Planning API Requirements:

ENDPOINT NAME: /Catering

Method: GET

Description: Fetches The Catering Menus & Deals To Be Rendered.

Response Example:

{ "Id": "1", "Name": "BBQ Grilla", "Details": "Chicken Biryani, Behari Boti, Seekh Kababs, Gola Kababs, Pasta, Salad", "Price": "$90" }

ENDPOINT NAME: /Decoration

Method: GET

Description: Fetches The Decoration Steps Info.

Response Example:

{ "Id": "2", "Name": "Engagement Pleasure", "Details": "A Modern Golden-Black Theme With An Innovative Visual Background Of The Stage Contrasting A Gota's Piece", "Price": "$150" }

ENDPOINT NAME: /Reservation

Method: GET

Description: Fetches The Location Of Reservation.

Response Example:

{ "Id": "3", "Name": "Loft 59", "Details": "A Scenic Beauty Environment With The Sparkling Sea Water Just Ahead" }

ENDPOINT NAME: /Bookinghotels

Method: GET

Description: Fetches Details About The Hotels For Booking.

Response Example:

{ "Id": "4", "Name": "PC HOTEL", "Details": "3 Rooms Air-Conditioned, Centralized Temperature With The Majestic View Of Airport From The Window" }

ENDPOINT NAME: /Customers

Method: GET And POST

Description: It Fetches Or Posts Fields To Customers.

Response Example:

{ Customers(Id, Name, Email Address, Phone No.) }

Method: POST

Description: It Adds New Orders Of The Customers.

Response Example: { Orderid, Customerid, Serviceid, Orderstatus, Orderdate, Rental Duration, Product Condition }

Schema For The Integration of Data from API to Handle Backend CMS :

```
import { defineField, defineType } from
'sanity';

export const orders = defineType({
name: 'orders',
type: 'document',
title: 'Orders',
fields: [
```

```
defineField({
name: 'orderId',
type: 'string',
title: "Order's ID",
}),
defineField({
name: 'customerId',
type: 'reference',
to: [{ type: 'customers' }],
title: "Customer's ID",
}),
defineField({
name: 'serviceId',
type: 'number',
title: 'Service ID',
}),
defineField({
name: 'orderStatus',
type: 'string',
title: 'Order Status',
options: {
list: ['Pending', 'Shipped', 'Good',
'Normal', 'Bad'],
},
}),
defineField({
name: 'orderDate',
type: 'date',
title: 'Order Date',
```

```
    }),
    defineField({
      name: 'rentalDuration',
      type: 'string',
      title: 'Rental Duration',
    }),
    defineField({
      name: 'routeCondition',
      type: 'string',
      title: 'Route Condition',
      options: {
        list: ['Good', 'Normal', 'Bad'],
      },
    }),
  ],
});

export const customers = defineType({
  name: 'customers',
  type: 'document',
  title: 'Customers',
  fields: [
    defineField({
      name: 'id',
      type: 'number',
      title: 'Customer ID',
    }),
    defineField({
      name: 'email',
```

```
    type: 'string',
    title: 'Email Address',
  }),
  defineField({
    name: 'phone',
    type: 'string',
    title: 'Phone Number',
  }),
  ],
});


export const rentalInfo = defineType({
  name: 'rentalInfo',
  type: 'document',
  title: 'Rental Info',
  fields: [
  defineField({
    name: 'rentalDuration',
    type: 'string',
    title: 'Rental Duration',
  }),
  defineField({
    name: 'rentalDetails',
    type: 'string',
    title: 'Rental Details',
  }),
  defineField({
    name: 'address',
```

```
    type: 'string',
    title: 'Address',
  }),
  defineField({
    name: 'date',
    type: 'date',
    title: 'Date',
  }),
  ],
});
// Schema for Decoration Service
export const decorationSchema = {
  name: 'Decoration',
  type: 'document',
  fields: [
  { name: 'id', type: 'number', title: 'ID'
  },
  { name: 'name', type: 'string', title:
  'Name' },
  { name: 'details', type: 'string', title:
  'Details' },
  { name: 'price', type: 'number', title:
  'Price' },
  ],
};

// Schema for Reserving Destination
export const reservingDestinationSchema =
{
```

```javascript
  name: 'Reserving',
  type: 'document',
  fields: [
  { name: 'id', type: 'string', title: 'ID'
  },
  { name: 'name', type: 'string', title:
  'Name' },
  { name: 'location', type: 'string',
  title: 'Location' },
  { name: 'availability', type: 'boolean',
  title: 'Availability' },
  { name: 'details', type: 'string', title:
  'Details' },
  ],
};

// Schema for Consultancy Manager
export const consultancyManagerSchema = {
  name: 'ConsultancyManagers',
  type: 'document',
  fields: [
  { name: 'id', type: 'number', title: 'ID'
  },
  { name: 'name', type: 'string', title:
  'Name' },
  { name: 'availability', type: 'boolean',
  title: 'Availability' },
  { name: 'specialization', type: 'string',
  title: 'Specialization' },
```

```
],
  };
```

System Architecture :