## Project Ideas

### 1. Real Estate App

\*\*Description:\*\* Create a web application that allows users to search for properties available for sale in the United States using the Realty API.

\*\*Key Features:\*\*

- Search properties by location, price range, and type.

- Display property details including images, price, and description.

- Filter results based on user preferences.

\*\*Question:\*\* How can you fetch and display a list of properties using the Realty API?

\*\*Solution:\*\*

```javascript

const apiKey = 'YOUR\_API\_KEY';

const apiUrl = `https://api.realty.com/properties?api\_key=${apiKey}&location=New York`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const properties = data.properties;

properties.forEach(property => {

// Create HTML elements to display property details

const propertyElement = document.createElement('div');

propertyElement.innerHTML = `

<h2>${property.name}</h2>

<img src="${property.image}" alt="${property.name}">

<p>Price: ${property.price}</p>

<p>${property.description}</p>

`;

document.body.appendChild(propertyElement);

});

})

.catch(error => console.error('Error fetching properties:', error));

```

### 2. Food Recipe App

\*\*Description:\*\* Build a web application that allows users to search for food recipes using the Spoonacular API.

\*\*Key Features:\*\*

- Search recipes by ingredients, cuisine, or meal type.

- Display recipe details including ingredients, instructions, and nutritional information.

- Save favorite recipes.

\*\*Question:\*\* How can you search for recipes using the Spoonacular API and display the results?

\*\*Solution:\*\*

```javascript

const apiKey = 'YOUR\_API\_KEY';

const apiUrl = `https://api.spoonacular.com/recipes/complexSearch?apiKey=${apiKey}&query=pasta`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const recipes = data.results;

recipes.forEach(recipe => {

// Create HTML elements to display recipe details

const recipeElement = document.createElement('div');

recipeElement.innerHTML = `

<h2>${recipe.title}</h2>

<img src="${recipe.image}" alt="${recipe.title}">

<p>Ready in: ${recipe.readyInMinutes} minutes</p>

<a href="https://spoonacular.com/recipes/${recipe.id}">View Recipe</a>

`;

document.body.appendChild(recipeElement);

});

})

.catch(error => console.error('Error fetching recipes:', error));

```

### 3. Crypto News App

\*\*Description:\*\* Develop a web application that displays the latest news related to cryptocurrencies using the Investing - Cryptocurrency Markets API.

\*\*Key Features:\*\*

- Display the latest news articles about cryptocurrencies.

- Filter news by specific cryptocurrencies.

- Show article summaries and links to full articles.

\*\*Question:\*\* How can you fetch and display the latest cryptocurrency news using the Investing - Cryptocurrency Markets API?

\*\*Solution:\*\*

```javascript

const apiKey = 'YOUR\_API\_KEY';

const apiUrl = `https://api.investing.com/crypto-news?apiKey=${apiKey}`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const newsArticles = data.articles;

newsArticles.forEach(article => {

// Create HTML elements to display news article details

const articleElement = document.createElement('div');

articleElement.innerHTML = `

<h2>${article.title}</h2>

<p>${article.summary}</p>

<a href="${article.url}">Read more</a>

`;

document.body.appendChild(articleElement);

});

})

.catch(error => console.error('Error fetching news:', error));

```

### 4. Streaming Availability Checker

\*\*Description:\*\* Create a web application that shows the availability of movies or TV shows across different streaming services using the OTT Details API.

\*\*Key Features:\*\*

- Search for movies or TV shows by title.

- Display availability on streaming platforms like Netflix, Hulu, HBO, etc.

- Show details such as release date, genre, and rating.

\*\*Question:\*\* How can you check the availability of a movie or TV show using the OTT Details API?

\*\*Solution:\*\*

```javascript

const apiKey = 'YOUR\_API\_KEY';

const apiUrl = `https://api.ott-details.com/search?apiKey=${apiKey}&title=Inception`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const results = data.results;

results.forEach(result => {

// Create HTML elements to display movie/TV show details

const resultElement = document.createElement('div');

resultElement.innerHTML = `

<h2>${result.title}</h2>

<p>Available on: ${result.platforms.join(', ')}</p>

<p>Release Date: ${result.releaseDate}</p>

<p>Genre: ${result.genre}</p>

<p>Rating: ${result.rating}</p>

`;

document.body.appendChild(resultElement);

});

})

.catch(error => console.error('Error fetching availability:', error));

```

### 5. Email Validation App

\*\*Description:\*\* Build an application that validates email addresses using the Email Check API.

\*\*Key Features:\*\*

- Input email address for validation.

- Display validation results (valid/invalid, domain information, etc.).

- Show suggestions for correcting invalid email addresses.

\*\*Question:\*\* How can you validate an email address using the Email Check API?

\*\*Solution:\*\*

```javascript

const apiKey = 'YOUR\_API\_KEY';

const email = 'test@example.com';

const apiUrl = `https://api.email-check.com/validate?apiKey=${apiKey}&email=${email}`;

fetch(apiUrl)

.then(response => response.json())

.then(data => {

const validationResult = data.isValid ? 'Valid' : 'Invalid';

// Create HTML elements to display validation result

const resultElement = document.createElement('div');

resultElement.innerHTML = `

<h2>Email Validation Result</h2>

<p>Email: ${email}</p>

<p>Result: ${validationResult}</p>

<p>Domain: ${data.domain}</p>

<p>Suggestions: ${data.suggestions.join(', ')}</p>

`;

document.body.appendChild(resultElement);

})

.catch(error => console.error('Error validating email:', error));

```