KONGU ENGINEERING COLLEGE

PERUNDURAI

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE : 22ITL42

COURSE NAME : WEB TECHNOLOGY LABORATORY

Visitor Management System

SUBMITTED BY

GIRIDHARAN S

22ITR025

IT – A

OBJECTIVE:

To enhance convenience and security by providing a user-friendly platform that allows organizations to manage visitor check-ins and check-outs efficiently. The system aims to streamline the registration process with digital forms, secure data handling, real-time updates on visitor status, and automated operations to ensure efficiency and reduce errors. The platform also integrates with existing security systems to enhance overall premises safety and provide comprehensive analytics for visitor data management.

TECHNOLOGY STACK:

HTML	-	HTML5
CSS	-	CSS3
BOOTSTRAP	-	Bootstrap 5.3. x
NODE JS	-	v20.12.2
MONGO DB	-	7.0.8
ANGULAR	-	17.3.6

GitHub Link : https://github.com/GiridharanS1729/visitor-mgt-system

Description:

The Visitor Management System is a web-based application designed to provide a convenient and secure platform for managing visitor check-ins and check-outs. It enables organizations to streamline the visitor registration process, ensuring a smooth and efficient experience for both visitors and administrators. The system integrates secure ID verification methods and supports multiple authentication options to ensure data integrity and safety.

Key features include visitor pre-registration, real-time updates on visitor status, and digital badge printing. The system offers detailed visitor information, including visit history and purpose of visit. Administrators can access comprehensive visitor logs for efficient data management.

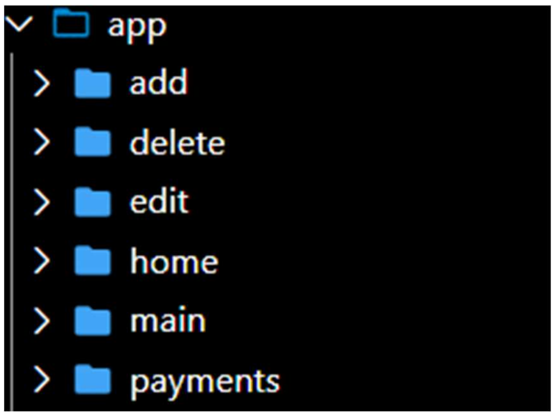
The Visitor Management System aims to improve operational efficiency by automating the visitor registration process, reducing manual errors, and providing valuable data insights for better decision-making. By offering a seamless and professional visitor experience, the system strives to enhance security and compliance while increasing overall satisfaction for both visitors and staff.

Angular:

Angular is used to build the entire application. With several components and services, it is very easy to integrate all the requirements.

Components used:

- **Home:**
  Displays all visitors and provides navigation within the Visitor Management System.
- **Add:**
  Allows administrators to add new visitors to the system.
- **Delete:**
  Handles the removal of visitor records from the system.
- **Edit:**
  Enables administrators to modify visitor information.
- **Payment:**
  Manages financial transactions related to visitor services.
- **Login:**
  Provides authentication for users to access the Visitor Management System.
- **Register:**
  Allows new users to create accounts and register for access to the system.



```
app
  add
  delete
  edit
  home
  main
  payments
```

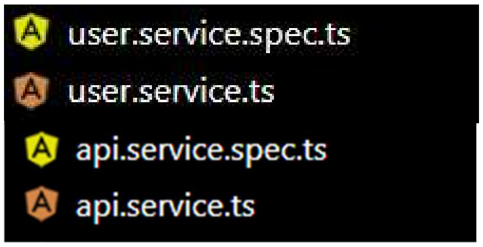
Services:

- Api service

  To retrieve and filter JSON.

- User service

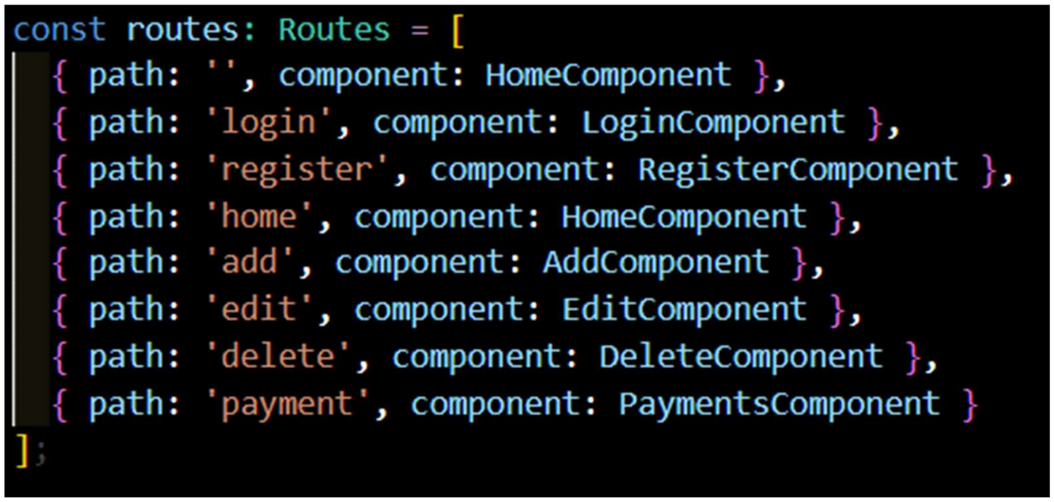
  To use the username of the currently logged in user across the app.



user.service.spec.ts
user.service.ts
api.service.spec.ts
api.service.ts

Routing:

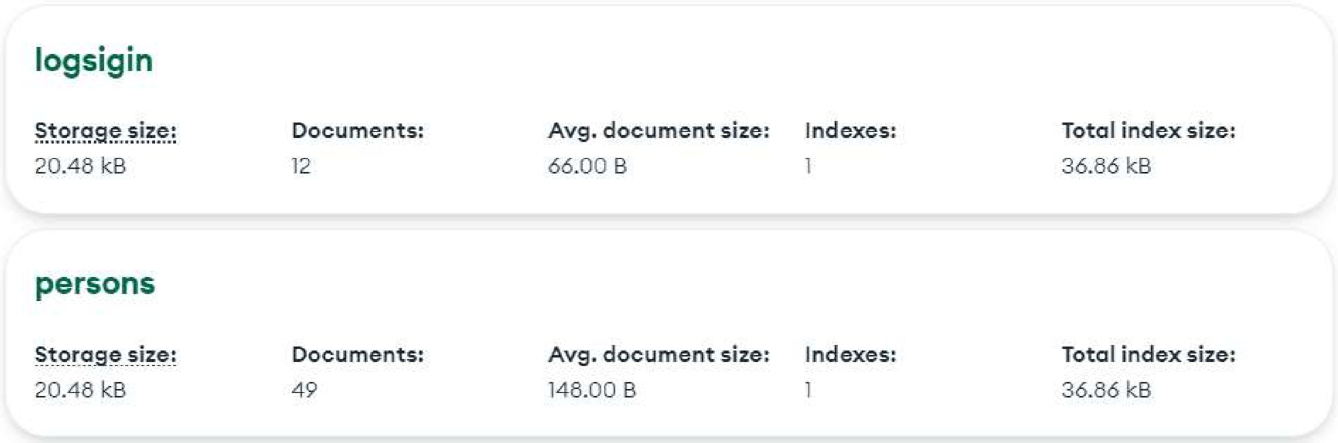
Several routing has been done in order to ensure smooth transition of pages.



```
const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'login', component: LoginComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'home', component: HomeComponent },
  { path: 'add', component: AddComponent },
  { path: 'edit', component: EditComponent },
  { path: 'delete', component: DeleteComponent },
  { path: 'payment', component: PaymentsComponent }
];
```

Mongo DB:

Three collection are present under Visitor Database .



logsigin

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	12	66.00 B	1	36.86 kB

persons

Storage size:	Documents:	Avg. document size:	Indexes:	Total index size:
20.48 kB	49	148.00 B	1	36.86 kB

- Users is used for storing the user registered details and later used for Login verification.

- To facilitate comprehensive visitor management, encompassing addition, deletion, editing, and viewing functionalities.

Coding:

App-routing-module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { EditComponent } from './edit/edit.component';
import { AddComponent } from './add/add.component';
import { DeleteComponent } from './delete/delete.component';
import { PaymentsComponent } from './payments/payments.component';
import { MainComponent } from './main/main.component';
const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'main', component: MainComponent },
  { path: 'home', component: HomeComponent },
  { path: 'add', component: AddComponent },
  { path: 'edit', component: EditComponent },
  { path: 'delete', component: DeleteComponent },
  { path: 'payment', component: PaymentsComponent }
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

App.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule, provideClientHydration } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { AddComponent } from './add/add.component';
import { DeleteComponent } from './delete/delete.component';
import { EditComponent } from './edit/edit.component';
import { MainComponent } from './main/main.component';
import { PaymentsComponent } from './payments/payments.component';
import { GooglePayButtonModule } from '@google-pay/button-angular';
import { LoginComponent } from './login/login.component';
import { RegisterComponent } from './register/register.component';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    AddComponent,
    DeleteComponent,
    EditComponent,
    MainComponent,
    PaymentsComponent,
      LoginComponent,
      RegisterComponent
   ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    GooglePayButtonModule
  ],
  providers: [
    provideClientHydration()
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.spec.ts

```
import { TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';
describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [ ],
      declarations: [AppComponent],
    }).compileComponents();
  });
  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });
  it(`should have as title 'visitor_management_system'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('visitor_management_system');
  });
  it('should render title', () => {

    const fixture = TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    const compiled = fixture.nativeElement as HTMLElement;

    expect(compiled.querySelector('h1')?.textContent).toContain('Hello,
visitor_management_system');

  });

});
```

app.component.html

```
<router-outlet>

  <div class="navbar">

    <a routerLinkActive="active-link" routerLink="home" class="nav-link">All
Visitors</a>

    <a routerLinkActive="active-link" routerLink="add" class="nav-link">Add new
Visitor</a>

    <a routerLinkActive="active-link" routerLink="edit" class="nav-link">Update
Visitor</a>

    <a routerLinkActive="active-link" routerLink="delete" class="nav-link">Delete
Visitor</a>

    <a routerLinkActive="active-link" routerLink="payment"  class="nav-
link">Payment</a>

    <a routerLinkActive="active-link" routerLink="main"  class="nav-
link">Login/SignUp</a>

  </div>  <router-outlet />
```

app.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrl: './app.component.css'
})
export class AppComponent {
  title = 'visitor_management';
}
```

user.service.ts:

```
import { Injectable } from '@angular/core';
@Injectable({ providedIn: 'root'
})
export class UserService { private username: string = '';
setUsername(username: string) { this.username = username;
localStorage.setItem('username', username);
}
getUsername(): string {
return this.username || localStorage.getItem('username') || '';
}
clearUsername() { this.username = '';
localStorage.removeItem('username');
}}
```

Services:

api.service.ts:

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
@Injectable({
  providedIn: 'root'
})
export class ApiService {
  private jsonUrl = '../assets/data.json';
  constructor(private http: HttpClient) { }
  getData(searchQuery: string = '', page: number = 1, perPage: number = 6):
Observable<any> {
    return this.http.get<any>(this.jsonUrl).pipe(
      map(data => {
        let filteredData = data.users;
        if (searchQuery) {
          const regex = new RegExp(`^${searchQuery}`, 'i');
          filteredData = filteredData.filter((user: { username: string; }) =>
regex.test(user.username));
        }
        const totalRecords = filteredData.length;
        const totalPages = Math.ceil(totalRecords / perPage);
        const startIndex = (page - 1) * perPage;
        const endIndex = Math.min(startIndex + perPage, totalRecords);
        return {
          users: filteredData.slice(startIndex, endIndex),totalPages
        };
      }));
}}
```

Components:

add.component.html

```
<div class="container" id="add">

   <form id="addForm" action="http://localhost:4201/add" method="POST">

      <div class="one">

         <label for="id">Room Number</label>

         <input type="number" id="idc" name="idc" placeholder="Enter Room Number"
max="500" required>

      </div>

      <br><br>

      <div class="one">

         <label for="name">Name</label>

         <input type="text" id="username" name="username" placeholder="Enter Name"
required>

      </div>

      <br><br>

      <div class="one">

         <label for="contact">Contact Number</label>

         <input type="tel" id="phone" name="phone" pattern="\d{10}" required
placeholder="1234567890">

      </div>

      <br><br>
```

```
        <div class="one">

            <label for="aadhar">Aadhar Number</label>

            <input type="text" id="aadhar" name="aadhar" minlength="14" maxlength="14"
pattern="\d{4} \d{4} \d{4}" placeholder="XXXX XXXX XXXX" required>

        </div>

        <br><br>

        <div class="one">

            <label for="intime">In-Time</label>

            <input type="datetime-local" id="intime" name="intime" required>

        </div>

        <br><br>

        <div class="one">

            <label for="outtime">Out-Time</label>

            <input type="datetime-local" id="outtime" name="outtime" required>

        </div>

        <br><br><br><br>

        <button type="submit">Add</button>

    </form>

</div>
```

add.component.ts

```
   import { Component } from '@angular/core';

   @Component({

     selector: 'app-add',

     templateUrl: './add.component.html',

     styleUrl: './add.component.css'

   })

   export class AddComponent {}
```

main.component.html

```
<div class="full">

  <div class="container out">

    <div class="container-in gradient-background">

      <h1 class="tit primary-text">Login / Signup</h1>

      <form id="authForm" action="#" method="POST">

        <label for="username" class="primary-text">Username</label><br />

        <input type="text" value="a" id="username" name="username" class="input-
field"><br><br>

        <label for="password" class="primary-text">Password</label><br />

        <input type="password" value="a" id="password" name="password" class="input-
field"><br><br>

        <div class="btns">
```

```
          <button type="button" onclick="submitForm('signup')"
class="button">Signup</button>

          <button type="button" onclick="submitForm('login')"
class="button">Login</button>

        </div>

      </form>

    </div>

  </div>
</div>
```

main.component.ts

```
  import { Component } from '@angular/core';
  import { Router } from '@angular/router';

  @Component({
    selector: 'app-main',
    templateUrl: './main.component.html',
    styleUrls: ['./main.component.css']
  })
  export class MainComponent {
    constructor(private router: Router) { }
    submitForm(action: string): void {
      const form = document.getElementById('authForm') as HTMLFormElement;
      const username = (document.getElementById('username') as
  HTMLInputElement).value;
      const password = (document.getElementById('password') as
  HTMLInputElement).value;

      if (action === 'login') {
        form.action = 'http://localhost:4201/login';
        this.router.navigate(['/home']);
      }
```

```
else if (action === 'signup') {
      form.action = 'http://localhost:4201/signup';
      form.submit();
    }
  }
}
```

home.component.html

```
<div class="container" id="view">

    <iframe src="http://localhost:4201/view" frameborder="0" id="viewtable"></iframe>

</div>
```

home.components.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-home',
  templateUrl: './home.component.html',
  styleUrl: './home.component.css'
})
export class HomeComponent {}
```

delete.component.html

```
<div class="container" id="delete">

    <form id="deleteForm" action="http://localhost:4201/delete" method="POST">

        <div class="one">

            <label for="idc">Room Number</label>

            <input type="number" id="idc" name="idc" placeholder="Enter Room Number"
max="500" required>

        </div>

        <br><br>
        <button type="submit">Delete</button>

    </form>

</div>
```

delete.component.ts

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-delete',
  templateUrl: './delete.component.html',
  styleUrl: './delete.component.css'
})
export class DeleteComponent {}
```

edit.component.html

```
<div class="container" id="update">

    <form id="updateForm" action="http://localhost:4201/update" method="POST">

        <div class="one">

            <label for="id">Room Number</label>

            <input type="number" id="idc" name="idc" placeholder="Enter Room Number"
max="500">

        </div>

        <br><br>

        <input type="hidden" id="update_id" name="id">

        <div class="one">

            <label for="update_name">Name</label>

            <input type="text" id="update_name" name="username" placeholder="Enter
Name" >
```

```
        </div>

        <br><br>

        <div class="one">

            <label for="update_contact">Contact Number</label>

            <input type="tel" id="update_contact" name="phone" pattern="\d{10}"
placeholder="1234567890">

        </div>

        <br><br>

        <div class="one">

            <label for="update_aadhar">Aadhar Number</label>

            <input type="text" id="update_aadhar" name="aadhar" minlength="14"
maxlength="14" pattern="\d{4} \d{4} \d{4}" placeholder="XXXX XXXX XXXX" >

        </div>

        <br><br>

        <div class="one">

            <label for="update_intime">In-Time</label>

            <input type="datetime-local" id="update_intime" name="intime" >

        </div>

        <br><br>

        <div class="one">

            <label for="update_outtime">Out-Time</label>

            <input type="datetime-local" id="update_outtime" name="outtime" >
```

```
          </div>

      <button type="submit">Update</button>

        </form>
    </div>
```

edit.component.ts

```
    import { Component } from '@angular/core';

    @Component({

      selector: 'app-edit',

      templateUrl: './edit.component.html',

      styleUrl: './edit.component.css'

    })

    export class EditComponent {}
```

payments.component.html

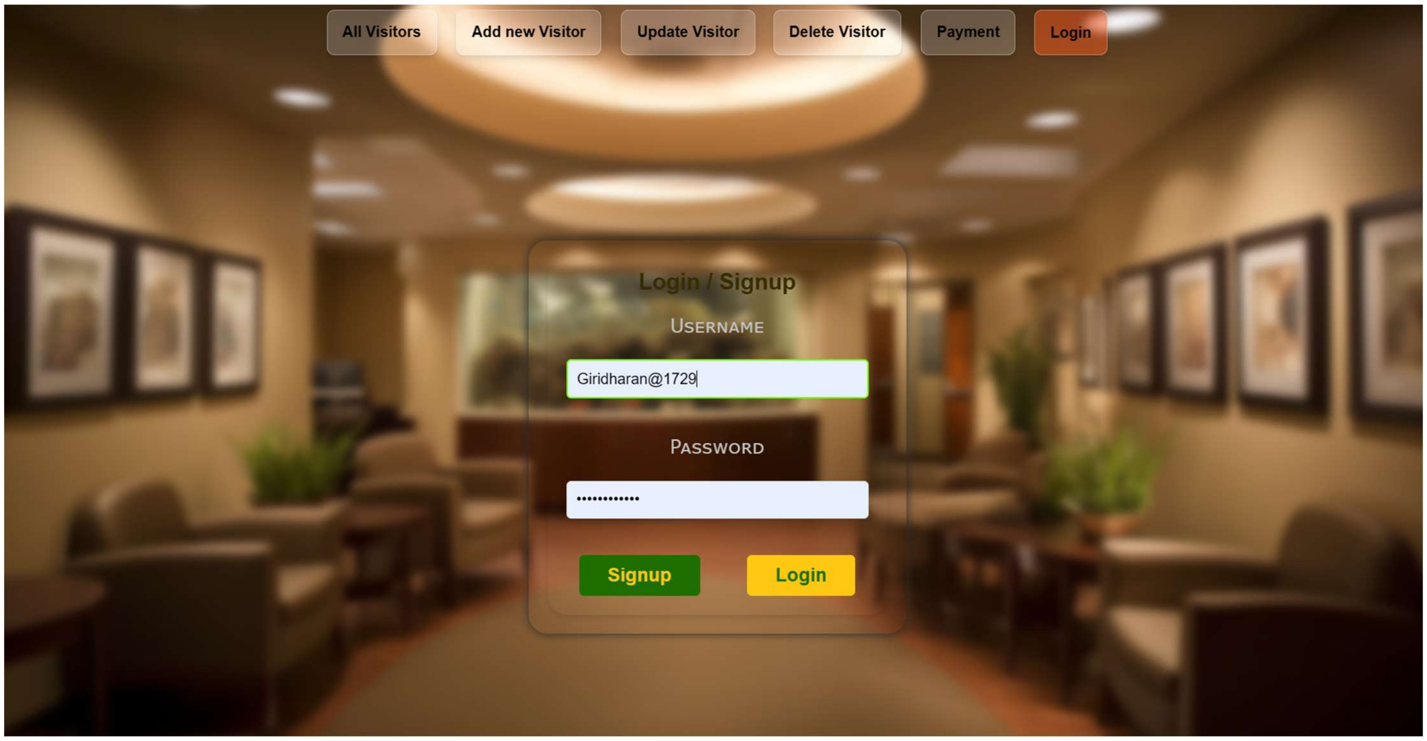
```
<div class="wrapper">
    <google-pay-button
      environment="TEST"
      buttonType="buy"
      buttonColor="black"
      [paymentRequest]="paymentRequest"
      (loadpaymentdata)="onLoadPaymentData($event)">
    </google-pay-button>
  </div>
```

payments.component.ts

```
import { Component } from '@angular/core';

import { GooglePayButtonModule } from '@google-pay/button-angular';

@Component({

  selector: 'app-payments',

  templateUrl: './payments.component.html',

  styleUrls: ['./payments.component.css']

})

export class PaymentsComponent {

  paymentRequest: google.payments.api.PaymentDataRequest = {

    apiVersion: 2,

    apiVersionMinor: 0,

    allowedPaymentMethods: [

      {

        type: 'CARD',

        parameters: {

          allowedAuthMethods: ["PAN_ONLY", "CRYPTOGRAM_3DS"],

          allowedCardNetworks: ["AMEX", "VISA", "MASTERCARD"]

        },

        tokenizationSpecification: {

          type: 'PAYMENT_GATEWAY',
          parameters: {
            'gateway': 'gatewayName',
            'gatewayMerchantId': 'GatewayMerchantId'
          }}}],
    merchantInfo: {
      merchantId: "12345678901234567890",
      merchantName: "Merchant Name"
    },
    transactionInfo: {
      totalPriceStatus: "FINAL",
      totalPriceLabel: "Total",
      totalPrice: "10.00",
      currencyCode: "INR",
      countryCode: "IN"
    }
  }

  onLoadPaymentData(e: any) {
    console.log(e, ">> Data");
  }
}
```

Output:

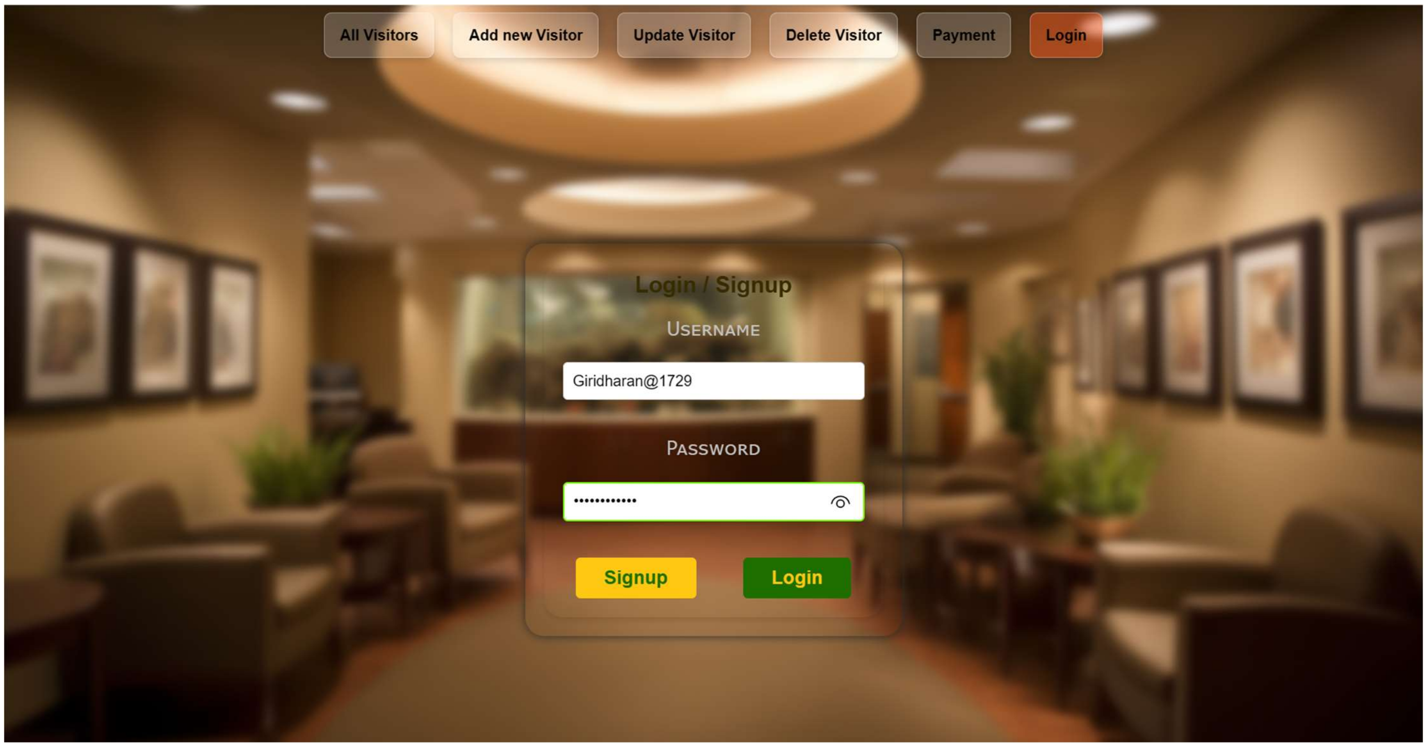
SignUp:



Giridharan@1729

••••••••••••

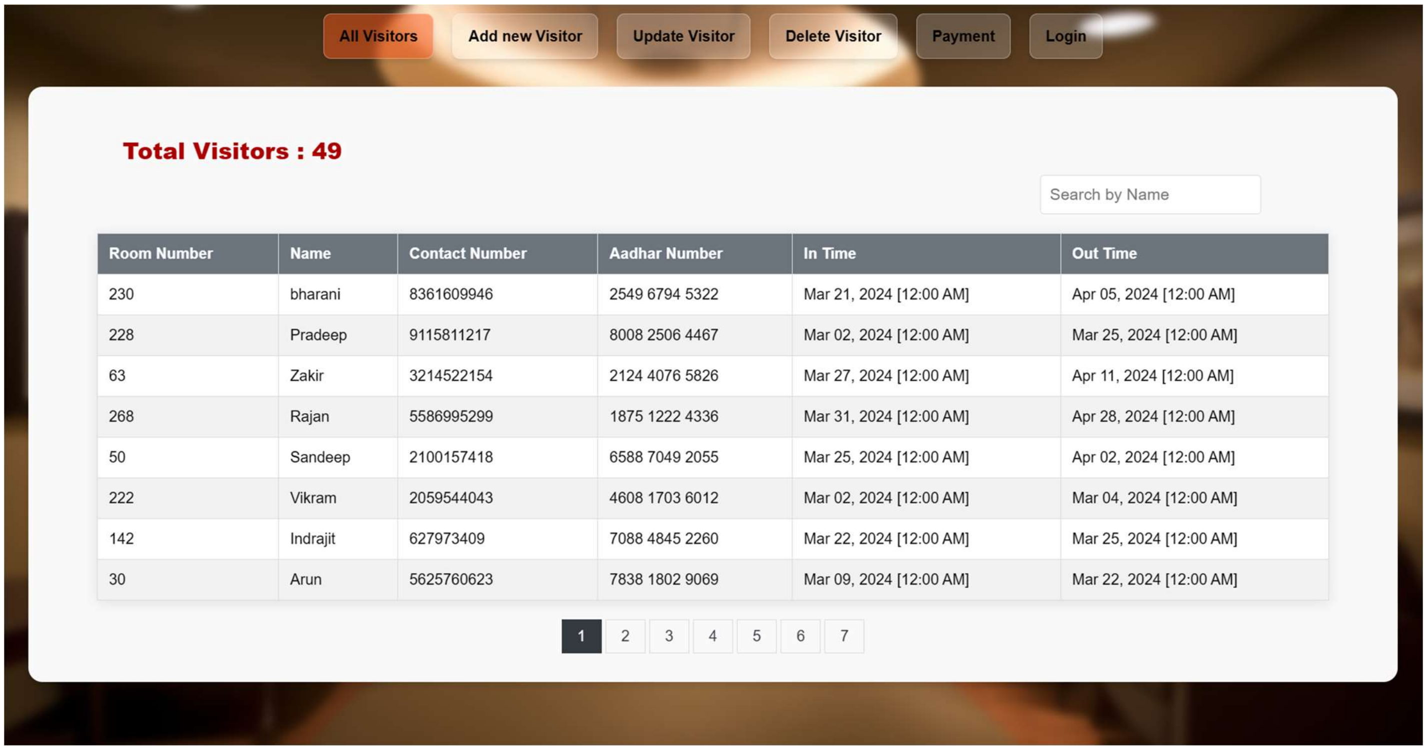
Login:



Giridharan@1729

••••••••••••

All Visitor:



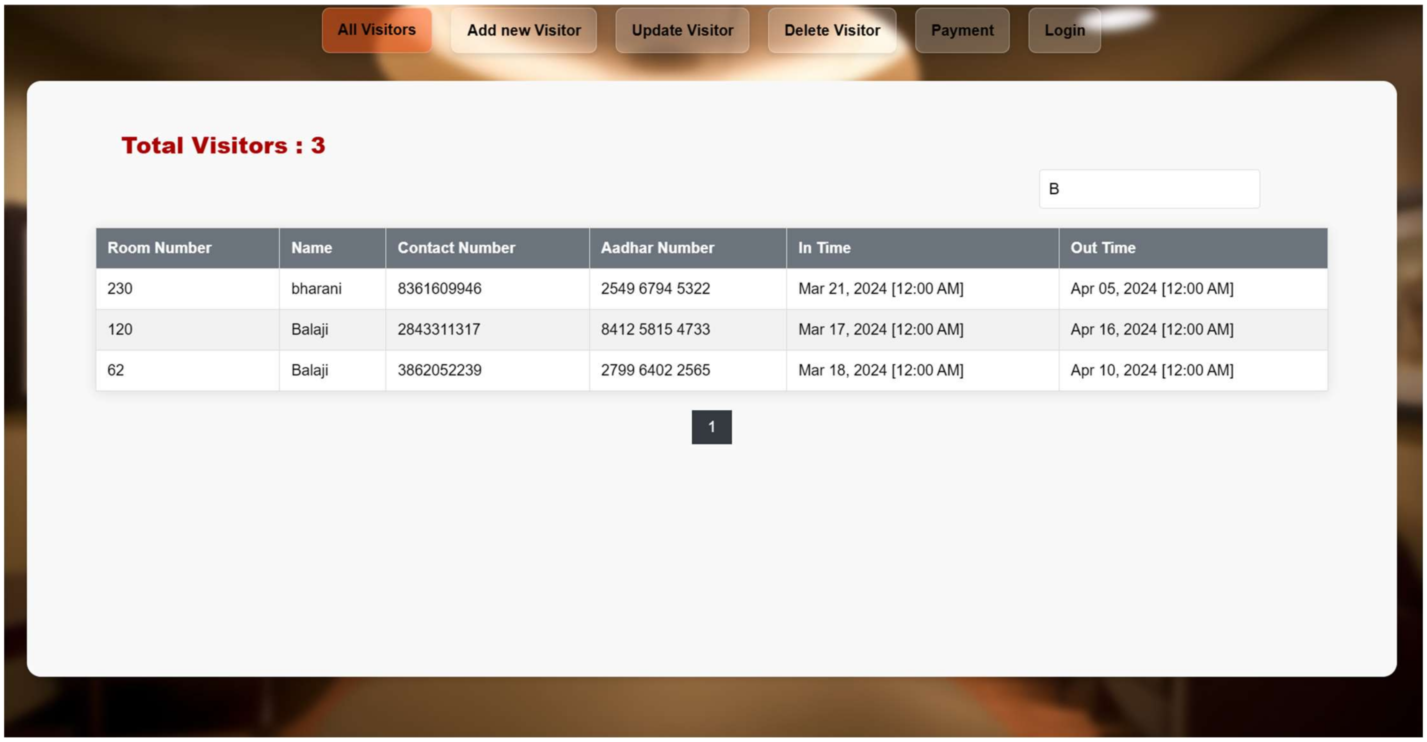
Total Visitors : 49

Search by Name

Room Number	Name	Contact Number	Aadhar Number	In Time	Out Time
230	bharani	8361609946	2549 6794 5322	Mar 21, 2024 [12:00 AM]	Apr 05, 2024 [12:00 AM]
228	Pradeep	9115811217	8008 2506 4467	Mar 02, 2024 [12:00 AM]	Mar 25, 2024 [12:00 AM]
63	Zakir	3214522154	2124 4076 5826	Mar 27, 2024 [12:00 AM]	Apr 11, 2024 [12:00 AM]
268	Rajan	5586995299	1875 1222 4336	Mar 31, 2024 [12:00 AM]	Apr 28, 2024 [12:00 AM]
50	Sandeep	2100157418	6588 7049 2055	Mar 25, 2024 [12:00 AM]	Apr 02, 2024 [12:00 AM]
222	Vikram	2059544043	4608 1703 6012	Mar 02, 2024 [12:00 AM]	Mar 04, 2024 [12:00 AM]
142	Indrajit	627973409	7088 4845 2260	Mar 22, 2024 [12:00 AM]	Mar 25, 2024 [12:00 AM]
30	Arun	5625760623	7838 1802 9069	Mar 09, 2024 [12:00 AM]	Mar 22, 2024 [12:00 AM]

1 2 3 4 5 6 7

Search Visitors:



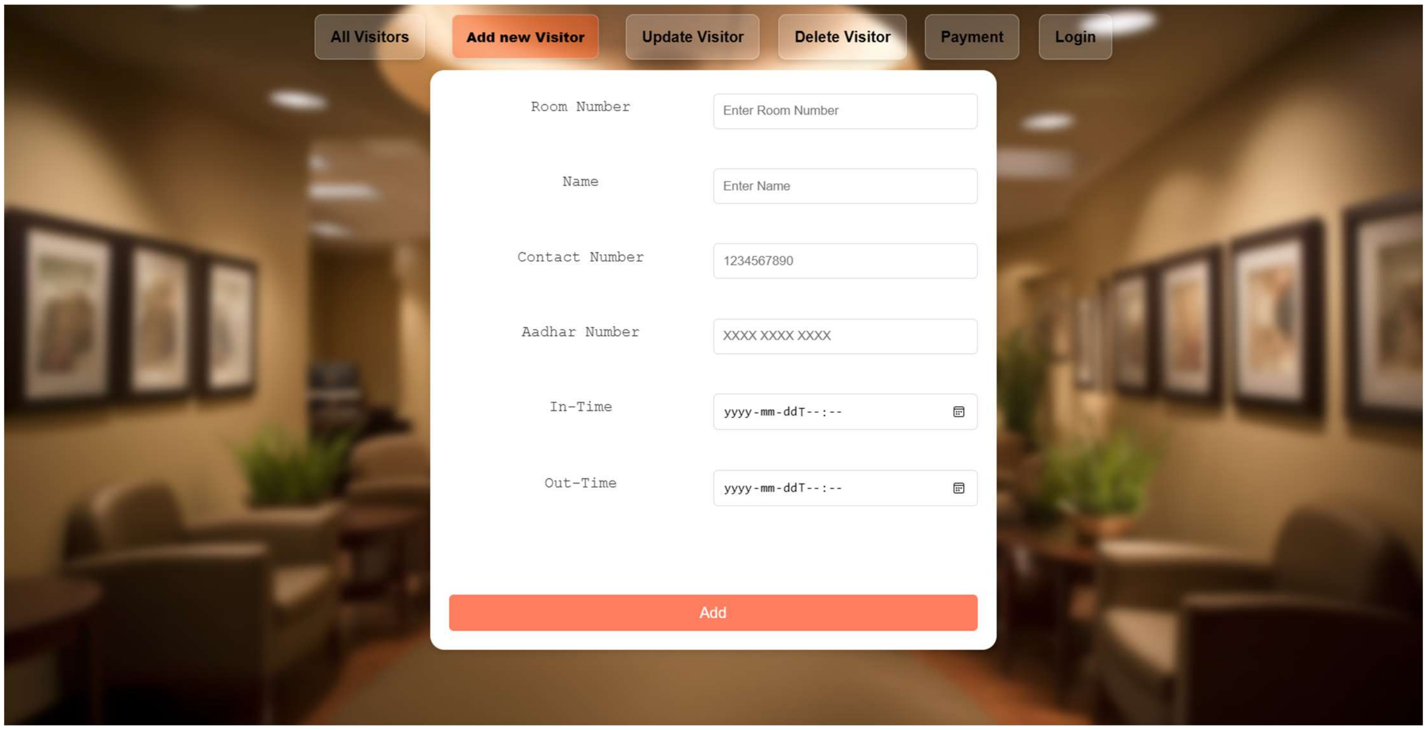
Total Visitors : 3

B

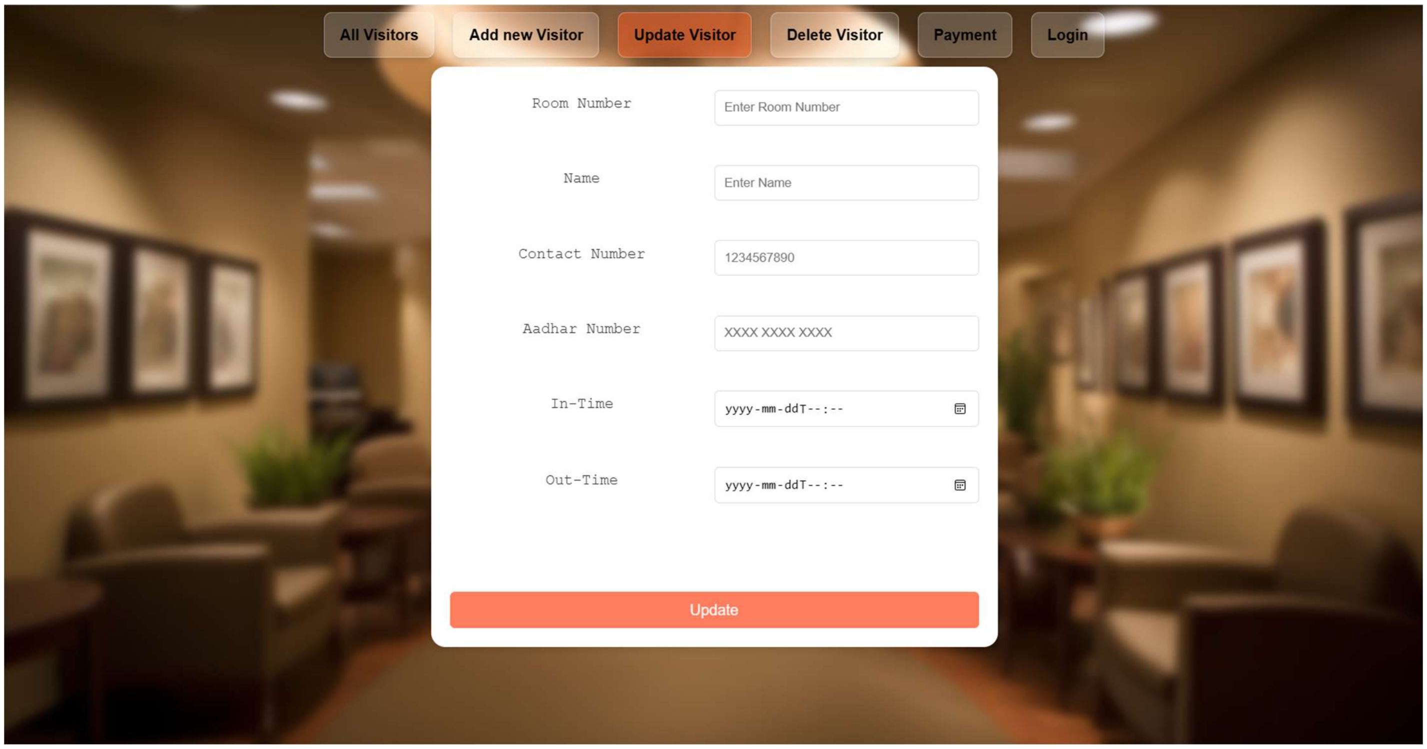
Room Number	Name	Contact Number	Aadhar Number	In Time	Out Time
230	bharani	8361609946	2549 6794 5322	Mar 21, 2024 [12:00 AM]	Apr 05, 2024 [12:00 AM]
120	Balaji	2843311317	8412 5815 4733	Mar 17, 2024 [12:00 AM]	Apr 16, 2024 [12:00 AM]
62	Balaji	3862052239	2799 6402 2565	Mar 18, 2024 [12:00 AM]	Apr 10, 2024 [12:00 AM]

1

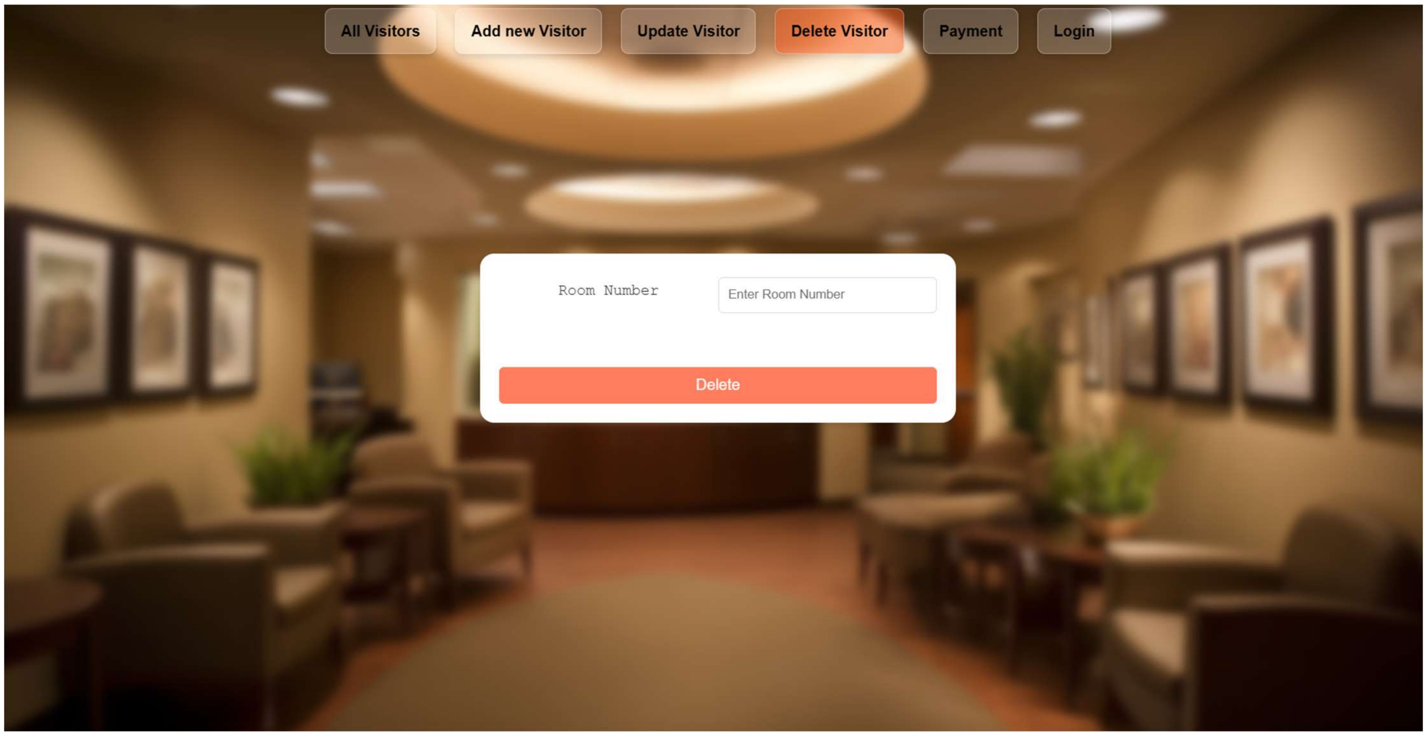
Add new Visitor:



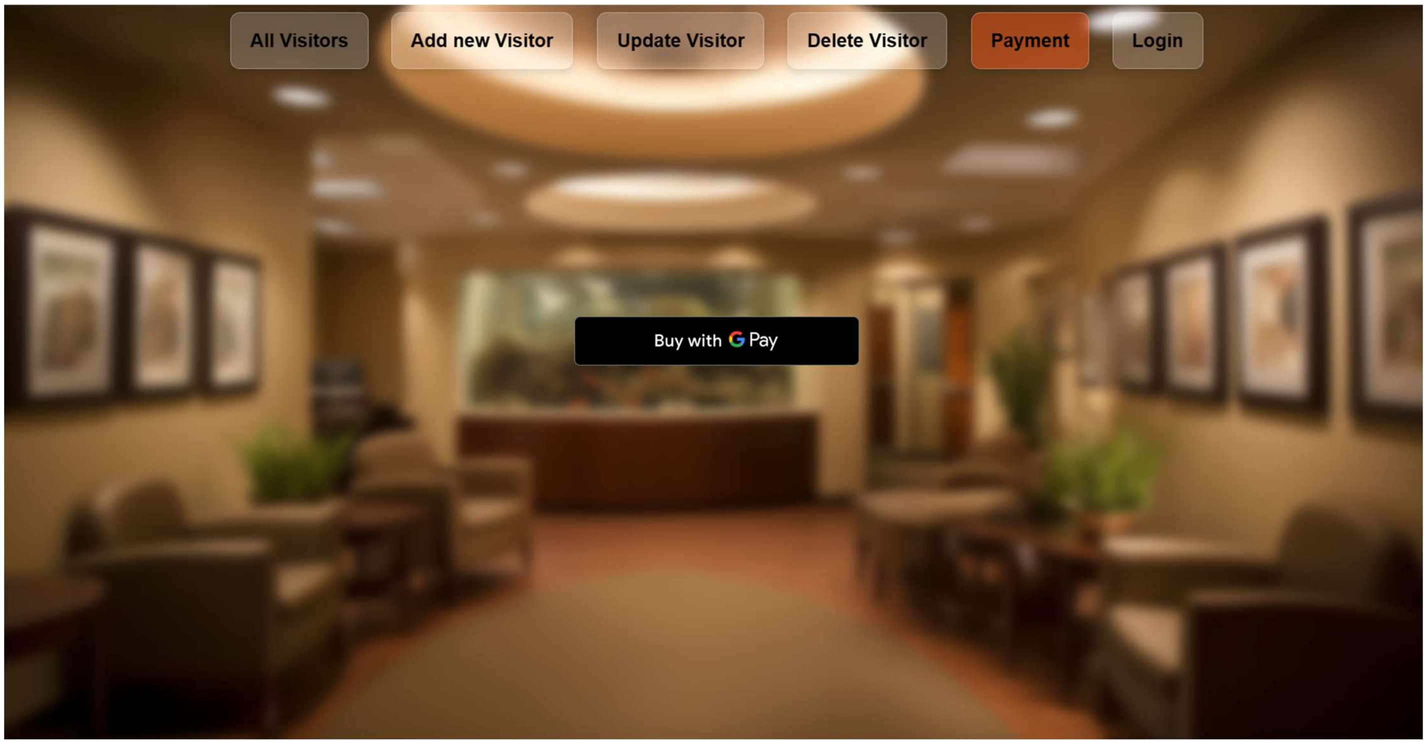
Update a Visitor:



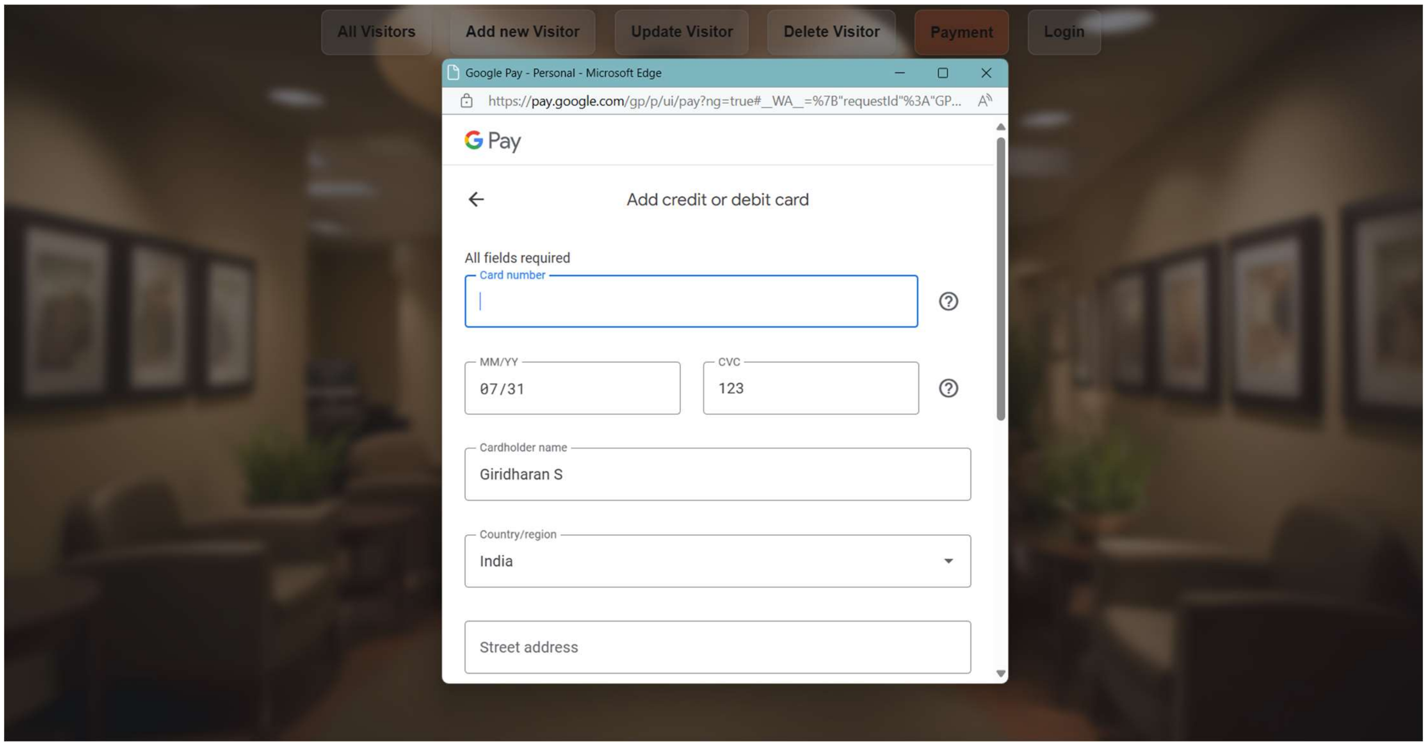
Delete Visitor by Room number:



Payments:



Payment by using Gpay:


All Visitors
Add new Visitor
Update Visitor
Delete Visitor
Payment
Login
Google Pay - Personal - Microsoft Edge
https://pay.google.com/gp/p/ui/pay?ng=true#_WA_=%7B"requestId"%3A"GP...
G Pay

Add credit or debit card

All fields required

Card number

MM/YY: 07/31

CVC: 123

Cardholder name: Giridharan S

Country/region: India

Street address

Conclusion:

A visitor management system built using Angular efficiently manages visitor information. It includes CRUD operations for records, Google Pay for secure payments, and login/signup for user authentication. This system enhances security, streamlines the check-in process, and ensures accurate record-keeping, significantly improving operational efficiency and visitor experience.