Title: Health Management System for VNIT Nagpur

Team: BT23CSE038 SHUBHAM KUMAR

BT23CSE097 PUSHKAR LADDA

BT23CSE107 NITIN BHATT

BT23CSE122 AAYAN SAHEB

BT23CSE126 RONAK SINGH

---

1. Problem Statement: Design a Health Management System (HMS) for the VNIT Nagpur Health Center to efficiently manage student health records, appointments, doctor availability, emergency alerts, and notifications. It must also include support for real-world healthcare needs and be integrated with a robust MySQL database backend.

---

2. Pain Points:

- Manual record keeping and appointment scheduling.
- Difficulty managing doctor leaves and availability.
- Lack of emergency response mechanism.

---

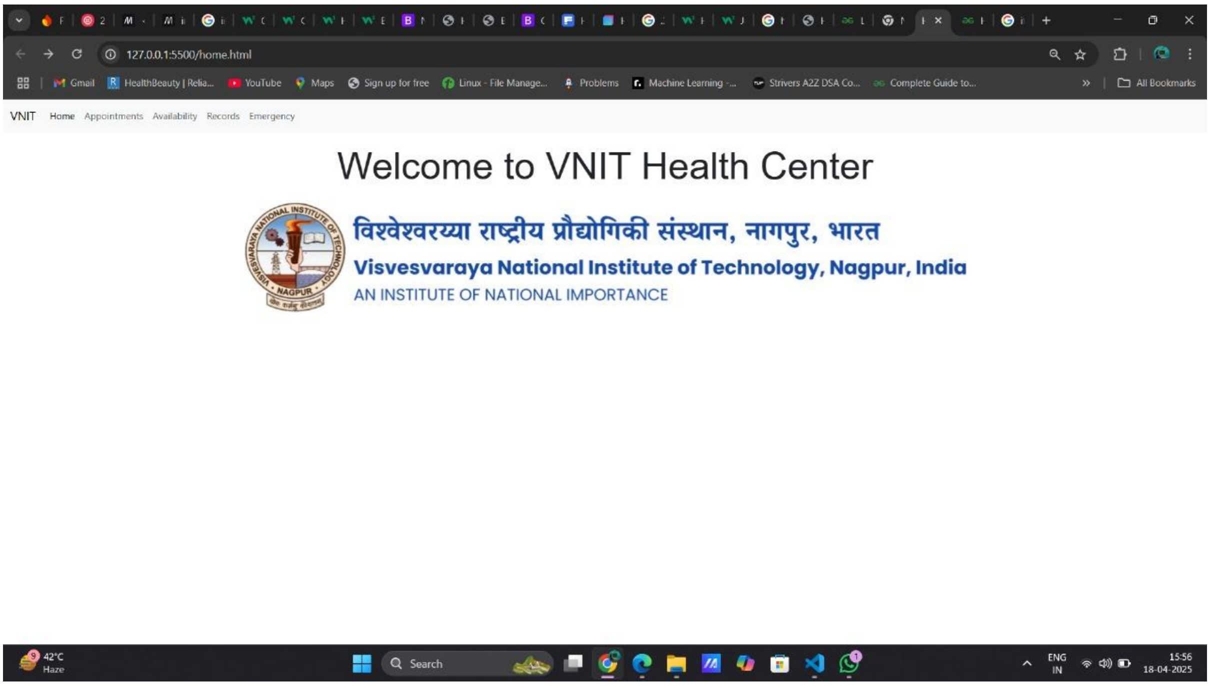
3. Solution:

- A web-based HMS with frontend and backend.
- MySQL database with entities like Students, Doctors, Appointments, etc.
- Triggers and subqueries to automate processes and maintain data integrity.
- Pages: Home, Appointment Booking, Doctor Availability, Health Records, Emergency Alerts, User Account.
- Enhancements: Reminder notifications, analytics, and mental health resources.

---

4. UI (Screenshots/Sketches):

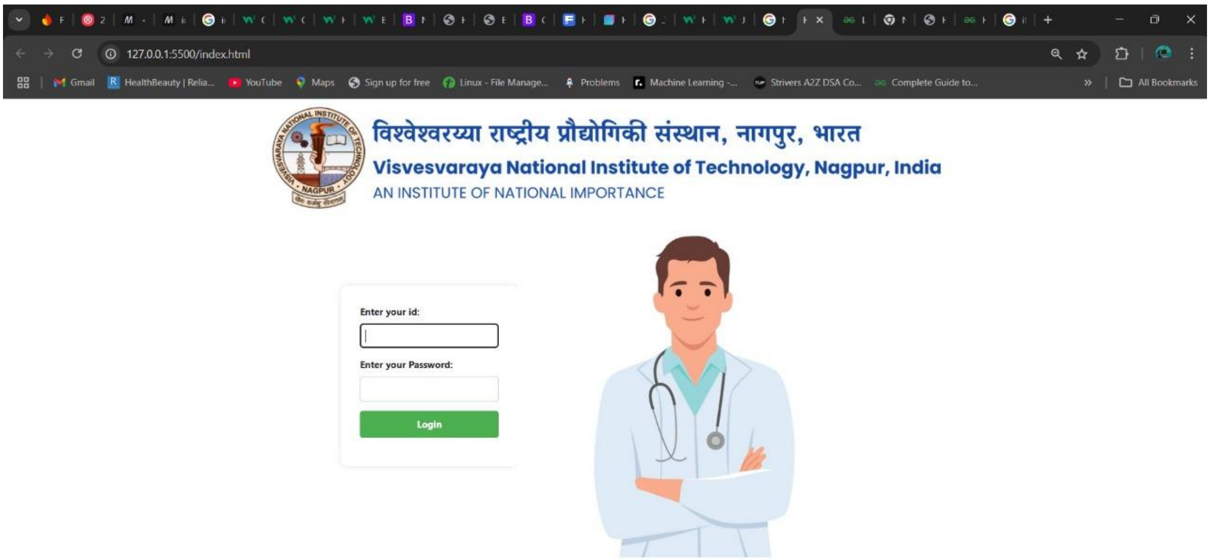
- Home Page



Welcome to VNIT Health Center

विश्वेश्वरय्या राष्ट्रीय प्रौद्योगिकी संस्थान, नागपुर, भारत
Visvesvaraya National Institute of Technology, Nagpur, India
AN INSTITUTE OF NATIONAL IMPORTANCE

-
-
-
- Login page

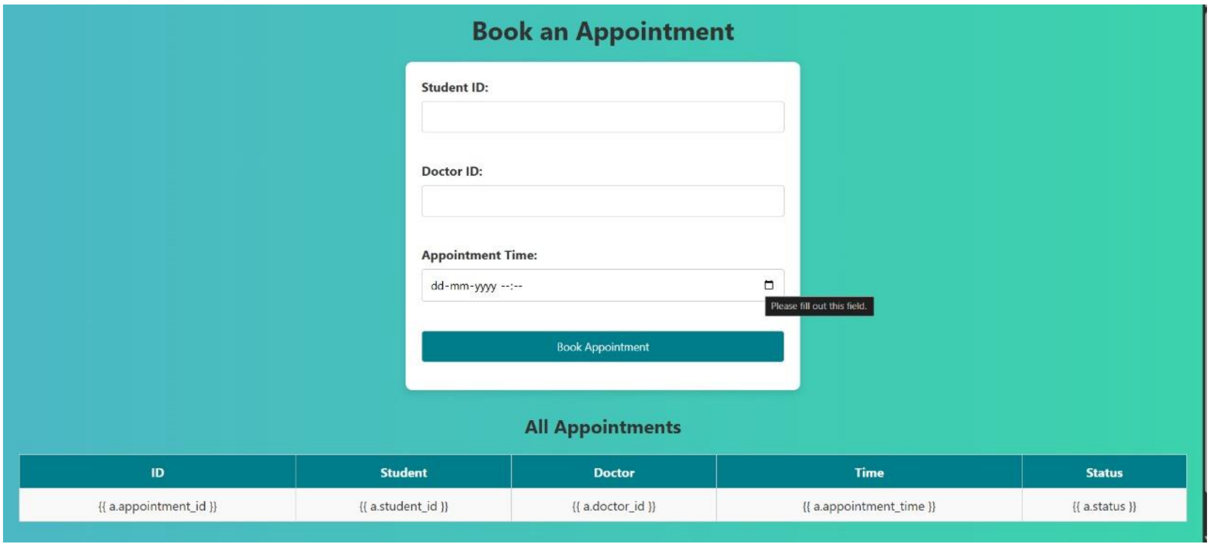


विश्वेश्वरय्या राष्ट्रीय प्रौद्योगिकी संस्थान, नागपुर, भारत
Visvesvaraya National Institute of Technology, Nagpur, India
AN INSTITUTE OF NATIONAL IMPORTANCE

Enter your id:

Enter your Password:

-
- Appointment Booking Page



Book an Appointment

Student ID:

Doctor ID:

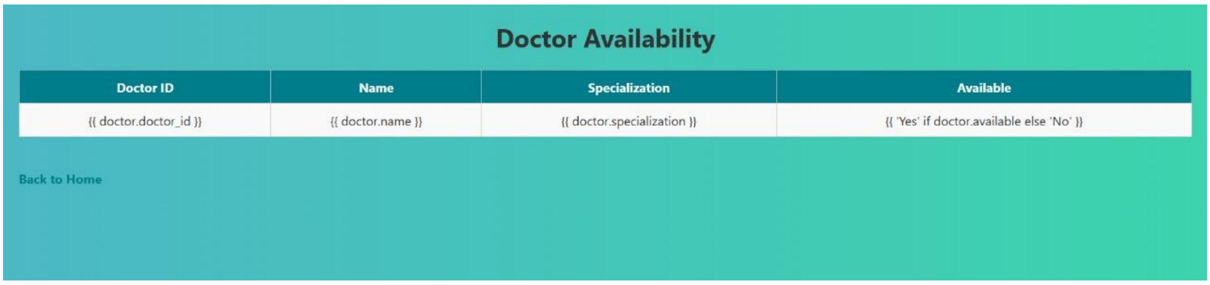
Appointment Time:

dd-mm-yyyy --:--

All Appointments

ID	Student	Doctor	Time	Status
{{ a.appointment_id }}	{{ a.student_id }}	{{ a.doctor_id }}	{{ a.appointment_time }}	{{ a.status }}

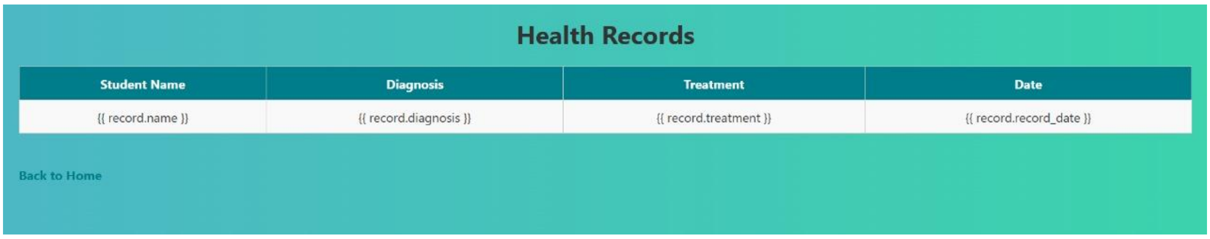
- Doctor Availability Page



Doctor Availability

Doctor ID	Name	Specialization	Available
{{ doctor.doctor_id }}	{{ doctor.name }}	{{ doctor.specialization }}	{{ 'Yes' if doctor.available else 'No' }}

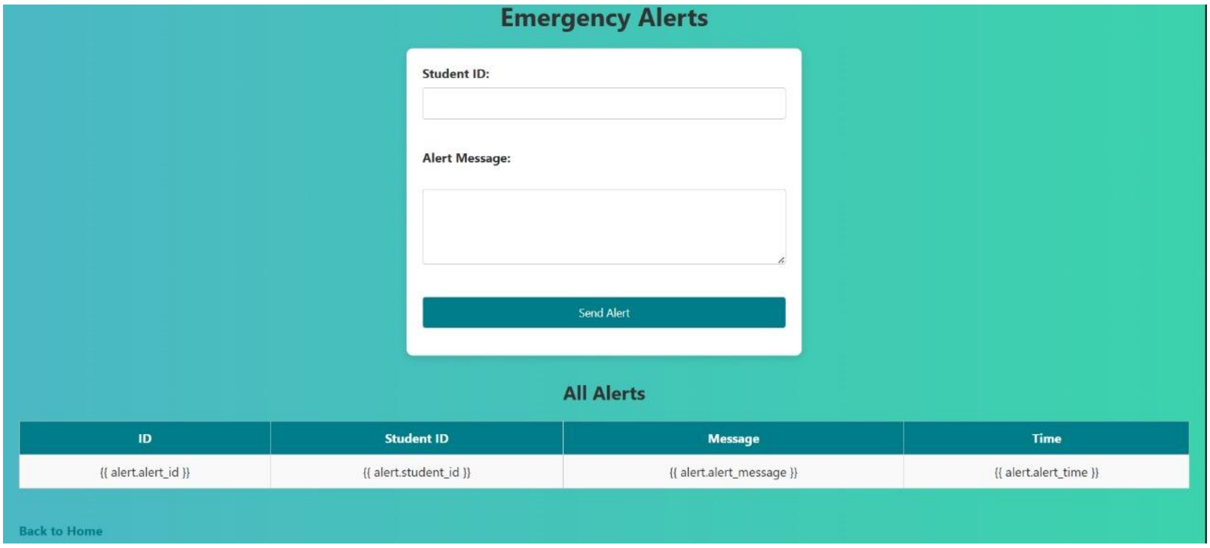
- Health Records



Health Records

Student Name	Diagnosis	Treatment	Date
{{ record.name }}	{{ record.diagnosis }}	{{ record.treatment }}	{{ record.record_date }}

- Emergency Alerts Page



Emergency Alerts

Student ID:

Alert Message:

All Alerts

ID	Student ID	Message	Time
{{ alert.alert_id }}	{{ alert.student_id }}	{{ alert.alert_message }}	{{ alert.alert_time }}

---

5. Database:

- **Entities:** Students, Doctors, Appointments, Health Records, Emergency Alerts, Notifications
- **Relationships:**
  - Appointments linked to Students and Doctors (foreign keys)
  - Health Records linked to Students
  - Emergency Alerts can be global or linked to individuals
- **Queries:**
  - Joins to fetch appointment details with student and doctor info.
  - Subquery: Fetch next available appointment for a doctor.
- **Trigger:**
  - Automatically update doctor availability on appointment booking.

---

6. Integration:

- Example:
  - Appointment Booking → Inserts into Appointments table → Triggers doctor availability update
  - Viewing Records → SELECT with JOIN on Students and Health Records

- Trigger Role:
  - Ensure real-time doctor availability updates.
  - Maintain consistency of notification alerts.

7. Testing (Prototype): Tested backend and all functionalities of using flask and mysql ,impleneted frontend using html and css.

8. Challenges: integrating backend and frontend together was challenging,

Implementing mysql database with flask as backend was quite challenging

9. Future Improvements: implementing project on large scale .

Making website more dynamic and responsive,

Increasing security and user encryption

Updation of the data dynamically

GitHub Repository: [Add Link Here]

Submitted on: 18/04/25