BHARATIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

Munshi nagar, Andheri (W) ,Mumbai - 400058

DEPARTMENT OF MASTER OF COMPUTER APPLICATION

CLASS: F.Y. MCA SEM: II

COURSE CODE: MC509 SUBJECT NAME: MOBILE PROGRAMMING LAB ROLL NO./UCID : 2023510032 BATCH: B

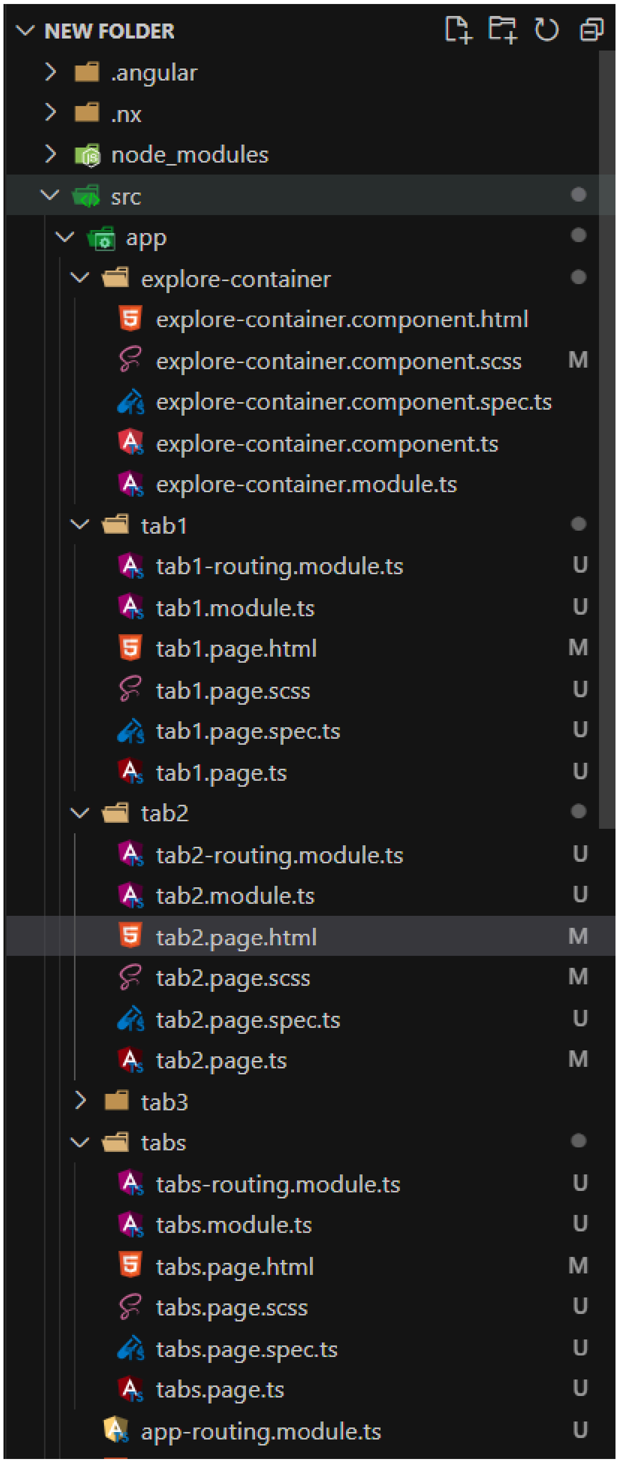
NAME: Durgesh Dilip Mandge

EXPERIMENT NO: 08

EXPERIMENT TITLE: Develop Camera and Calendar API integrated in one app.

Tools Required: Ionic, VSCode

Files which are used for the application (.html, .scss, .ts etc) : .js



```
NEW FOLDER
  > .angular
  > .nx
  > node_modules
  v src
    v app
      v explore-container
          explore-container.component.html
          explore-container.component.scss          M
          explore-container.component.spec.ts
          explore-container.component.ts
          explore-container.module.ts
      v tab1
          tab1-routing.module.ts                    U
          tab1.module.ts                            U
          tab1.page.html                            M
          tab1.page.scss                            U
          tab1.page.spec.ts                         U
          tab1.page.ts                              U
      v tab2
          tab2-routing.module.ts                    U
          tab2.module.ts                            U
          tab2.page.html                            M
          tab2.page.scss                            M
          tab2.page.spec.ts                         U
          tab2.page.ts                              M
      > tab3
      v tabs
          tabs-routing.module.ts                    U
          tabs.module.ts                            U
          tabs.page.html                            M
          tabs.page.scss                            U
          tabs.page.spec.ts                         U
          tabs.page.ts                              U
        app-routing.module.ts                       U
```

Code :

```
<ion-header>
```

```
  <ion-toolbar>
    <ion-title>Camera</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-button class="camera-icon" (click)="openCamera()"><ion-icon
name="camera" slot="start"></ion-icon></ion-button>
  <ion-button class="camera-icon" (click)="openGallery()"><ion-icon
name="images"></ion-icon></ion-button>
  <ion-button expand="block" (click)="captureImage()"
*ngIf="showCamera">Capture Image</ion-button>
  <video #video autoplay *ngIf="showCamera"></video>
  <canvas #canvas *ngIf="showCamera"></canvas>

  <div *ngIf="showGallery" class="gallery-container">
    <ion-card *ngFor="let image of capturedImages; let i = index"
class="gallery-card">
      <img [src]="image" (click)="deleteImage(i)" class="gallery-image">
      <ion-button (click)="deleteImage(i)" color="danger"
class="delete-button">Delete</ion-button>
    </ion-card>
  </div>

  <input type="file" accept="image/*" (change)="selectFromGallery($event)"
*ngIf="showGallery" class="gallery-input">
</ion-content>
<style>
  .camera-icon {
    font-size: 3rem; /* Adjust the size as needed */
    margin: 10% 35%; /* Center the icon horizontally */
  }
</style>
```

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

@Component({
  selector: 'app-tab1',
  templateUrl: 'tab1.page.html',
  styleUrls: ['tab1.page.scss']
```

```
})

export class Tab1Page {
  @ViewChild('video') videoElement!: ElementRef<HTMLVideoElement>;
  @ViewChild('canvas') canvasElement!: ElementRef<HTMLCanvasElement>;

  capturedImages: string[] = [];
  showCamera: boolean = false;
  showGallery: boolean = false;
  showButtonsAtTop: boolean = false;

  constructor() {}

  openCamera() {
    this.showCamera = true;
    this.showGallery = false;
    this.showButtonsAtTop = true;
    this.initializeCamera();
    this.capturedImages = [];
  }

  openGallery() {
    this.showGallery = true;
    this.showCamera = false;
    this.showButtonsAtTop = false;
  }

  initializeCamera() {
    navigator.mediaDevices.getUserMedia({ video: true })
      .then(stream => {
        this.videoElement.nativeElement.srcObject = stream;
      })
      .catch(err => console.error(err));
  }

  captureImage() {
    const canvas = this.canvasElement.nativeElement;
    canvas.width = this.videoElement.nativeElement.videoWidth;
    canvas.height = this.videoElement.nativeElement.videoHeight;
    const ctx = canvas.getContext('2d');
    if (ctx) {
      ctx.drawImage(this.videoElement.nativeElement, 0, 0);
    }
    const capturedImage = canvas.toDataURL('image/png');
    this.capturedImages.push(capturedImage);
  }

  selectFromGallery(event: Event) {
    const target = event.target as HTMLInputElement;
    if (target.files) {
      const files = Array.from(target.files);
      files.forEach(file => {
        const reader = new FileReader();
        reader.onload = () => {
          this.capturedImages.push(reader.result as string);
        }
        reader.readAsDataURL(file);
      });
    }
  }

  deleteImage(index: number) {
    this.capturedImages.splice(index, 1);
  }
}
```

```
<ion-header>
  <ion-toolbar>
    <ion-title>Calendar</ion-title>
    <ion-buttons slot="primary">
      <ion-button (click)="openEventModal()">
        <!-- <ion-icon name="target" slot="icon-only"></ion-icon> -->
        <ion-icon name="pricetags-outline"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content>
```

```
  <div class="year-month-navigation">
    <!-- Month navigation buttons -->
  </div>
  <div class="calendar-container">
    <div class="calendar-grid">
      <div class="calendar-header">
        <div *ngFor="let day of daysOfWeek" class="day-header">{{ day
}}</div>
      </div>
      <div *ngFor="let week of currentMonthData.weeks" class="week">
        <div *ngFor="let day of week" class="day"
(click)="selectDay(day)">
          <div class="day-number">{{ day.day }}</div>
          <div *ngIf="day.events.length > 0"
class="event-indicator"></div>
        </div>
      </div>
    </div>
  </div>
</ion-content>

<ion-footer *ngIf="showEventModal">
  <ion-toolbar>
    <ion-title>Add Event</ion-title>
    <ion-buttons slot="primary">
      <ion-button (click)="saveEvent()">
        <ion-icon name="checkmark" slot="icon-only"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>

  <ion-item>
    <ion-label position="stacked">Title</ion-label>
    <ion-input type="text" [(ngModel)]="event.title"></ion-input>
  </ion-item>

  <ion-item>
    <ion-label position="stacked">Description</ion-label>
    <ion-textarea [(ngModel)]="event.description"></ion-textarea>
  </ion-item>
  <ion-item>
    <ion-label position="stacked">Date</ion-label>
    <ion-datetime
      [(ngModel)]="event.date"
      displayFormat="MMM DD, YYYY"
    ></ion-datetime>
  </ion-item>
</ion-footer>
```

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

@Component({
  selector: 'app-tab2',
  templateUrl: 'tab2.page.html',
  styleUrls: ['tab2.page.scss'],
})
export class Tab2Page {
  daysOfWeek: string[] = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri',
'Sat'];
  months = [
    { name: 'January', index: 0 },
    { name: 'February', index: 1 },
    { name: 'March', index: 2 },
    { name: 'April', index: 3 },
    { name: 'May', index: 4 },
    { name: 'June', index: 5 },
    { name: 'July', index: 6 },
    { name: 'August', index: 7 },
    { name: 'September', index: 8 },
    { name: 'October', index: 9 },
    { name: 'November', index: 10 },
    { name: 'December', index: 11 }
  ];
  currentYear: number;
  currentMonth: number;
  currentMonthData: any;
  events: any[] = [];
  showEventModal = false;
  event = {
    title: '',
    description: '',
    date: null,
  };
  todayDate: string;
  maxDate: string;
  selectedDayDate: string;

  constructor(private modalController: ModalController) {
    const currentDate = new Date();
    this.currentYear = currentDate.getFullYear();
    this.currentMonth = currentDate.getMonth();
    this.generateMonthData(this.currentYear, this.currentMonth);
    this.todayDate = currentDate.toISOString();
    const endDate = new Date(currentDate.getFullYear(),
currentDate.getMonth() + 1, 0);
    this.maxDate = endDate.toISOString();
  }

  generateMonthData(year: number, month: number) {
    const startDate = new Date(year, month, 1);
    const endDate = new Date(year, month + 1, 0);
    const weeks = this.generateWeeks(startDate, endDate);
    this.currentMonthData = { year: year, month: month, weeks: weeks };
  }

  generateWeeks(startDate: Date, endDate: Date): any[] {
    const weeks = [];
    let currentWeek = [];
    let currentDate = new Date(startDate);
    while (currentDate <= endDate) {
      const dayEvents = this.events.filter(
        (event) =>
          event.date.getFullYear() === currentDate.getFullYear() &&
          event.date.getMonth() === currentDate.getMonth() &&
          event.date.getDate() === currentDate.getDate()
      );
      currentWeek.push({ day: currentDate.getDate(), events: dayEvents });
      currentDate.setDate(currentDate.getDate() + 1);
      if (currentDate.getDay() === 0 || currentDate > endDate) {
        weeks.push(currentWeek);
        currentWeek = [];
      }
    }
    return weeks;
  }

  previousMonth() {
    this.currentMonth--;
    if (this.currentMonth < 0) {
      this.currentMonth = 11;
      this.currentYear--;
    }
    this.generateMonthData(this.currentYear, this.currentMonth);
  }

  nextMonth() {
    this.currentMonth++;
    if (this.currentMonth > 11) {
      this.currentMonth = 0;
      this.currentYear++;
    }
    this.generateMonthData(this.currentYear, this.currentMonth);
  }

  changeMonth() {
    this.generateMonthData(this.currentYear, this.currentMonth);
  }

  previousYear() {
    this.currentYear--;
    this.generateMonthData(this.currentYear, this.currentMonth);
  }

  nextYear() {
    this.currentYear++;
    this.generateMonthData(this.currentYear, this.currentMonth);
  }
  selectDay(day: any) {
    this.selectedDayDate = new Date(this.currentYear, this.currentMonth, day.day).toISOString();
    const selectedDateEvents = day.events.map((event: { title: any; description: any; }) => `${event.title}: ${event.description}`).join('\n');
    alert(selectedDateEvents);
    // Alternatively, you can display event details in a modal or another UI element
  }

  async openEventModal() {
    this.showEventModal = true;
    this.event = {
      title: '',
      description: '',
      date: null,
    };
  }

  async saveEvent() {
    if (this.event.date) {
      this.events.push({
        title: this.event.title,
        description: this.event.description,
        date: new Date(this.event.date)
      });
      this.generateMonthData(this.currentYear, this.currentMonth); // Update calendar UI
    }
    this.showEventModal = false;
  }
}
```

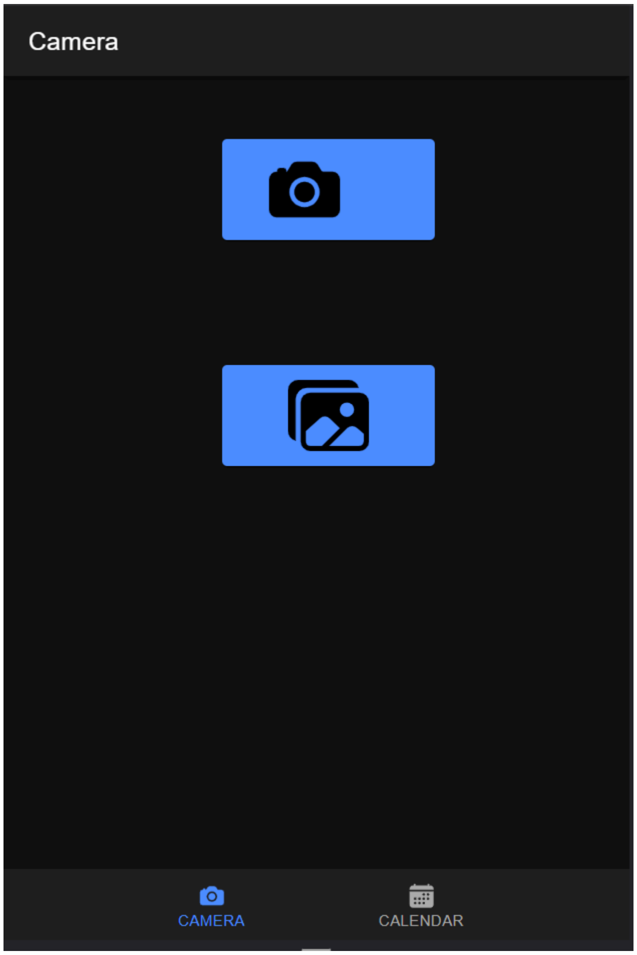
```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { TabsPage } from './tabs.page';
```

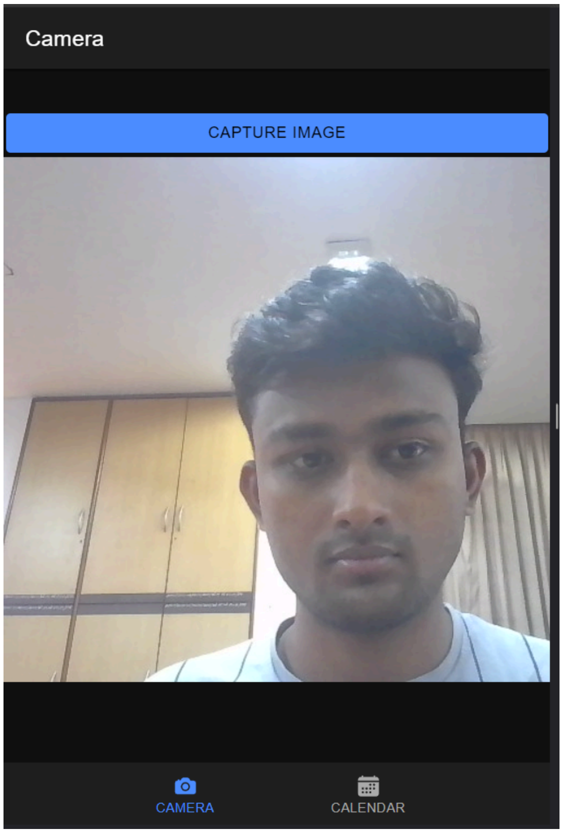
```
const routes: Routes = [
  {
    path: 'tabs',
    component: TabsPage,
    children: [
      {
        path: 'tab1',
        loadChildren: () => import('../tab1/tab1.module').then(m => m.Tab1PageModule)
      },
      {
        path: 'tab2',
        loadChildren: () => import('../tab2/tab2.module').then(m => m.Tab2PageModule)
      },

      {
        path: '',
        redirectTo: '/tabs/tab1',
        pathMatch: 'full'
      }
    ]
  },
  {
    path: '',
    redirectTo: '/tabs/tab1',
    pathMatch: 'full'
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
})
export class TabsPageRoutingModule {}
```

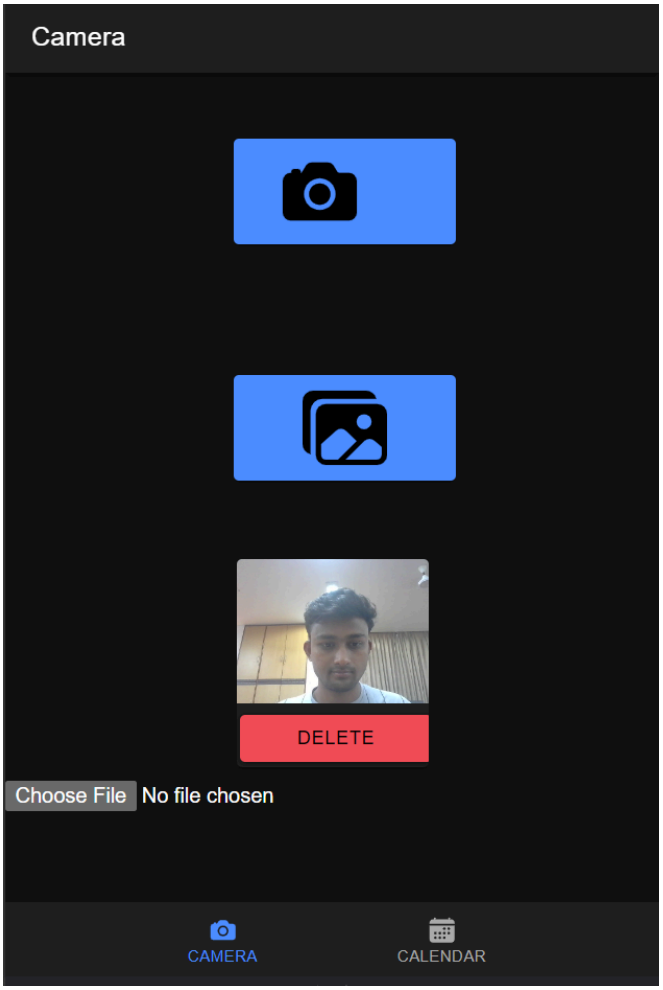
```
<ion-tabs>

  <ion-tab-bar slot="bottom">
    <ion-tab-button tab="tab1" href="/tabs/tab1">
```

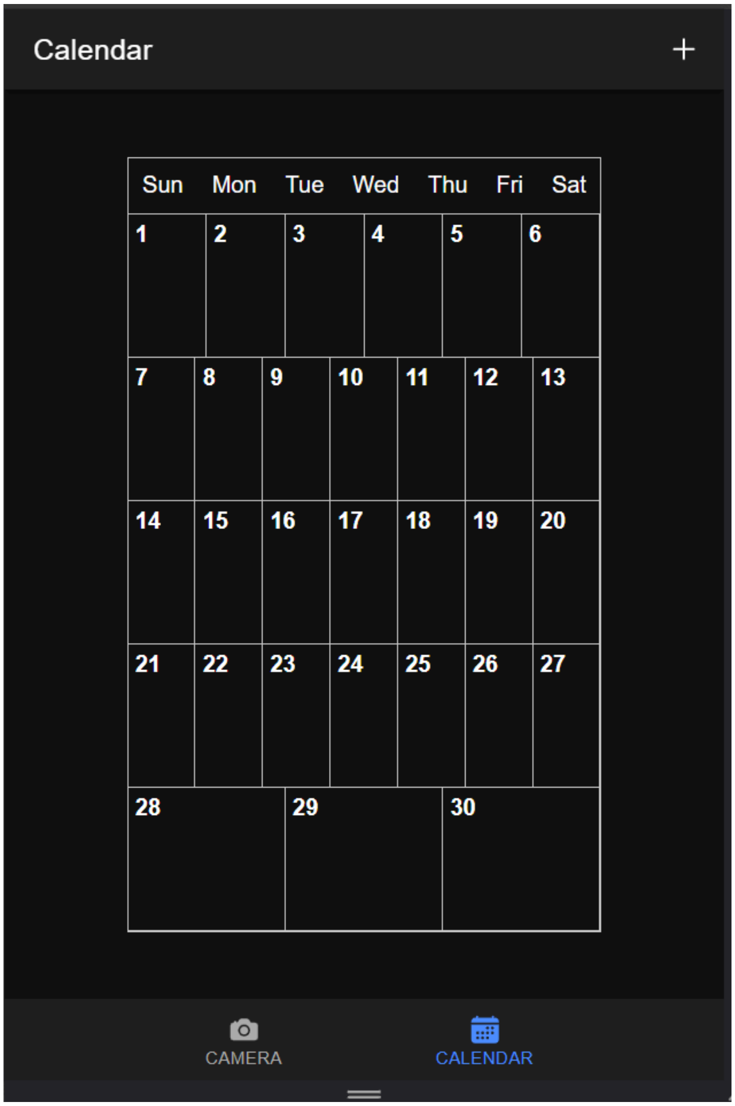
```
      <ion-icon aria-hidden="true" name="camera"></ion-icon>
      <ion-label>CAMERA</ion-label>
    </ion-tab-button>

    <ion-tab-button tab="tab2" href="/tabs/tab2">
      <ion-icon aria-hidden="true" name="calendar"></ion-icon>
      <ion-label>CALENDAR</ion-label>
    </ion-tab-button>

  </ion-tab-bar>

</ion-tabs>
```

OUTPUT:



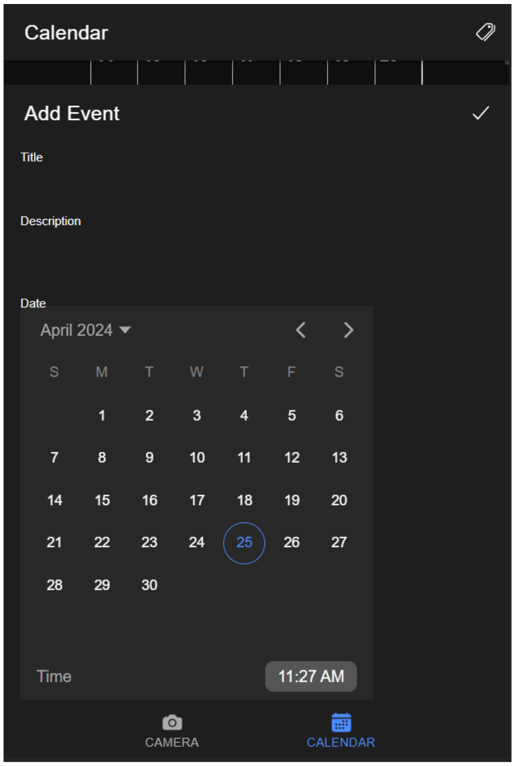



Camera
DELETE
Choose File
No file chosen
CAMERA
CALENDAR



Calendar

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				



Add Event

Title

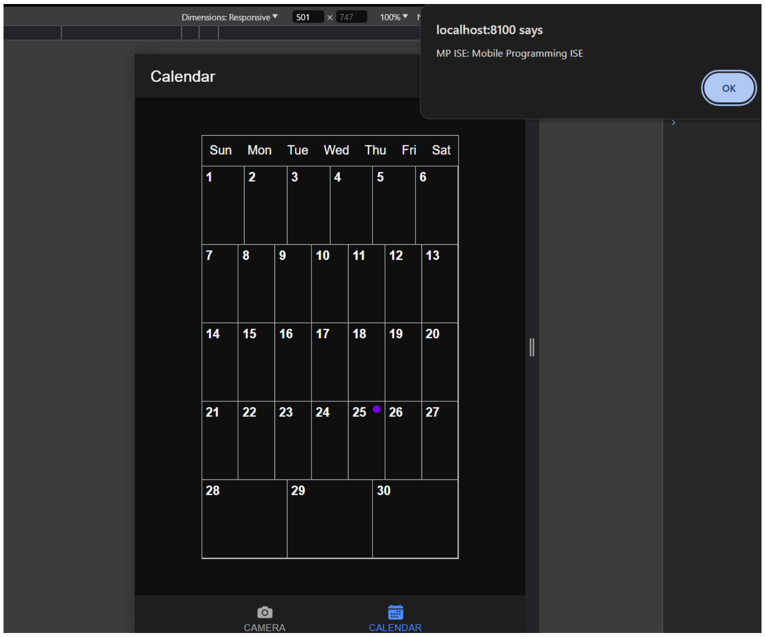
Description

Date

April 2024

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Time 11:27 AM



localhost:8100 says

MP ISE: Mobile Programming ISE

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25 ●	26	27
28	29	30				

Questions n Answers :

a) Which camera library do we import for the camera API?

- >

In an Ionic application, you typically use the @ionic-native/camera plugin for accessing device camera functionality. You would import it like this:

b) What is the use of camera.getPicture()?

->
camera.getPicture() is a method provided by the Ionic Native Camera plugin. It allows you to capture a photo from the device's camera or retrieve one from the device's image gallery, depending on the options you provide. It returns a promise that resolves with the image data.

c) What parameter do we set when we use CameraOptions?
When using CameraOptions, you can set various parameters to customize the behavior of the camera. Some common parameters include destinationType, sourceType, quality, allowEdit, encodingType, etc.
These parameters control aspects like the format of the image returned, the source of the image (camera or gallery), image quality, whether to allow editing before selection, and the encoding type.

d) What is base64 image? What is the use of it?
Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string format. A base64 image is an image represented as a base64 encoded string. It's commonly used in web development and mobile apps for various purposes such as embedding images directly into HTML/CSS, transmitting images over protocols that don't support binary data, and storing images in databases.

e) In app.module.ts, where do we initialize the camera (in which ngModel)?
->
In app.module.ts, you typically import the Camera module and include it in the providers array. You don't initialize the camera directly in app.module.ts, but you make it available for dependency injection throughout your app.

f)Which service do we require for saving images in the gallery?
- >
To save images in the device's gallery, you typically don't need a specific service. The @ionic-native/camera plugin itself provides methods to capture photos and save them to the device's gallery when configured appropriately.

g) What is the use of importing an Ionic Storage module?
->
The Ionic Storage module is used for storing data persistently in Ionic applications. It provides a simple key-value store interface that allows you to store and retrieve data asynchronously. It's commonly used for caching data, storing user preferences, and managing app state

h) Which calendar library do we import for working with calendar in an app?
->

For working with calendars in an Ionic app, you can use various plugins available. One popular choice is the @ionic-native/calendar plugin, which provides access to device calendar features. You would import it like this:

```
import { Calendar } from '@ionic-native/calendar/ngx';
```