CITL
EXPERIMEN
T 5

Name:

Pranav Sati - 2021300107 Div B Batch C

Virinchi Shettigar - 2021300118

Vignesh Shinde - 2021300120

Jyoti Bagul - 2022300112

1. **Aim:**

Create a web mashup of web services using open source framework

Theory/Methodology:

A web service is a software system designed to allow different applications to communicate with each other over the internet or a network. It provides a standardized way for software components, regardless of their programming languages or platforms, to exchange data and perform various tasks. Web services enable the integration of different systems, making it possible to create distributed and interoperable applications.

APIs that provide services that a normal developer couldn't feasibly implement offer a wide range of benefits. These APIs typically serve as abridge between developers and complex, specialized services or data sources.

WEB MASHUP

A web mashup is a web application or web page that combines contentand functionality from multiple sources or web services to create a

unified and enriched user experience. In a web mashup, data or services from different websites or APIs are integrated to provide users with a single, cohesive interface and access to various types of information or services. The term "mashup" comes from the idea of mixing or combiningelements from various sources, similar to how DJs create music remixes by blending different tracks.

Benefits of Web mashup:

- Web mashups consolidate data from multiple sources for userconvenience.
- They create a more user-friendly and engaging experience withunified interfaces.
- Mashups aggregate data from diverse providers into a singleplatform.
- Users can customize web mashups to match their specific needs andpreferences.

- They stimulate innovation and creativity among developers.

- Mashups serve as a single entry point for accessing distributedinformation and services.
- Developers benefit from efficiency gains by reusing existing APIs andservices.
- They are compatible with various platforms, ensuring widespreadaccessibility.
- Mashups offer economic value for businesses, improving userengagement and retention.
- Real-time data integration is a strong suit of web mashups, making them ideal for applications requiring dynamic information.

Types of Web Mashup:

Server-Side Mashups:
Server-side mashups involve data integration and processing on the webserver that hosts the application. The server retrieves data from various sources, processes it, and sends the aggregated content to the client's web browser for display.
Advantages:

- Data security and control: Sensitive data can be processed and controlled on the server, reducing exposure to potential securityrisks.
- Centralized management: Data aggregation and processing logic canbe maintained centrally, making it easier to update and manage.

Disadvantages:

- Increased server load: Server-side processing can lead to higher server resource utilization, potentially impacting scalability andperformance.
- Limited interactivity: Server-side mashups may offer less interactivity, as the client primarily receives pre-processed data.

Example: An e-commerce website that fetches product information, pricing, and reviews from various third-party sources, processes it on theserver, and then sends a single, unified product page to the user's browser.

Client-Side Mashups:

Client-side mashups involve data integration and processing within theuser's web browser. The client (browser) retrieves

data from multiple sources and combines them, creating an integrated view or interactionfor the user.

Advantages:

- Reduced server load: Client-side mashups offload data processingfrom the server, potentially improving server scalability and reducing server-side resource usage.
- Enhanced interactivity: Client-side mashups often provide a more interactive and responsive user experience as data processing occursin real-time within the user's browser.

Disadvantages:

- Security considerations: Client-side processing may expose data from different sources to potential security risks, such as cross-sitescripting (XSS) attacks.
- Limited control: Control over data integration and presentation maybe dispersed across different client devices, making updates and maintenance less centralized.

```
"use server"
import axios from 'axios';

export async function getWeatherData(latitude:number, longitute :
    number) {const options = {
        method: 'GET',
        url: 'https://weatherapi-com.p.rapidapi.com/current.json',
```

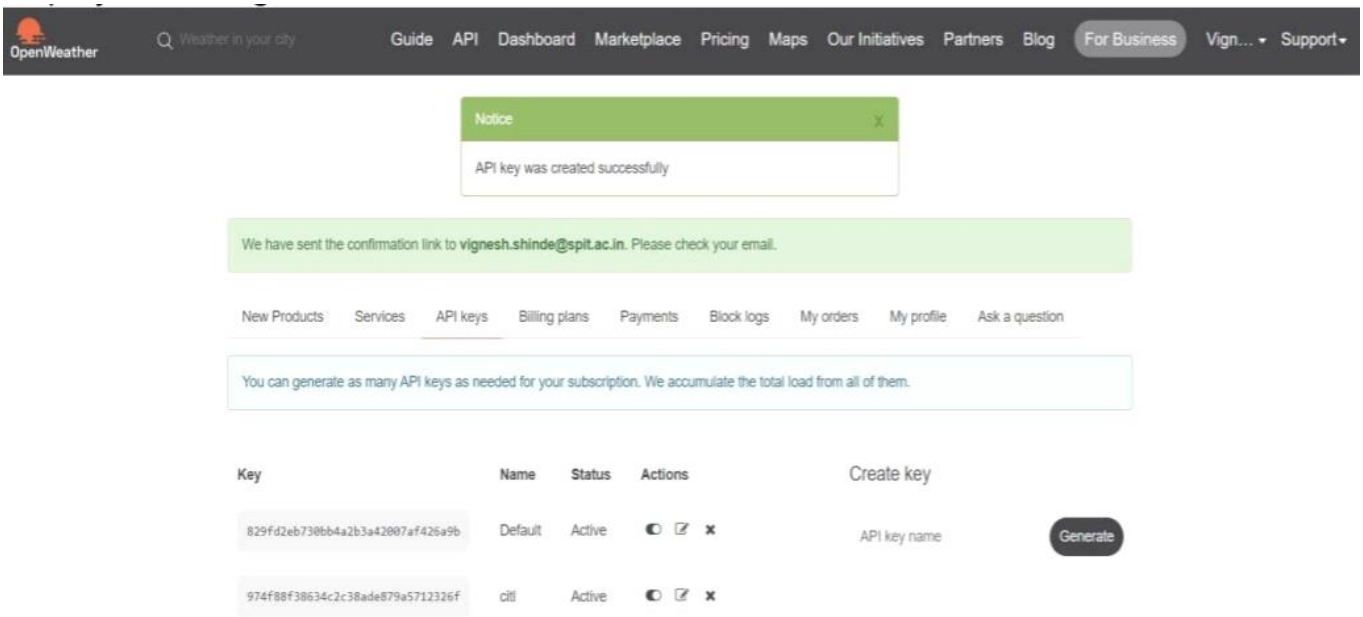
Example: A news aggregator web app that retrieves articles from multiple news sources via client-side JavaScript, combines them, anddisplays the integrated content in real-time for the user.

```
        params: {q:
        `${latitude},${longitute}`},headers:
        {
          'X-RapidAPI-Key':
'43de42a3d6msh884781a135634b1p17c5efjsna54bd644d065',
          'X-RapidAPI-Host': 'weatherapi-com.p.rapidapi.com'
        }
      };

    const response = await
    axios.request(options);return
    response.data;
}
async function
    getNewsData() {
    const options =
    {
        method: "GET",
        url: "https://bing-news-
        search1.p.rapidapi.com/news/search",params: {
          q:"news
          india",
          cc:
          "in",
          freshne
          ss:
          "Day",
          textFor
          mat:
          "Raw",
          safeSea
          rch:
          "Off",
        },
        headers: {
```

```
            "X-BingApis-
            SDK": "true",
            "X-RapidAPI-
            Key":
"43de42a3d6msh884781a135634b1p17c5efjsna54bd644d065",
            "X-RapidAPI-Host": "bing-news-search1.p.rapidapi.com",
          },
        };
    const response = await
    axios.request(options);return
    response.data;
}

export async function mashup(){
    const news = await getNewsData();
    const weather = await getWeatherData(19.1364,
    72.8296);return {news, weather};
}
```



References:

https://www.geeksforgeeks.org/what-is-a-mashup-in-web-technology/https://www.lifewire.com/what-is-a-mashup-3486655

Conclusion:

By performing this experiment, I understood the concept of web mashups in web development. I understood what It is, its benefits andhow to do it server side using node js