORT || 3000,process.env.IP,function()  {  console.log("Server started!!!");  }); |

* Algorithmic code in JavaScript:

|  |
| --- |
| //**input will come as an array of objects**  const prev\_medical\_history=200.0;  //**function will return the total score**  function totscore(age\_score,symptoms\_score,medical\_history\_score){  return ((0.4)\*symptoms\_score+(0.3)\*age\_score+(0.3)\*medical\_history\_score);  }  //**function will return the age score**  function get\_age\_score(age){  var retval=0.0;  if (age<=13){  return (14.0-age)\*(3.0);  }  if (age>13 && age<=50){  return 1.0;  }  if (age>50){  return (age-51)\*(2.0);  }  return retval;  }  //**function will return the symptoms score**  function get\_symptoms\_score(severity\_cough,severity\_fever,severity\_breathlessness){  var estimate=0.0;  var symptomsscore=[0.0,100.0,200.0];  estimate=estimate+((0.5)\*(symptomsscore[severity\_breathlessness-1]));  estimate=estimate+((0.3)\*(symptomsscore[severity\_fever-1]));  estimate=estimate+((0.2)\*(symptomsscore[severity\_cough-1]));  return estimate;  }  //**function will return the priority score**  function get\_history\_priority(neuro\_history,cardio\_history,gastric\_history){  var ret=0.0;  if (neuro\_history==1){  ret=ret+((0.2)\*prev\_medical\_history);  }  if (gastric\_history==1){  ret=ret+((0.3)\*prev\_medical\_history);  }  if (cardio\_history==1){  ret=ret+((0.5)\*prev\_medical\_history);  }  return ret;  }  var details=data;  var static\_scores=[];  for(var i=0;i<=details.length-1;i++){  var subscores=[];  subscores.push(details[i].reference\_number);  var age\_priority=get\_age\_score(parseFloat(details[i].age));  var symptoms\_priority=get\_symptoms\_score(details[i].Severity\_cough,details[i].Severity\_fever,details[i].Severity\_breathlessness);  var history\_priority=get\_history\_priority(details[i].neurological\_history,details[i].cardiovascular\_history,details[i].gastric\_history);  var final\_score=totscore(age\_priority,symptoms\_priority,history\_priority);  subscores.push(final\_score);  subscores.push(details[i].reference\_number);  subscores.push(details[i].Booking\_time);  subscores.push(details[i].name);  static\_scores.push(subscores);  }  //**This comparator function will a Boolean value according to the score and** //**appointment time**  function comparator(a,b){  if (a[0]==b[0]){  if (a[2][0]!=b[2][0]){  var ha=parseInt(a[2][0]);  var hb=parseInt(b[2][0])  return (a[2][0]<=b[2][0]-1)?-1: 1;  }  if (a[2][0]==b[2][0]){  //7:18:18  var mina=a[2][2]+a[2][3];  mina=parseInt(mina);  var minb=b[2][2]+b[2][3];  minb=parseInt(minb);  if (mina!=minb){  return (mina<=minb-1)?-1: 1;  }  if (mina==minb){  var seca=a[2][5]+a[2][6];  seca=parseInt(seca);  var secb=b[2][5]+b[2][6];  secb=parseInt(secb);  return (seca<=secb)?-1:1;  }  }  }  if (a[0]!=b[0]){  return (a[0]>b[0])?-1: 1;  }  }  static\_scores.sort(comparator);  var refno=prefno;  var ind;  for(var i=0;i<=static\_scores.length-1;i++){  if (static\_scores[i][1]==refno){  ind=i+1;  break;  }  }  var hr=10;  var min=0;  for(var i=2;i<=ind;i++){  if (i%2==0){  min=30;  }  if (i%2!=0){  hr+=1;  min=0;  }  }  if (ind>=7){  hr-=12;  }  var finaltime="";  finaltime+=hr;  finaltime+=':';  if (min==0){  finaltime+='00';  }  if (min!=0){  finaltime+=min;  }  finaltime+=':00';  if (ind>=5){  finaltime+=' PM';  }  if (ind<5){  finaltime+=' AM';  } |