

# Day 3 - API Integration Report – SpeedyQEats

## API Integration Process

### Overview:

The integration involves connecting an external API that provides data for foods and chefs to a **Sanity CMS** project, allowing seamless data migration and management.

---

### 2. Steps Taken:

- **Environment Setup:**
  - Used **dotenv** to securely load environment variables from `.env.local`.
  - Key variables included:
    - `NEXT_PUBLIC_SANITY_PROJECT_ID`
    - `NEXT_PUBLIC_SANITY_DATASET`
    - `SANITY_TOKEN`
- **Sanity Client Creation:**
  - Utilized **@sanity/client** to establish a connection to the Sanity project.
  - Configured the client with the following credentials:
    - **Project ID**
    - **Dataset**
    - **API Version**
    - **Authentication Token**
- **Data Fetching:**
  - Made concurrent API calls using **axios** to fetch food and chef data from the endpoints:
    - **Foods API:** <https://sanity-nextjs-rouge.vercel.app/api/foods>
    - **Chefs API:** <https://sanity-nextjs-rouge.vercel.app/api/chefs>
- **Data Processing:**
  - Iterated through the fetched data to process it.
  - Uploaded images to **Sanity's asset library** using the `client.assets.upload()` method.
- **Sanity Document Creation:**
  - Transformed the fetched API data into Sanity-compatible document structures.
  - Uploaded each document to Sanity using `client.create()`.
- **Error Handling:**
  - Implemented **try-catch** blocks for handling errors during API calls and Sanity operations.
  - Logged errors for efficient debugging and troubleshooting.

### Adjustments Made to Schemas:

- **API Endpoints Used:**
  - **Foods:** <https://sanity-nextjs-rouge.vercel.app/api/foods>
  - **Chefs:** <https://sanity-nextjs-rouge.vercel.app/api/chefs>

**By Taha Saif**

# Day 3 - API Integration Report – SpeedyQEats

## 1. Chef Schema Enhancements

- **Image Field:** Enabled the hotspot option, allowing for flexible cropping and adaptable displays across different screen sizes.

### Key Fields Added:

- **Category:** A string field to categorize items (e.g., Burgers, Drinks).
- **Price:** A number field to specify the current price of each item.
- **Original Price:** An optional field for showing discounted prices.
- **Tags:** An array field for categorizing and tagging the food items.
- **Available:** A boolean field indicating stock availability.

These changes help provide better control over product data, ensuring a seamless and adaptable content management experience within **Sanity CMS**.

## Migration Steps and Tools Used

### 1. Migration Steps:

- **Preparation:**
  - Analyzed the API data structure and aligned it with the existing Sanity CMS schemas.
  - Created new Sanity schemas for food and chef, incorporating all necessary fields to support the migration process.
- **Data Import Script:**
  - Developed the `import-data.mjs` script to automate the process of data fetching, processing, and uploading to the CMS.
  - Used **axios** for making API calls to fetch data and **@sanity/client** for interacting with the Sanity CMS to upload and manage content.
- **Image Handling:**
  - Downloaded images from the API using **axios** with `responseType: 'arraybuffer'` to fetch raw image data.
  - Uploaded the images to **Sanity CMS** and stored the image references in the appropriate document fields.
- **Document Creation:**
  - Mapped the fields from the API data to the corresponding Sanity schema fields, ensuring proper alignment.
  - Implemented fallback values for optional or missing fields, ensuring data consistency.

### 2. Tools Used:

- **Node.js Modules:**
  - **axios:** Used for making HTTP requests to interact with the API.
  - **dotenv:** Managed environment variables securely for the integration.

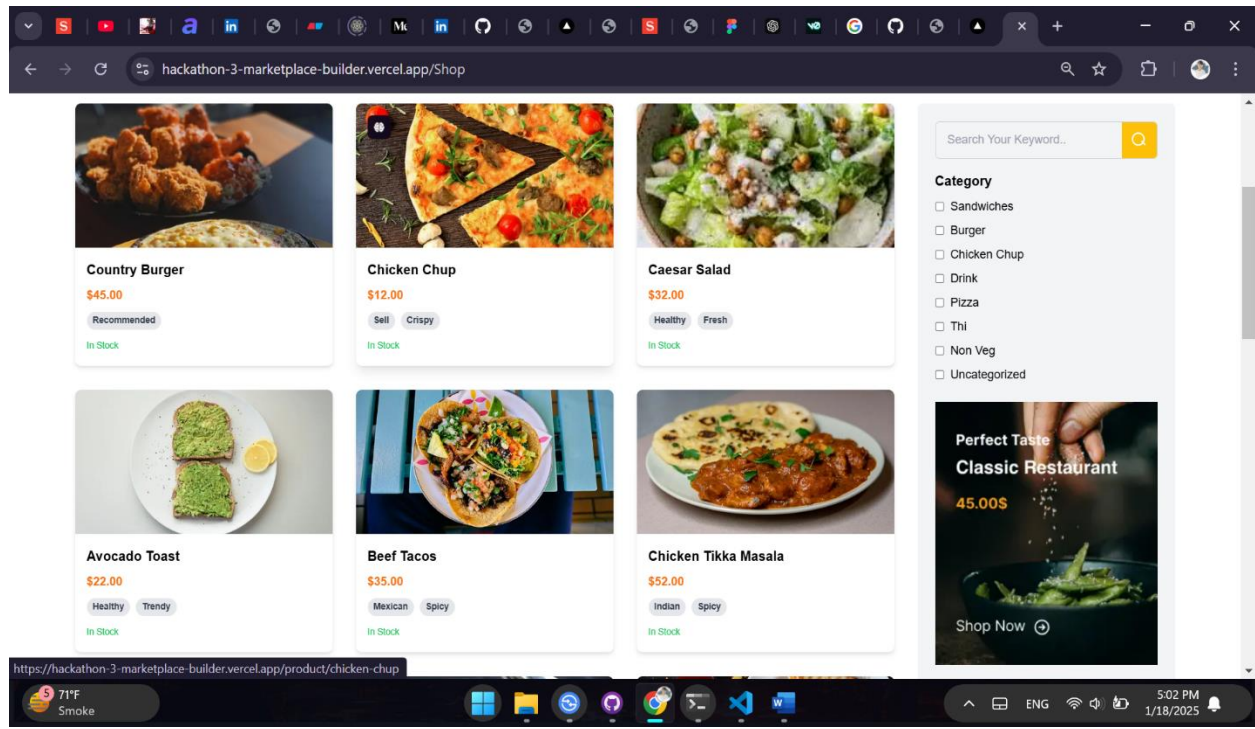
**By Taha Saif**

## Day 3 - API Integration Report – SpeedyQEats

- **@sanity/client**: Interfaced with the Sanity CMS for managing documents and assets.
- **Sanity Features:**
  - **Asset Management**: Utilized Sanity's asset management system to handle image uploads efficiently.
  - **API Versioning**: Ensured consistent schema support across different environments by using API versioning.
- **Utilities:**
  - **fileURLToPath** and **path**: Used these utilities for resolving file paths during the image import process, ensuring compatibility with the file system.

### Data successfully displayed in the frontend:

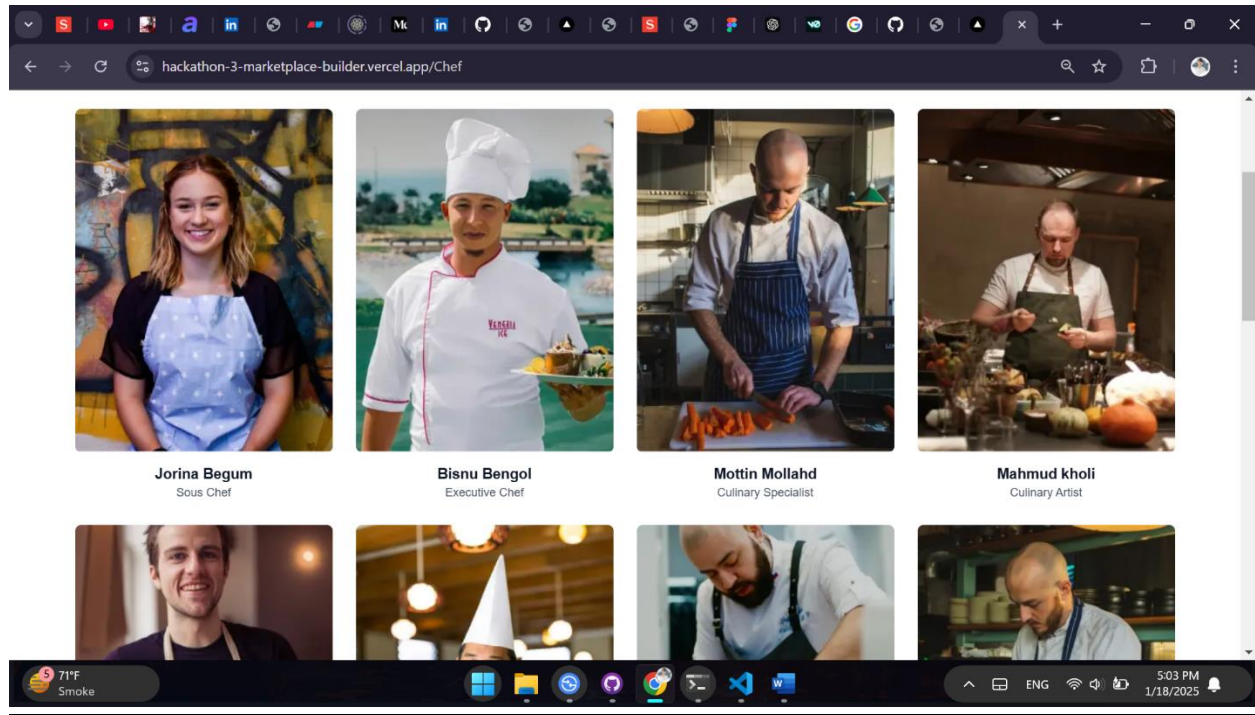
#### Foods Data:



#### Chefs Data:

By Taha Saif

## Day 3 - API Integration Report – SpeedyQEats



### Data Population in Sanity CMS:

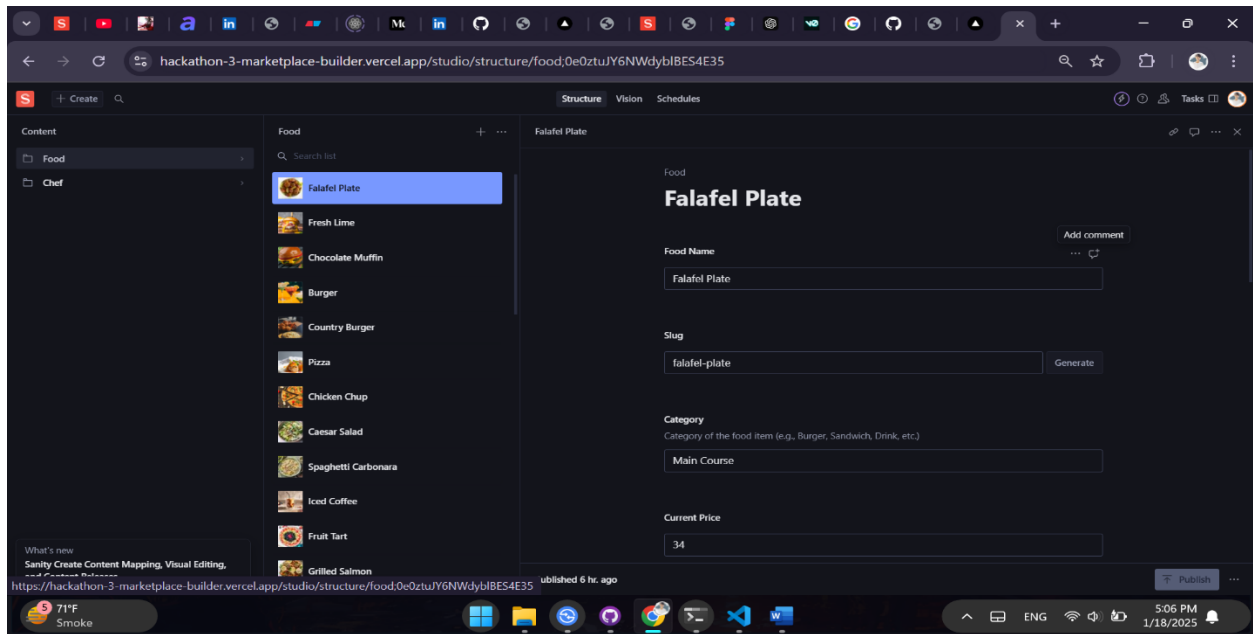
- **Chef Schema:**
  - **Position:** Added a string field to specify the chef's role.
  - **Experience:** Added a number field to capture the number of years of experience.
  - **Image:** Enabled the hotspot option to allow flexible cropping of images.
  - **Availability:** Added a boolean field to track whether the chef is available.
- **Food Schema:**
  - **Category:** Created a string field to categorize foods (e.g., Burgers, Drinks).
  - **Price:** Added a number field to store the current price of items.
  - **Original Price:** Included an optional number field for discounted prices.
  - **Tags:** Enabled an array field for food item tags (e.g., spicy, vegetarian).
  - **Available:** Added a boolean field to indicate food availability.

**Foods:**

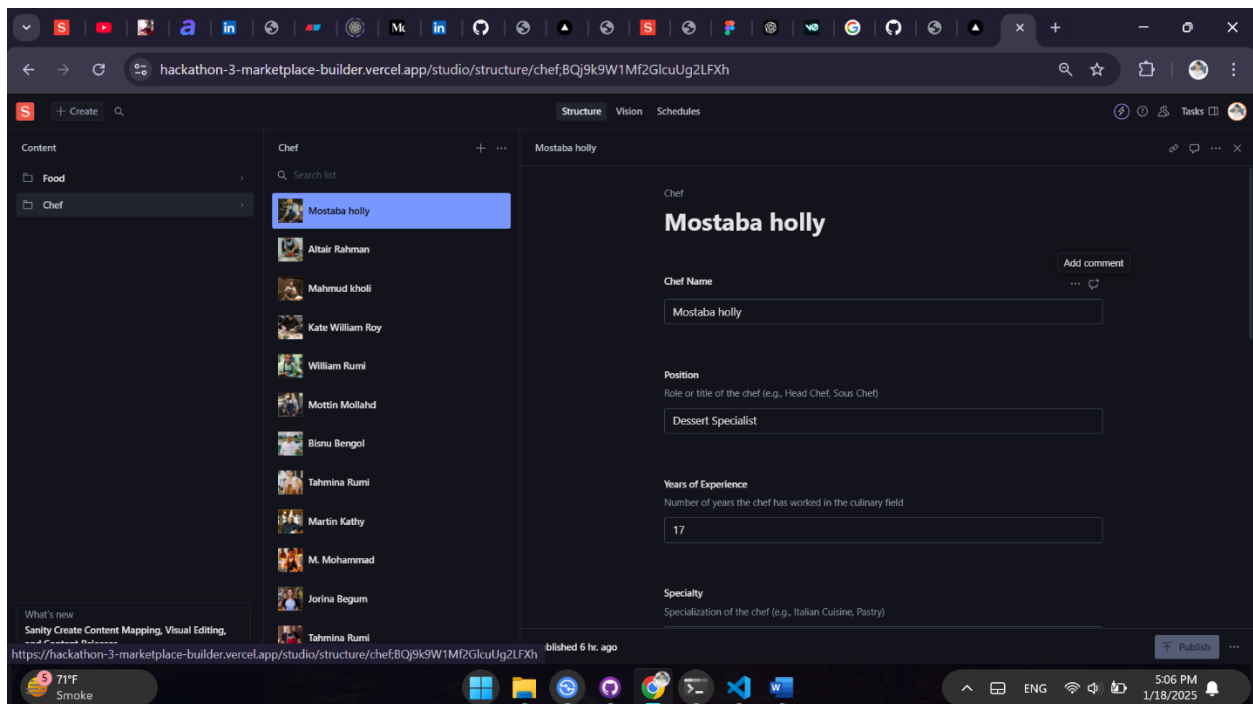
**Foods:**

**By Taha Saif**

# Day 3 - API Integration Report – SpeedyQEats



## Chefs:



## Code Snippets for API Integration and Migration Scripts

By Taha Saif

## Day 3 - API Integration Report – SpeedyQEats

```
1  import { createClient } from
   @sanity/client';
2  import axios from 'axios';
3  import dotenv from 'dotenv';
4  import { fileURLToPath } from 'url';
5  import path from 'path';
6
7  // Load environment variables from .env.local
8  const __filename = fileURLToPath(import.meta.url);
9  const __dirname = path.dirname(__filename);
10  dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12  // Create Sanity client
13  const client = createClient({
14    projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15    dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16    useCdn: false,
17    token: process.env.SANITY_API_TOKEN,
18    apiVersion: '2021-08-31',
19  });
20
21  async function uploadImageToSanity(imageUrl) {
22    try {
23      console.log(`Uploading image: ${imageUrl}`);
24      const response = await axios.get(
25        imageUrl, { responseType: 'arraybuffer',
26        }, { timeout: 10000 });
27      const buffer = Buffer.from(response.data);
28      const asset = await client.assets.upload('image', buffer, {
29        filename: imageUrl.split('/').pop(),
30      });
31      console.log(`Image uploaded successfully: ${asset._id}`);
32      return asset._id;
33    } catch (error) {
34      console.error('Failed to upload image:', imageUrl, error);
35      return null;
36    }
37  }
38
39  async function deleteAllDocuments(type) {
40    try {
41      const query = `*[_type == "${type}"]._id`
42      const ids = await client.fetch(query)
43      for (const id of ids) {
44        await client.delete(id)
45        console.log(`Deleted ${type} with ID: ${id}`)
46      }
47      console.log(`All ${type} documents deleted successfully`)
48    } catch (error) {
49      console.error(`Error deleting ${type} documents:`, error)
50    }
51  }
52
53  async function importData() {
54    try {
55      console.log('Fetching food, chef data from API...');
56      // Delete existing documents
57      await deleteAllDocuments('food')
58      await deleteAllDocuments('chef')
59      // API endpoint containing data
60      const $Promise = [];
61      $Promise.push(
62        axios.get(
63          'http://localhost:3000/api/foods'
64        )
65      );
66      $Promise.push(
67        axios.get(
68          'http://localhost:3000/api/chefs'
69        )
70      );
71    } catch (error) {
72      console.error('Error importing data:', error);
73    }
74  }
```

**Foods:**

**By Taha Saif**

## Day 3 - API Integration Report – SpeedyQEats

```
1 import { defineField, defineType } from
  'sanity'
2
3 export default defineType({
4   name: 'food',
5   type: 'document',
6   title: 'Food',
7   fields: [
8     {
9       name: 'name',
10      type: 'string',
11      title: 'Food Name',
12    },
13    {
14      name: 'slug',
15      type: 'slug',
16      title: 'Slug',
17      options: {
18        source: 'name',
19        maxLength: 96,
20      },
21      validation: Rule => Rule.required()
22    },
23    {
24      name: 'category',
25      type: 'string',
26      title: 'Category',
27      description:
28        'Category of the food item (e.g., Burger, Sandwich, Drink, etc.)'
29    },
30    {
31      name: 'price',
32      type: 'number',
33      title: 'Current Price',
34    },
35    {
36      name: 'originalPrice',
37      type: 'number',
38      title: 'Original Price',
39      description:
40        'Price before discount (if any)',
41    },
42    {
43      name: 'tags',
44      type: 'array',
45      title: 'Tags',
46      of: [{ type: 'string' }],
47      options: {
48        layout: 'tags',
49      },
50      description:
51        'Tags for categorization (e.g., Best Seller, Popular, New)'
52    },
53    {
54      name: 'image',
55      type: 'image',
56      title: 'Food Image',
57      options: {
58        hotspot: true,
59      },
60    },
61    {
62      name: 'description',
63      type: 'text',
64      title: 'Description',
65      description:
66        'Short description of the food item',
67    },
68    {
69      name: 'available',
70      type: 'boolean',
71      title: 'Available',
72      description:
73        'Availability status of the food item',
74    },
75  ],
76 })
```

**Chefs:**

**By Taha Saif**



## Day 3 - API Integration Report – SpeedyQEats

```
1 export default {
2   name: 'chef',
3   type: 'document',
4   title: 'Chef',
5   fields: [
6     {
7       name: 'name',
8       type: 'string',
9       title: 'Chef Name',
10    },
11    {
12      name: 'position',
13      type: 'string',
14      title: 'Position',
15      description:
16        'Role or title of the chef (e.g., Head Chef, Sous Chef)'
17    },
18    {
19      name: 'experience',
20      type: 'number',
21      title: 'Years of Experience',
22      description:
23        'Number of years the chef has worked in the culinary field'
24    },
25    {
26      name: 'specialty',
27      type: 'string',
28      title: 'Specialty',
29      description:
30        'Specialization of the chef (e.g., Italian Cuisine, Pastry)'
31    },
32    {
33      name: 'image',
34      type: 'image',
35      title: 'Chef Image',
36      options: {
37        hotspot: true,
38      },
39    },
40    {
41      name: 'description',
42      type: 'text',
43      title: 'Description',
44      description:
45        'Short bio or introduction about the chef',
46    },
47    {
48      name: 'available',
49      type: 'boolean',
50      title: 'Currently Active',
51      description:
52        'Availability status of the chef',
53    },
54  ],
55 }
```

API Calls

Food:

By Taha Saif



## Day 3 - API Integration Report – SpeedyQEats

```
[
  {
    name: 'Chicken Chup',
    category: 'Appetizer',
    price: 12,
    originalPrice: 15,
    image: { _type: 'image', asset: [Object] },
    description: 'Crispy fried chicken bites served with dipping sauce.',
    available: true,
    tags: [ 'Sell', 'Crispy' ]
  },
  {
    name: 'Fresh Lime',
    category: 'Drink',
    price: 38,
    originalPrice: 45,
    image: { _type: 'image', asset: [Object] },
    description: 'Refreshing fresh lime drink made with natural ingredients.',
    available: true,
    tags: [ 'Healthy', 'Popular' ]
  },
  {
    available: true,
    tags: [ 'Sell', 'Sweet' ],
    name: 'Chocolate Muffin',
    category: 'Dessert',
    price: 28,
    originalPrice: 30,
    image: { _type: 'image', asset: [Object] },
    description: 'Soft and rich chocolate muffin topped with chocolate chips.'
  },
  {
    name: 'Country Burger',
    category: 'Sandwich',
    price: 45,
    originalPrice: 50,
    image: { _type: 'image', asset: [Object] },
    description: 'Classic country-style burger served with fries.',
    available: true,
    tags: [ 'Recommended' ]
  },
  {
    image: { _type: 'image', asset: [Object] },
    description: 'Juicy beef burger with fresh lettuce, tomatoes, and cheese.',
    available: true,
    tags: [ 'Popular' ],
    name: 'Burger',
    category: 'Sandwich',
    price: 21,
    originalPrice: 45
  },
]
```

**Chefs:**

**By Taha Saif**

## Day 3 - API Integration Report – SpeedyQEats

```
[
  {
    position: 'Head Chef',
    experience: 12,
    specialty: 'Italian Cuisine',
    image: { _type: 'image', asset: [Object] },
    description: 'Expert in crafting authentic Italian dishes and pastries.',
    available: true,
    name: 'Tahmina Rumi'
  },
  {
    description: 'Renowned for creating perfectly grilled meats and vegetables.',
    available: true,
    name: 'M. Mohammad',
    position: 'Grill Master',
    experience: 10,
    specialty: 'Grilled Dishes',
    image: { asset: [Object], _type: 'image' }
  },
  {
    image: { _type: 'image', asset: [Object] },
    description: 'Expert in international cuisines and menu planning.',
    available: true,
    name: 'Bisnu Devgon',
    position: 'Executive Chef',
    experience: 20,
    specialty: 'Global Cuisine'
  },
  {
    position: 'Chef de Cuisine',
    experience: 18,
    specialty: 'Seafood Specialties',
    image: { _type: 'image', asset: [Object] },
    description: 'Master of crafting exquisite seafood dishes with unique flavors.',
    available: true,
    name: 'William Rumi'
  },
  {
    available: true,
    name: 'Jorina Begum',
    position: 'Sous Chef',
    experience: 8,
    specialty: 'Pastry and Desserts',
    image: { _type: 'image', asset: [Object] },
    description: 'Specializes in creative pastries and dessert innovations.'
  },
]
```

Final check:

API Understanding	Schema Validation	Data Migration	API Integration in Next.js	Submission Preparation
✓	✓	✓	✓	✓

**By Taha Saif**

## **Day 3 - API Integration Report – SpeedyQEats**

**By Taha Saif**