

Tops Technology

Module 16)

Python DB and Framework

Presented By : Nandni Vala

Google Maps API

1. Integrating Google Maps API into Django projects.

- Google Maps API allows you to embed interactive maps, geolocation, and location-based features into your Django application.
 - **Obtain a Google Maps API Key**
 - Go to the Google Cloud Console.
 - Create a new project or select an existing one.
 - Navigate to **APIs & Services > Credentials**.
 - Click **Create Credentials > API Key**.
 - Copy the generated API key.
 - Enable the required APIs under **APIs & Services > Library**:
 - **Maps JavaScript API** (for embedding maps)
 - **Geocoding API** (for address to coordinates conversion, optional)
 - **Places API** (for places search, optional)
-



➤ **Install Required Dependencies**

- Install django-dotenv to manage environment variables (recommended for securing the API key).

- pip install django-dotenv

➤ **Store the API Key Securely**

- Create a .env file in your project root directory.

- `GOOGLE_MAPS_API_KEY=your_api_key_here`

- Load the environment variables in your settings.py:import os

- `from dotenv import load_dotenv`

- `load_dotenv()`

- `GOOGLE_MAPS_API_KEY = os.getenv('GOOGLE_MAPS_API_KEY')`

➤ **Add Google Maps to Your HTML Template**

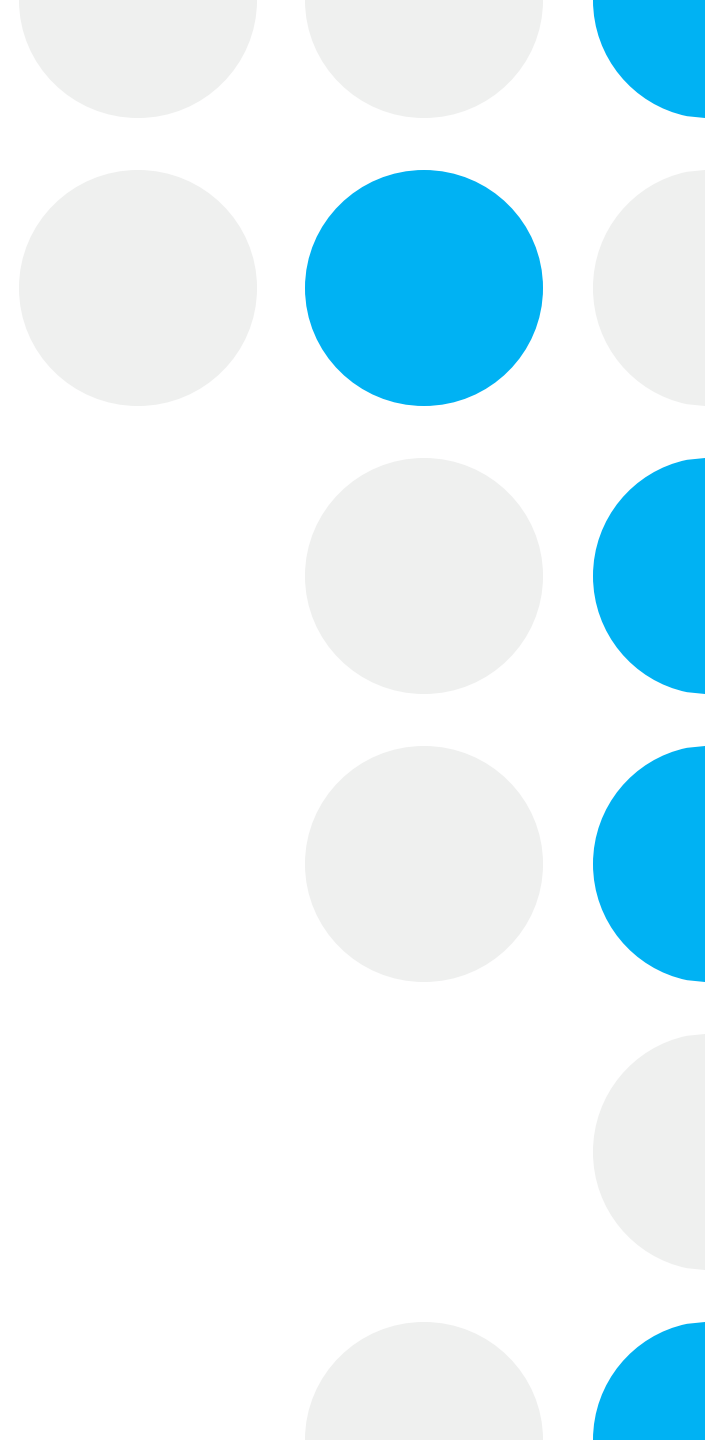
- In the template where you want to display the map:



- <!DOCTYPE html>
 - <html>
 - <head>
 - <title>Google Maps Integration</title>
 - <script src="https://maps.googleapis.com/maps/api/js?key={{ GOOGLE_MAPS_API_KEY }}"></script>
 - <style>
 - #map {
 - height: 400px;
 - width: 100%;
 - }
 - </style>
 - </head>
 - <body>
 - <h1>Google Maps Integration</h1>
 - <div id="map"></div>
-

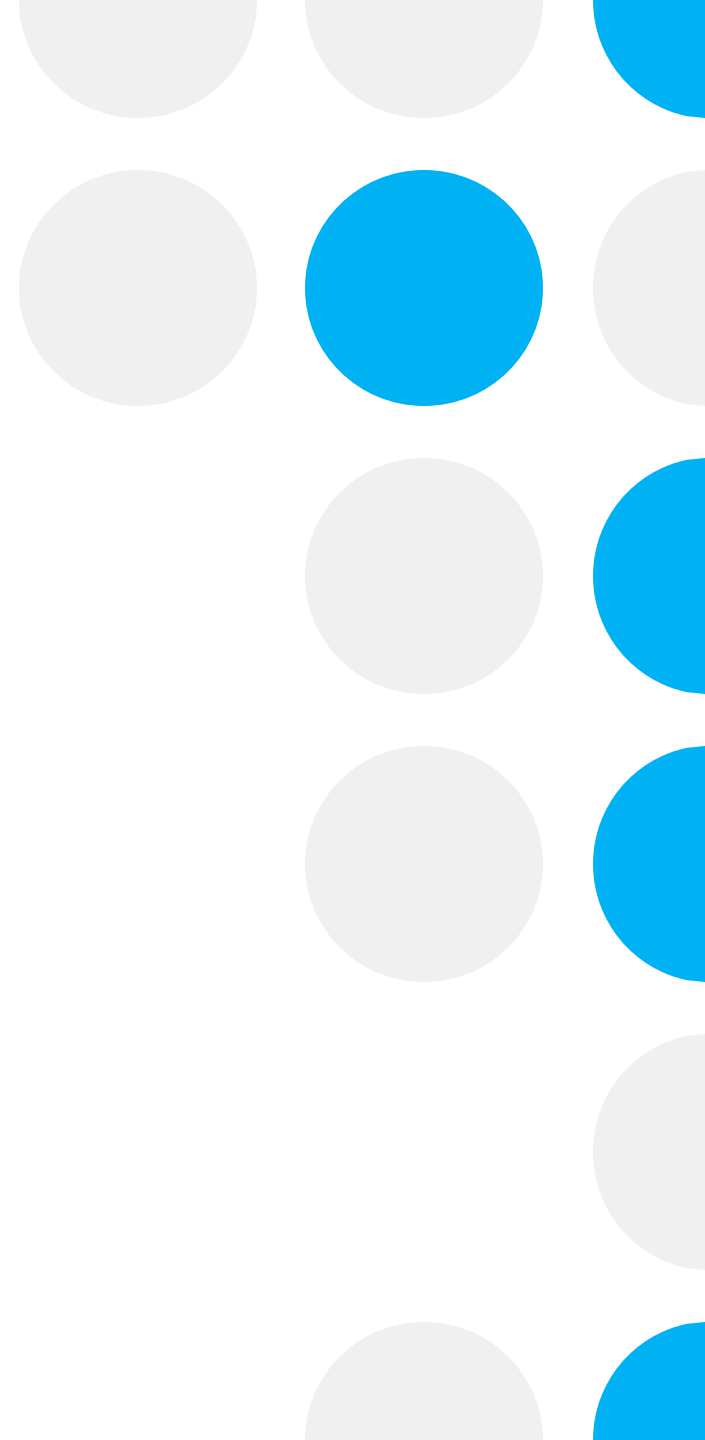
```
➤ <script>
➤   function initMap() {
➤     // Map options
➤     const mapOptions = {
➤       center: { lat: 37.7749, lng: -122.4194 }, // San Francisco
➤       zoom: 10,
➤     };
➤     // Create a map instance
➤     const map = new
google.maps.Map(document.getElementById('map'), mapOptions);
➤     // Add a marker
➤     const marker = new google.maps.Marker({
➤       position: { lat: 37.7749, lng: -122.4194 },
➤       map: map,
➤       title: "San Francisco",
➤     });
➤   }
```

- `// Initialize the map after the page loads`
 - `window.onload = initMap;`
 - `</script>`
 - `</body>`
 - `</html>`
 - **Pass the API Key from the View (Optional)**
 - If you want to pass the API key dynamically:
 - Modify the view to send the API key to the template:
 - `from django.shortcuts import render`
 - `from django.conf import settings`
 - `def map_view(request):`
 - `return render(request, 'map_template.html',`
 `{'GOOGLE_MAPS_API_KEY': settings.GOOGLE_MAPS_API_KEY})`
 - Update the `urls.py`:
 - `from django.urls import path`
 - `from .views import map_view`
-



- urlpatterns = [
 - path('map/', map_view, name='map'),
 -]
 - **Adding Interactivity**
 - You can enhance the map with the following features:
 - **Geocoding API:** Convert addresses to latitude and longitude.
 - **Places API:** Add search boxes for places.
 - **Markers and InfoWindows:** Show information about locations.
 - Example to add a search box:
 - `<input id="search-box" type="text" placeholder="Search a place">`
 - `<script>`
 - `function initMap() {`
 - `const mapOptions = {`
 - `center: { lat: 37.7749, lng: -122.4194 },`
 - `zoom: 10,`
 - `};`
-

- `const map = new google.maps.Map(document.getElementById('map'), mapOptions);`
- `const searchBox = new google.maps.places.SearchBox(document.getElementById('search-box'));`
- `// Bias the SearchBox results towards current map bounds.`
- `map.addListener('bounds_changed', () => {`
- `searchBox.setBounds(map.getBounds());`
- `});`
- `searchBox.addListener('places_changed', () => {`
- `const places = searchBox.getPlaces();`
- `if (places.length === 0) return;`
- `// Center the map on the first place`
- `const bounds = new google.maps.LatLngBounds();`
- `places.forEach((place) => {`
- `if (!place.geometry || !place.geometry.location) return;`
- `bounds.extend(place.geometry.location);`



- `});`
 - `map.fitBounds(bounds);`
 - `});`
 - `}`
 - `window.onload = initMap;`
 - `</script>`
 - **Test the Integration**
 - Run the Django server:
 - `python manage.py runserver`
 - **Deploying to Production**
 - Update the Google Maps API key settings to include your domain under **HTTP referrers** for security.
 - Set `DEBUG=False` in production and configure `ALLOWED_HOSTS`.
-

