Project Design Phase-**||**

**OPEN SOURCE FRAMEWORKS**

**--------------------------------------------------------------------------------------------**

|  |  |
| --- | --- |
| NAME | NAVEEN. V |
| NM ID | C259747173E9454BEE44375156B7964E |
| PROJECT | SEARCH ENGINE OPTIMIZATION |

**1. \*\*Google Ads API Client Libraries:\*\* Google provides official client libraries for various programming languages (e.g., Python, Java, PHP) that allow you to interact with the Google Ads API. These libraries are open-source and can help you streamline campaign management tasks programmatically.**

**2. \*\*Google Ads Scripts:\*\* Google Ads allows you to create and run custom JavaScript scripts within your Google Ads account. These scripts can automate various tasks like bid management, reporting, and ad optimization. While Google Ads Scripts are not a standalone framework, they provide a powerful way to enhance your campaigns.**

**3. \*\*Google Ads API PHP Client:\*\* If you're working with PHP, you can use the open-source Google Ads API PHP Client library. It simplifies interaction with the Google Ads API and offers code examples and documentation.**

**4. \*\*Open-source Reporting and Analytics Tools:\*\* Tools like Google Data Studio, which is free and can connect to your Google Ads data, can help you create custom reports and visualizations for campaign performance analysis.**

**5. \*\*Open-source Data Analysis and Machine Learning Libraries:\*\* Libraries like Pandas, NumPy, and scikit-learn in Python can be used for data analysis, and you can apply machine learning techniques to optimize your campaigns.**

**6. \*\*Web Development Frameworks:\*\* If you're building a custom campaign management system, you might consider open-source web development frameworks like Django (Python), Ruby on Rails (Ruby), or Laravel (PHP) to build your application.**

**7. \*\*Content Management Systems (CMS):\*\* If your campaign involves managing a website, open-source CMS like WordPress, Drupal, or Joomla can be helpful for content updates and landing page creation.**

**8. \*\*Data Storage and Databases:\*\* Open-source databases like MySQL, PostgreSQL, or NoSQL databases like MongoDB can be used for storing campaign data securely.**

**9. \*\*Containerization and Orchestration:\*\* Technologies like Docker and Kubernetes, which are open source, can help manage your application's deployment and scalability.**

**10. \*\*Monitoring and Logging:\*\* Open-source monitoring tools like Prometheus and Grafana can be used for real-time system health and performance monitoring.**

**SOURCE CODE:**

**from google.auth import exceptions**

**from google.auth.exceptions import RefreshError**

**from google.auth.transport.requests import Request**

**from google.ads.google\_ads.errors import GoogleAdsException**

**from google.ads.google\_ads.client import GoogleAdsClient**

**from google.ads.google\_ads.errors.google\_ads\_exception import load\_from\_storage**

**from google.auth.transport.requests import Request**

**# Set up the Google Ads API client**

**try:**

**client = GoogleAdsClient.load\_from\_storage("/path/to/your/config\_file.yaml")**

**except exceptions.RefreshError as ex:**

**client = GoogleAdsClient.load\_from\_storage("/path/to/your/config\_file.yaml")**

**client.client.renew\_credentials(Request())**

**except GoogleAdsException as ex:**

**print(f"Google Ads API request failed: {ex.message}")**

**exit(1)**

**# Create a campaign**

**def create\_campaign(client, customer\_id):**

**try:**

**campaign\_service = client.service.campaign**

**campaign\_operation = client.operation**

**campaign = campaign\_operation.create\_resource.campaign**

**campaign.name = "My Campaign"**

**campaign.advertising\_channel\_type = client.enums.AdvertisingChannelTypeEnum.SEARCH**

**campaign.status = client.enums.CampaignStatusEnum.PAUSED**

**campaign.manual\_cpc.enhanced\_cpc\_enabled = False**

**campaign.campaign\_budget = "INSERT\_YOUR\_CAMPAIGN\_BUDGET\_ID"**

**campaign.network\_settings.target\_google\_search = True**

**campaign.network\_settings.target\_search\_network = True**

**campaign.network\_settings.target\_content\_network = False**

**campaign.network\_settings.target\_partner\_search\_network = False**

**response = campaign\_service.mutate\_campaigns(**

**customer\_id=customer\_id, operations=[campaign\_operation]**

**)**

**print(f"Created campaign: {response.results[0].resource\_name}")**

**except GoogleAdsException as ex:**

**print(f"Google Ads API request failed: {ex.message}")**

**exit(1)**

**# Replace the following with your Google Ads API credentials**

**customer\_id = "INSERT\_YOUR\_CUSTOMER\_ID"**

**create\_campaign(client, customer\_id)**