Day 5 –-Testing and Backend Refinement –

SHOP.CO

Key Focus Areas

• Comprehensive Functional Testing:

o Ensuring seamless operation of all marketplace functionalities, such as product

listing, filters, and checkout workflows.

• Robust Error Handling:

o Implementing user-friendly fallback mechanisms for API errors and dynamic error messages.

• Performance Optimization:

o Enhancing speed and responsiveness with tools like Lighthouse and implementing caching strategies.

• Cross-Device Compatibility:

o Testing across major browsers and devices for a consistent user experience.

• Security and User Acceptance Testing (UAT):

o Preventing vulnerabilities and gathering feedback for real-world usability improvements

Implementation Plan:

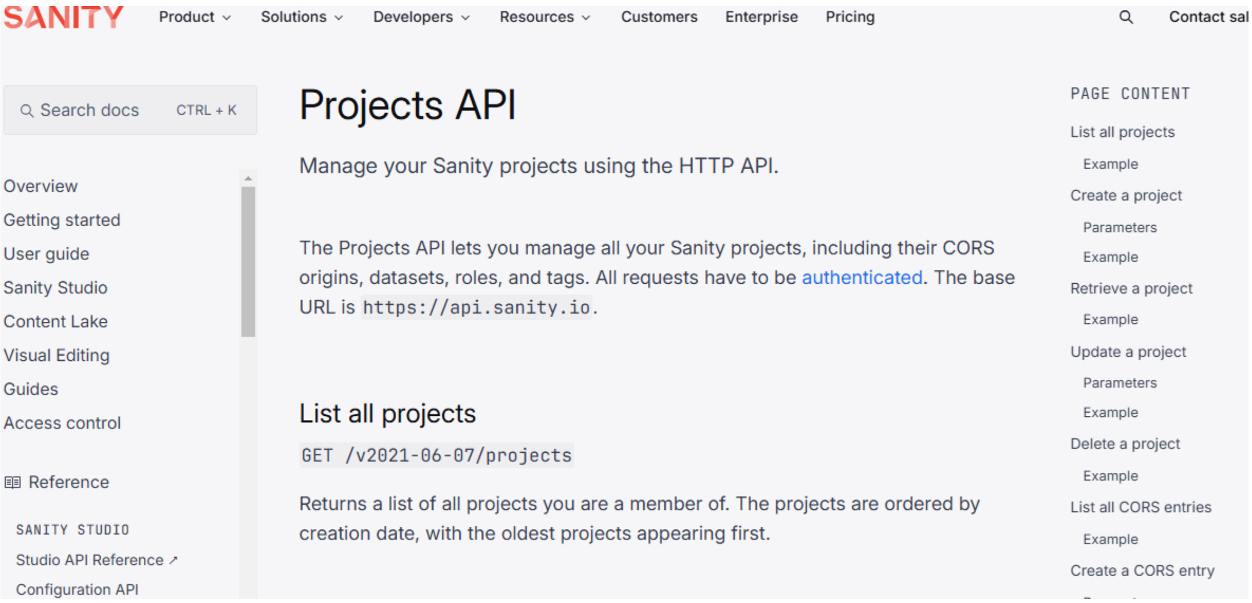
1. Functional Testing:

Goal: Validate key features, including:

- Product listing and filtering.
- Cart management.
- Secure user account operations.

Tools Used:

- Postman: For API responses.



Projects API

Manage your Sanity projects using the HTTP API.

The Projects API lets you manage all your Sanity projects, including their CORS origins, datasets, roles, and tags. All requests have to be authenticated. The base URL is `https://api.sanity.io`.

List all projects

```
GET /v2021-06-07/projects
```

Returns a list of all projects you are a member of. The projects are ordered by creation date, with the oldest projects appearing first.

PLATFORM

Sanity Studio
Flexible editing environment
APIs
Connect to anything
Content Lake

Fully decoupled back end

Sanity Create

Better writing experience

Try product demo

FEATURES

Real-time collaboration

Fearlessly work with content

Precise content querying

Treat content as data with GROQ

Localization

Coherent messaging across territories

AI Assist

Built-in AI assistant

Search docs

2. Enhanced Error Handling:

- Performance Optimization:

o Enhancing speed and responsiveness with tools like Lighthouse and implementing caching strategies.

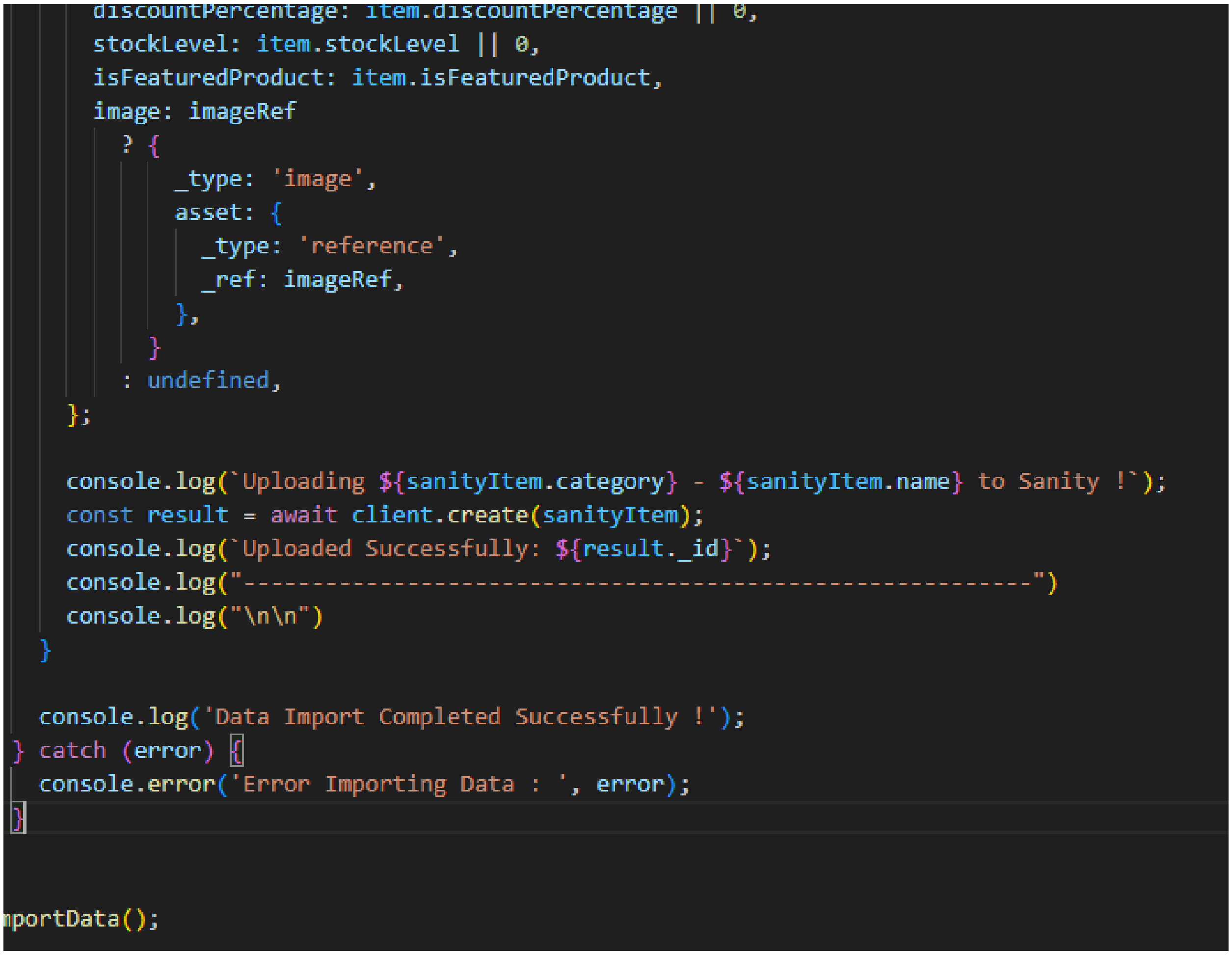
- Cross-Device Compatibility:

o Testing across major browsers and devices for a consistent user experience.

- Security and User Acceptance Testing (UAT):

o Preventing vulnerabilities and gathering feedback for real-world usability improvements

Implementation Plan:



```
        discountPercentage: item.discountPercentage || 0,
        stockLevel: item.stockLevel || 0,
        isFeaturedProduct: item.isFeaturedProduct,
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
      };

      console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);
      const result = await client.create(sanityItem);
      console.log(`Uploaded Successfully: ${result._id}`);
      console.log("----------------------------------------------------------")
      console.log("\n\n")
    }

    console.log('Data Import Completed Successfully !');
  } catch (error) {
    console.error('Error Importing Data : ', error);
  }


mportData();
```

3. Performance Optimization:

- Used Priority To Image Tags To load Images before rendering

- Image compression and lazy loading for faster page loads.

- Implementing browser caching for repeat visits.

- Performance Optimization:

o Enhancing speed and responsiveness with tools like Lighthouse and implementing caching strategies.

- Cross-Device Compatibility:

o Testing across major browsers and devices for a consistent user experience.

- Security and User Acceptance Testing (UAT):

o Preventing vulnerabilities and gathering feedback for real-world usability improvements

Implementation Plan:

- Performance Optimization:

o Enhancing speed and responsiveness with tools like Lighthouse and implementing caching strategies.

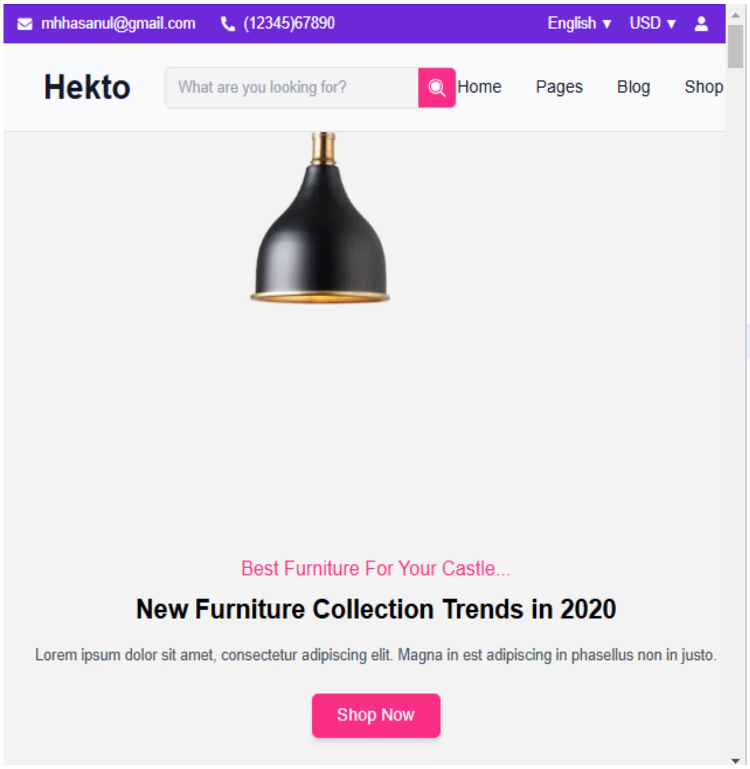
- Cross-Device Compatibility:

o Testing across major browsers and devices for a consistent user experience.

- Security and User Acceptance Testing (UAT):

o Preventing vulnerabilities and gathering feedback for real-world usability improvements

Implementation Plan:

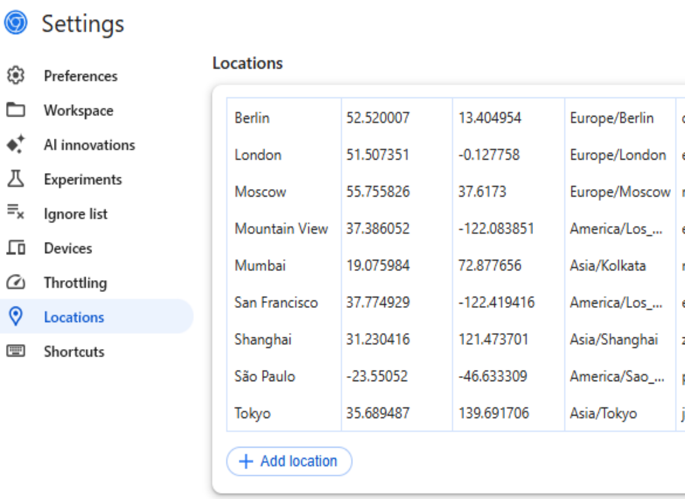


Best Furniture For Your Castle...

New Furniture Collection Trends in 2020

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Magna in est adipiscing in phasellus non in justo.

Shop Now



Locations

Berlin	52.520007	13.404954	Europe/Berlin	c
London	51.507351	-0.127758	Europe/London	e
Moscow	55.755826	37.6173	Europe/Moscow	r
Mountain View	37.386052	-122.083851	America/Los_...	e
Mumbai	19.075984	72.877656	Asia/Kolkata	r
San Francisco	37.774929	-122.419416	America/Los_...	e
Shanghai	31.230416	121.473701	Asia/Shanghai	z
São Paulo	-23.55052	-46.633309	America/Sao_...	p
Tokyo	35.689487	139.691706	Asia/Tokyo	j

+ Add location

4. Cross-Browser and Device Testing:

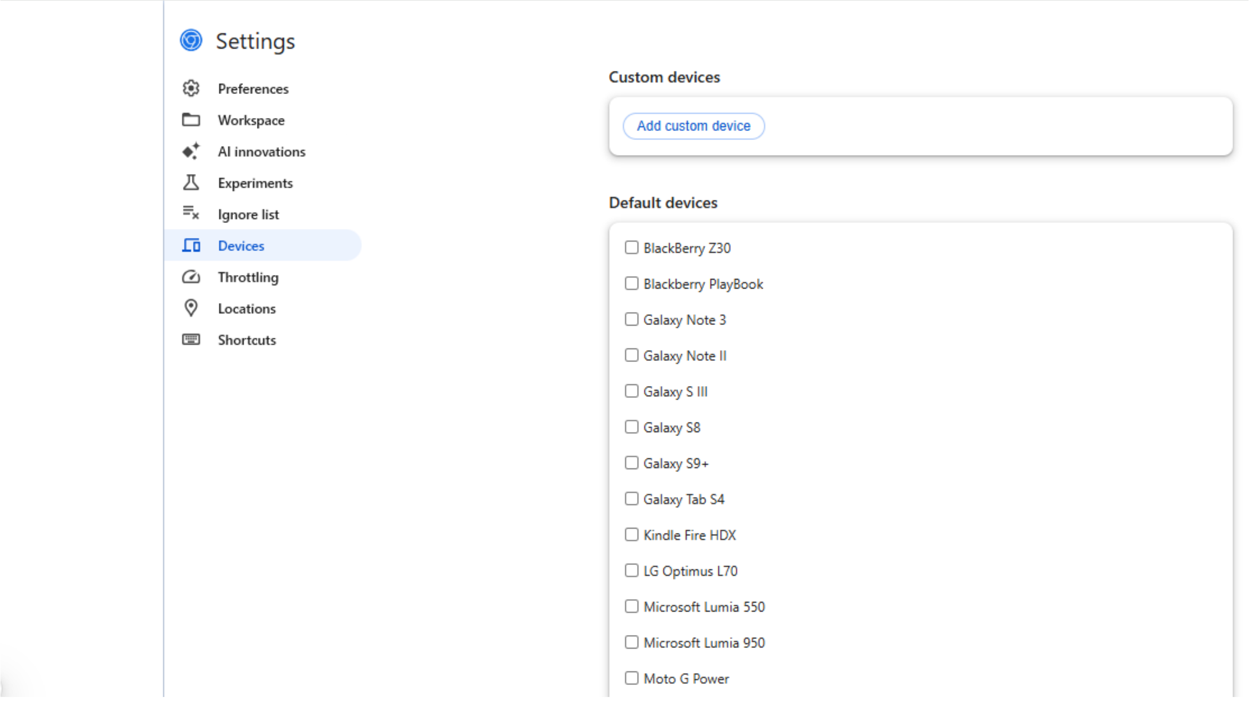
optimization details prepared.

This checklist serves as a testament to the progress made and ensures a high-quality,

deployment-ready application. Moving forward, the foundation laid here will pave the

way for seamless integration and continuous improvements.

Steps



Custom devices

Default devices

☐ BlackBerry Z30

☐ Blackberry PlayBook

☐ Galaxy Note 3

☐ Galaxy Note II

☐ Galaxy S III

☐ Galaxy S8

☐ Galaxy S9+

☐ Galaxy Tab S4

☐ Kindle Fire HDX

☐ LG Optimus L70

☐ Microsoft Lumia 550

☐ Microsoft Lumia 950

☐ Moto G Power

- Use BrowserStack for virtual device testing.
- Manual tests on physical mobile and desktop devices.

Results:

- Websites is working fine on all browsers.

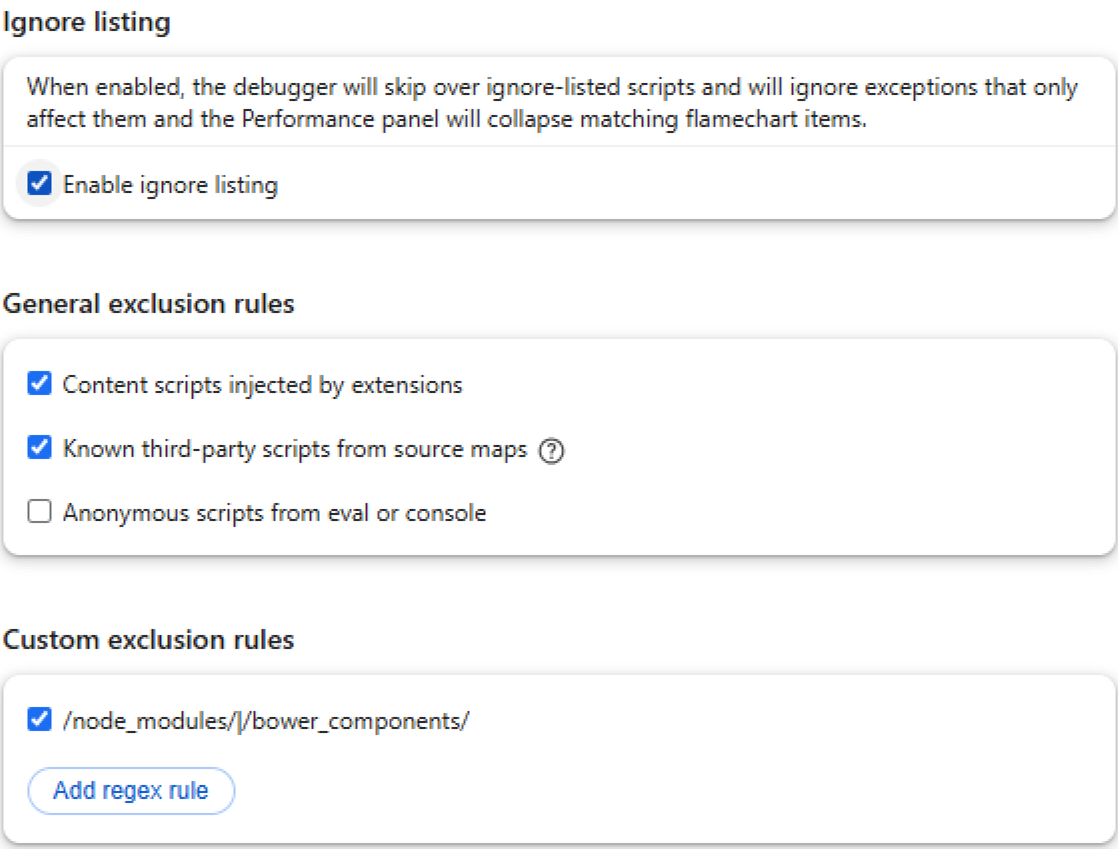
6. User Acceptance Testing (UAT):

Purpose: Simulate real-world user behavior.

- Test scenarios: Browsing categories, adding to the cart, and completing purchases.

Results: Websites all the functionalities are working well from login to checkout

7. CSV-based testing report.



Ignore listing

When enabled, the debugger will skip over ignore-listed scripts and will ignore exceptions that only affect them and the Performance panel will collapse matching flamechart items.

☑ Enable ignore listing

General exclusion rules

☑ Content scripts injected by extensions

☑ Known third-party scripts from source maps ⓘ

□ Anonymous scripts from eval or console

Custom exclusion rules

☑ /node_modules/|/bower_components/

Final Thoughts:

Day 5 has been a pivotal phase in ensuring the readiness of our marketplace for real world deployment. By focusing on testing, error handling, and optimization, we have built a robust and user-friendly platform. Each improvement has enhanced the functionality, security, and overall user experience of the application.

As we wrap up, here's a final checklist of accomplishments for Day 5:

✅ Functional Testing: All core features tested and validated.

✅ Error Handling: User-friendly error messages and fallback mechanisms

✅ Performance Optimization: Faster page loads and improved responsiveness

✅ Cross-Browser and Device Testing: Consistent functionality verified across

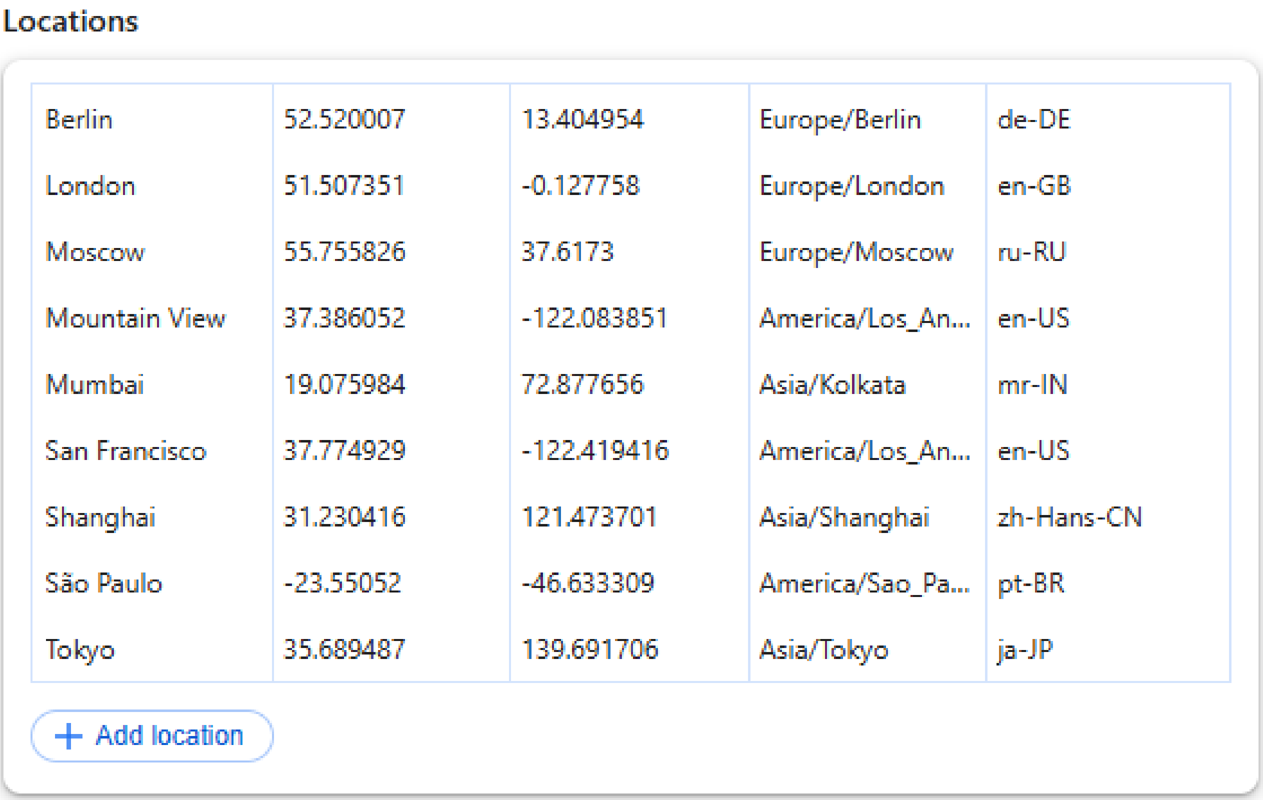
✅ User Acceptance Testing: Real-world scenarios tested with valuable feedback

✅ Documentation: Professional test reports and optimization details prepared.

This checklist serves as a testament to the progress made and ensures a high-quality,

deployment-ready application. Moving forward, the foundation laid here will pave the

way for seamless integration and continuous improvements.



Locations

Berlin	52.520007	13.404954	Europe/Berlin	de-DE
London	51.507351	-0.127758	Europe/London	en-GB
Moscow	55.755826	37.6173	Europe/Moscow	ru-RU
Mountain View	37.386052	-122.083851	America/Los_An...	en-US
Mumbai	19.075984	72.877656	Asia/Kolkata	mr-IN
San Francisco	37.774929	-122.419416	America/Los_An...	en-US
Shanghai	31.230416	121.473701	Asia/Shanghai	zh-Hans-CN
São Paulo	-23.55052	-46.633309	America/Sao_Pa...	pt-BR
Tokyo	35.689487	139.691706	Asia/Tokyo	ja-JP