NOIDA INSTITUTE OF ENGG & TECHNOLOGY

DEPARTMENT OF CYBERSECURITY

PROJECT REPORT

Name of the Candidate : Himanshu Kumar Singh

Class: : CYS-3rd Year

ROLL NO : 2201331720033

Year: 2023-2024

NOIDA INSTITUTE OF ENGG & TECHNOLOGY

DEPARTMENT OF CYBERSECURITY

BONAFIDE CERTIFICATE

This is to certify that Himanshu Kumar Singh of class Cyber-Security has done a project work on Ticket chat-bot during the year 2023-2024 at the **NOIDA INSTITUTE OF ENGG & TECHNOLOGY**

Guide-In- Charge

HOD

CYBERSECURITY

Dated: ……………..

Acknowledgement

I would like to express my heartfelt gratitude to everyone who supported me in the development of this ticket booking web page for the museum.
I am especially thankful to [Sumit Sir] for their invaluable guidance and encouragement throughout this project. Their expertise and insights have been instrumental in helping me design and implement the features effectively.

I am grateful for the availability of open-source tools and resources such as [List relevant technologies, e.g., HTML, CSS, JavaScript, React, Bootstrap], which greatly facilitated the development process.
This project has been a significant learning experience, and I am deeply thankful to everyone who contributed to its success.

Theory of project

Theory

The main goal of this project is to create a web-based platform where users can easily book tickets to visit a museum. The system simplifies the ticket booking process by allowing users to select dates, the number of tickets, and make payments online.

This project is based on the following concepts:

1. Front-End Development:
   - The user interface is designed to be simple, attractive, and easy to use. Technologies like HTML, CSS, and JavaScript are used to create the design and interactivity of the web page.
   - Responsive design ensures that the web page works well on different devices, like computers, tablets, and smartphones.
2. Back-End Development:
   - The back-end handles the functionality of the system, like storing user details, processing bookings, and managing ticket availability.
   - A server-side language like PHP, Python (Django/Flask), or Node.js is used, along with a database like MySQL or MongoDB, to store information.
3. Database Management:
   - A database is used to keep track of bookings, available tickets, and user information securely. This ensures that the system is accurate and efficient.

4. Payment Integration:
    - The system integrates a payment gateway to allow users to pay for tickets online securely. This makes the process more convenient for users.
5. User Experience:
    - The project focuses on creating a smooth and hassle-free experience for users. Features like confirmation messages and the ability to view ticket details enhance the overall usability.

By combining these components, **the** project creates a functional, user-friendly, and secure ticket booking platform for museums.

Hardware and software requirements

Hardware Requirements

1. For Development:
   - Processor: Intel i3 or above
   - RAM: 4 GB or more
   - Storage: 20 GB free space
   - Operating System: Windows/Linux/Mac
2. For Deployment:
   - Server:
     - Processor: Dual-core or higher
     - RAM: 8 GB or more
     - Storage: 50 GB free space
     - Internet Connection: Stable with sufficient bandwidth
3. For Users:
   - Any device with internet access (PC, smartphone, tablet)
   - Browser: Chrome, Firefox, Edge, etc.
   -

Software Requirements

1. For Development:
   - Code Editor: Visual Studio Code, Sublime Text, or similar
   - Frameworks: React.js/Angular.js (for front-end), Node.js/Django/Flask (for back-end)
   - Database: MySQL/MongoDB/PostgreSQL
   - Version Control: Git/GitHub

2. For Deployment:
    - Web Server: Apache/Nginx
    - Hosting: AWS, Heroku, or any hosting service
    - Database Server: MySQL Server or equivalent
3. Other Tools:
    - Browser for Testing: Chrome/Firefox with developer tools
    - Payment Gateway API: PayPal, Stripe, or Razorpay.

Source code

HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>Online Ticket Making System</title>
   <link rel="stylesheet" href="styles1.css">
</head>
<body>
   <header>
      <h1>Online Ticket Making System</h1>
   </header>

   <main>
      <section class="chatbot">
         <h2>Chatbot</h2>
         <input type="text" id="user-input"
placeholder="Type a message...">
         <button id="send-btn">Send</button>
         <div id="chatbot-response"></div>
      </section>
      <section class="ticket-booking">
         <h2>Book a Ticket</h2>
         <form id="ticket-form">
            <label for="name">Name:</label>
            <input type="text" id="name"
```

```
name="name"><br>



            <label for="State">State:</label>
            <select name="state" id="state">
            <option value="Select State">Select
State</option>
            <option value="Bihar">Bihar</option>
            <option value="Gujrat">Gujrat</option>
            <option value="Uttar Pradesh">Uttar
Pradesh</option>
            <option value="MP">MP</option>
            <option value="KERALA">KERALA</option>
            <option value="Arunachal Pradesh">Arunachal
Pradesh</option>
            <option
value="Uttarakhand">Uttarakhand</option>
            <option value="Punjab">Punjab</option>
            <option
value="Chattishgarh">Chattishgarh</option>
            <option value="Haryana">Haryana</option>
            <option value="Odisha">Odisha</option>
            <option value="Delhi">Delhi</option>
            <option value="West Bengal">West
Bengal</option>
            <option
value="Ahmadabad">Ahmadabad</option>
            <option value="Gujrat">Gujrat</option>
            <option value="Assam">Assam</option>
            <option value="Andhra Pradesh">Andhra
Pradesh</option>
            <option value="Mumbai">Mumbai</option>
            <option value="Goa">Goa</option>
            <option
value="Hyderabad">Hyderabad</option>
            <option value="Jharkhand">Jharkhand</option>
            <option value="J&k">J&k</option>
            <option value="Karnataka">Karnataka</option>
            <option value="Manipur">Manipur</option>
            <option value="Meghalaya">Meghalaya</option>
            <option value="Mizoram">Mizoram</option>
            <option value="Bangalore">Bangalore</option>
            <option value="Nagaland">Nagaland</option>
          </select><br><br>
            <section class="Museum"></section>
            <label for="Museum">Museum</label>
            <select id="museum" name="museum" id="">
              <option value="Select Museum">Select
Museum</option>
              <option value="kolkata museum">Kolkata
Museum</option>
              <option value="Natonal museum">National
Museum Delhi</option>
              <option value="Prince of wales
Museum,Delhi">Prince of wales Museum,Delhi</option>
              <option value="Calico Museum of
Textiles,Ahmadabad">Calico Museum of
Textiles,Ahmadabad</option>
              <option value="Salar Jung Museum,
Hyderabad">Salar Jung Museum, Hyderabad</option>
              <option value="Government Museum,
Chennai">Government Museum, Chennai</option>
              <option value="HAL Heritage Centre and
Aerospace Museum, Bengaluru">HAL Heritage Centre and
Aerospace Museum, Bengaluru</option>
```

```
          <option value="Darshan Museum">Darshan 
Museum</option>
          <option value="Napier Museum">Napier 
Museum,Thiruvananthapuram</option>
          <option value=" The Partion Museum"> The 
Partion Museum,Amritsar</option>
          <option value="Sarnath Museum">Sarnath 
Museum,Varanasi</option>
          <option value="Patna Museum">Patna Museum, 
Patna</option>
          <option value="Jehangir Art Gallery">Jehangir 
Art Gallery,Mumbai</option>
        </select><br>
        <br>
        <label for="time-slot">Time Slot:</label>
        <select id="time-slot" name="time-slot">
          <option value="Select the Time-Slot">Select the 
Time-Slot</option>
          <option value="10 to 12 AM">10 to 12 
AM</option>
          <option value="12 to 02 PM">12 to 2 
PM</option>
          <option value="02 to 04 PM">02 to 04 
PM</option>
          <option value="04 to 06 PM">04 to 06 
PM</option>

      </select><br>

      <br>
        <button id="book-btn">Book Ticket</button>
      </form>
    </section>
    <section class="payment">
      <h2>Make a Payment</h2>
      <form id="payment-form">
        <label for="payment-mode">Payment
Mode:</label>
        <select id="payment-mode" name="payment-
mode">
          <option value="Select Payment Mode">Select
Payment Mode</option>
          <option value="Credit Card">Credit
Card</option>
          <option value="Debit Card">Debit
Card</option>
          <option value="PayPal">PayPal</option>
        </select><br><br>
        <button id="pay-btn">Make Payment</button>
      </form>
    </section>
  </main>
  <!-- <script src="script.js"></script>
  <script src="server.js"></script>
  <script src="script2.js"></script> -->
</body>
</html>
```

CSS:-

```
body {
   font-family: Arial, sans-serif;
   background-color: #c1f3f5;

}

header {
   background-color: #333;
   color: #fff;
   padding: 1em;
   text-align: center;
}

image{
   width: 50;
   height: 50;
}

main {
   display: flex;
   flex-direction: column;
   align-items: center;
   padding: 2em;
}

.chatbot {
   background-color: #c6c1c1;
   padding: 1em;
   border: 20px solid #ddd;
   box-shadow: 10px 10px 30px;
   margin-bottom: 2em;
```

```
}

.ticket-booking {
   background-color: #eeb8b8;
   padding: 1em;
   border: 10px solid #ddd;
   box-shadow: 10px 10px 30px;
   margin-bottom: 2em;
}

.payment {
   background-color: #f8969682;
   padding: 1em;
   border: 1px solid #ddd;
   box-shadow: 10px 10px 20px;
}

#user-input {
   width: 100%;
   padding: 0.5em;
   font-size: 1em;
}

#send-btn {
   background-color: #333;
   color: #fff;
   padding: 0.5em 1em;
   border: none;
   border-radius: 0.25em;
   cursor: pointer;
}

#send-btn:hover {
  background-color: blue;
}

#chatbot-response {
  padding: 0.5em;
}

#ticket-form {
  display: flex;
  flex-direction: column;
  align-items: center;
}

#name, #time-slot {
  width: 100%;
  padding: 0.5em;
  font-size: 1em;
  margin-bottom: 0.5em;
}

#book-btn {
  background-color: rgb(60, 60, 60);
  color: white;
  padding: 0.5em 1em;
  border: none;
  border-radius: 0.25em;
  cursor: pointer;
}

#book-btn:hover {
  background-color: blue;
}
```

```
#payment-form {
   display: flex;
   flex-direction: column;
   align-items: center;
}

#payment-mode {
   width: 100%;
   padding: 0.5em;
   font-size: 1em;
   margin-bottom: 0.5em;
}

#pay-btn {
   background-color: #333;
   color: white;
}
#pay-btn:hover {
   background-color: blue;
}
.state {
```

Input and outputs

Inputs

1. User Inputs
   - Visitor's personal details (name, email, phone number).
   - Ticket selection (date, time, number of tickets, type of ticket – adult, child, senior, etc.).
   - Payment information (credit/debit card details, online wallet, or payment service).
2. Technical Inputs
   - Design specifications (UI/UX wireframes, website layout, and responsive design requirements).
   - Web technologies (HTML, CSS, JavaScript, backend frameworks like Python/Django, Node.js, etc.).
   - Database schema (visitor information, ticket records, payment records).
   - APIs (payment gateway APIs like Stripe, PayPal, or Razorpay for processing transactions).
3. Operational Inputs
   - Museum-specific details (ticket pricing, schedules, available slots, policies).
   - Server and hosting requirements for the webpage.

  - Security measures (SSL certificates, data encryption for payment).

Outputs

1. Functional Outputs
   - A responsive and user-friendly webpage that allows users to book tickets online.
   - Confirmation of successful ticket booking (on-screen message or email with ticket details).
   - Payment receipt and invoice generation.
2. Data Outputs
   - Storage of visitor details and booking information in the database.
   - Real-time availability updates for ticket inventory.
3. User Experience Outputs
   - E-tickets delivered via email or downloadable from the website.
   - Secure payment processing with confirmation messages for successful or failed transactions.
4. Administrative Outputs
   - Backend dashboard for museum staff to monitor ticket sales and manage bookings.
   - Reports on ticket sales, visitor statistics, and revenue.

References

Web Development Technologies

- W3Schools: "HTML, CSS, and JavaScript Tutorials." https://www.w3schools.com
- Mozilla Developer Network (MDN): "Web Development Documentation and Best Practices." https://developer.mozilla.org

Payment Integration

- Stripe API Documentation: "Secure Payment Gateway Integration." https://stripe.com/docs
- PayPal Developer Resources: "Integrating PayPal with Web Applications." https://developer.paypal.com

Database Management

- MySQL Documentation: "Database Design and Integration." https://dev.mysql.com/doc/
- MongoDB Documentation: "Using NoSQL Databases for Web Applications." https://www.mongodb.com/docs

UI/UX Design

- Nielsen Norman Group: "Principles of User-Centered Design." https://www.nngroup.com
- Material Design by Google: "Design Guidelines for Web Applications." https://material.io/design

Hosting and Deployment

- AWS (Amazon Web Services): "Hosting Web

Applications."
https://aws.amazon.com

- Netlify Documentation: "Deploying Static Websites Easily."
https://www.netlify.com/docs/

Cybersecurity and Data Protection

- OWASP: "Web Security Best Practices."
https://owasp.org
- General Data Protection Regulation (GDPR): "Ensuring Privacy Compliance."
https://gdpr-info.eu

Case Studies and Related Projects

- The Met Museum: "Digital Ticketing Platform Overview."
https://www.metmuseum.org
- Louvre Museum Online Tickets: "Implementation of Online Ticketing Systems."
https://www.louvre.fr/en