Name	Manish Shashikant Jadhav
UID	2023301005
Subject	CITL (Cloud and Internet Technology Lab)
Experiment No.	10
Aim	Create a web mashup of web services using open source framework
Theory	A **web mashup** is a web application that combines content, data, or functionality from multiple online sources into a single, integrated experience. The goal is to provide users with a new service or feature by blending various elements that weren't originally designed to work together. For example, a mashup could combine a map with weather data, showing users a location on the map with the current weather conditions. Another example could be combining restaurant reviews from multiple sites into a single interface for easy comparison. Mashups typically rely on APIs (Application Programming Interfaces) to fetch data from different services and display it in a unified way on a web page. This allows developers to create more dynamic, feature-rich applications without having to build everything from scratch. Implementing **Stripe** as an external API in a web mashup involves integrating Stripe's payment processing capabilities into your existing web application, which already pulls data or functionality from multiple sources. This enables your users to make payments, manage subscriptions, and interact with financial data while utilizing other services that the mashup integrates.
Implementation	We have External api which is stripe.

```ts
src > app > api > stripe > route.ts > RequestData
     You, last month | 1 author (You)
 1   import Stripe from "stripe";
 2
 3   import { NextResponse } from 'next/server';
 4   import { getServerSession } from "next-auth";
 5   import { authOptions } from "@/libs/auth";
 6   import { getRoom } from "@/libs/apis";
 7
 8   const stripe = new Stripe(process.env.STRIPE_SECRET_KEY as string, {
 9     apiVersion: '2024-09-30.acacia',
10   });
11
12   type RequestData = {
13     checkinDate: string;
14     checkoutDate: string;
15     adults: number;
16     children: number;
17     numberOfDays: number;          You, last month • Completed Stripe Integration
18     hotelRoomSlug: string;
19   };
20
21   export async function POST(req: Request) {
22     const {
23       checkinDate,
24       adults,
25       checkoutDate,
26       children,
27       hotelRoomSlug,
28       numberOfDays,
29     }: RequestData = await req.json();
30
31     if (
32       !checkinDate ||
33       !checkoutDate ||
34       !adults ||
35       !hotelRoomSlug ||
36       !numberOfDays
37     ) {
38       return new NextResponse('Please all fields are required', { status: 400 });
39     }
```

⊗ [Ycm] Invalid ycm path

New business

🔍 Search   Test mode ⬤

Home
Balances
Transactions
Customers
Product catalog

Activate payments on your account
Fill out your business profile to accept payments. Any progress you make will be
saved, so you can always finish later.

Activate payments →

Products
Payments
Billing
Reporting
More

Get started with Stripe   ⊗

                                    ⬚⬚ Explore all products   >

                                    🖥 For developers   ⓘ Test mode
                                    Publishable key   pk_test_51Q8n3f2KB7FnPj...
                                    Secret key   •••••••••• 👁
                                    View docs →

Today

Developers

Conclusion	Hence by completing this experiment I got to know how to Use docker to run a multi-container web application