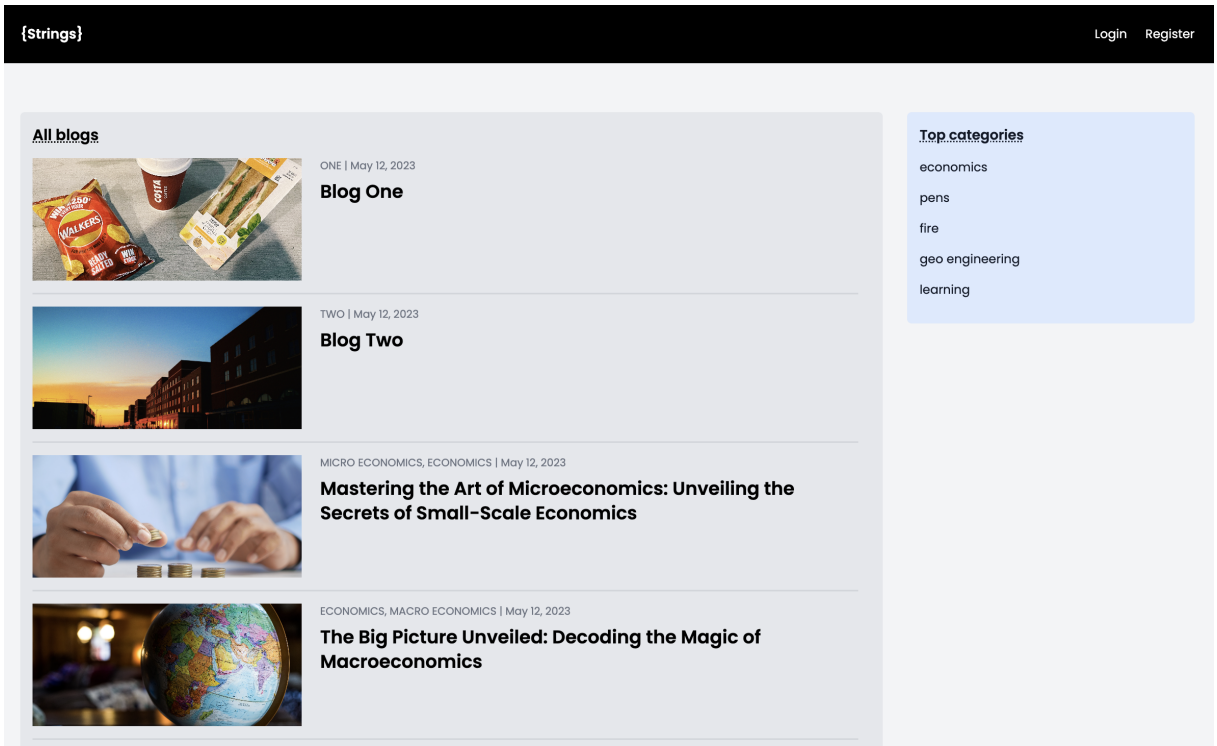
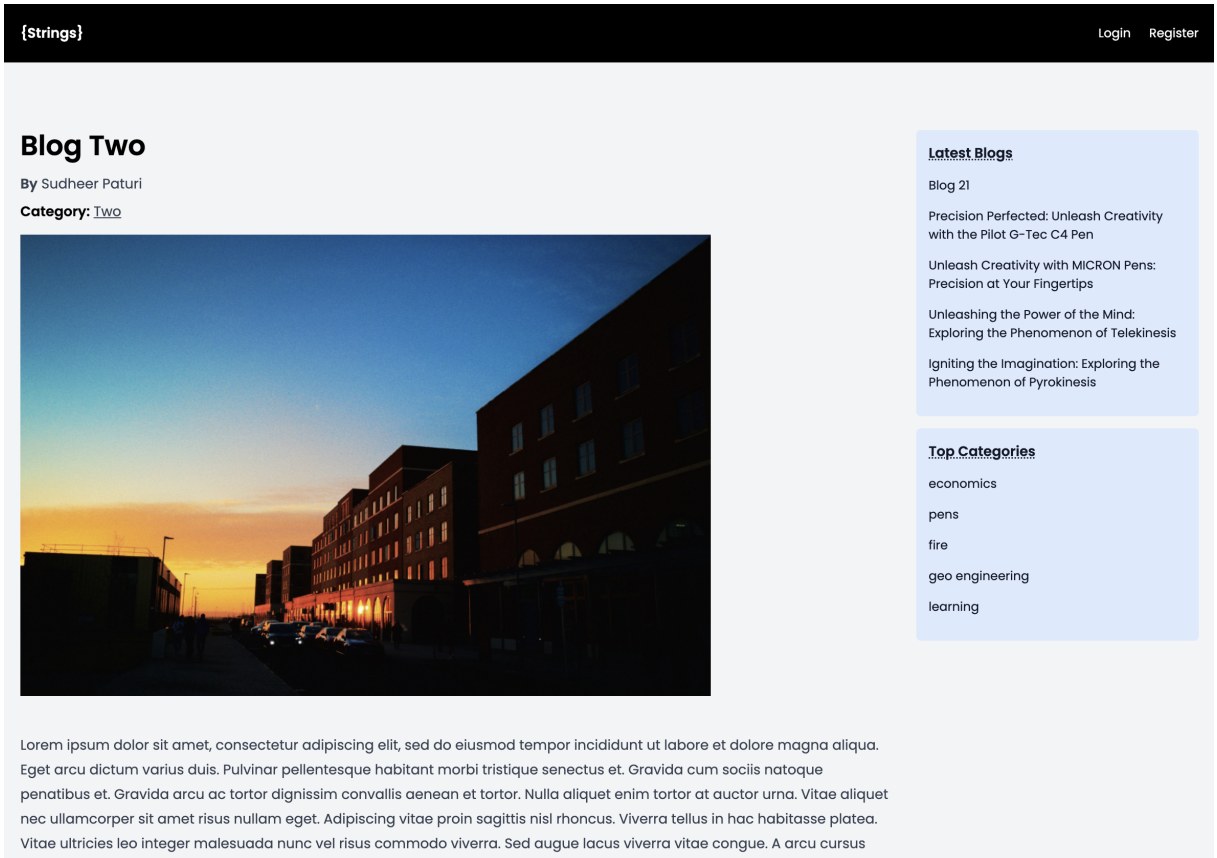
2232843 - Coursework 3b

1. **Give an example along with screenshots of some code you have implemented yourself.**




```
@extends('layout')

@section('content')
<x-app-layout>
    <div class="container mx-auto py-2 flex flex-wrap md:flex-nowrap">
        <!-- Blog List -->
        <div class="w-full md:w-3/4 md:pr-8 bg-gray-200 mr-8 p-4 rounded-md">
            <h1 class="text-xl font-bold mb-4 underline decoration-dotted">All blogs</h1>

            @if ($blogs->count() > 0)
                @foreach ($blogs as $blog)
                    <div class="mb-4 flex flex-wrap md:flex-nowrap pb-4 border-solid border-b-2 border-gray-300">
                        <img src="/images/{{ $blog->featured_image }}" alt="{{ $blog->title }}" class="w-full md:w-1/3 h-auto md:h-40 o
                        <div class="w-full md:w-2/3">
                            <p class="text-sm text-gray-500 mb-2">
                                @foreach($blog->categories as $category)
                                    <a href="{{ route('categories.show', $category) }}" class="hover:underline uppercase"> {{ $category
                                @endforeach
                                | {{ $blog->created_at->format('M d, Y') }}
                            </p>
                            <h2 class="text-2xl font-bold hover:underline mb-2"><a href="{{ route('blogs.show', ['slug' => $blog->slug]
                        </div>
                    </div>
                @endforeach
                <!-- Render the pagination links -->
                {{ $blogs->links() }}
            @else
                <p>No blogs found.</p>
            @endif
        </div>
```

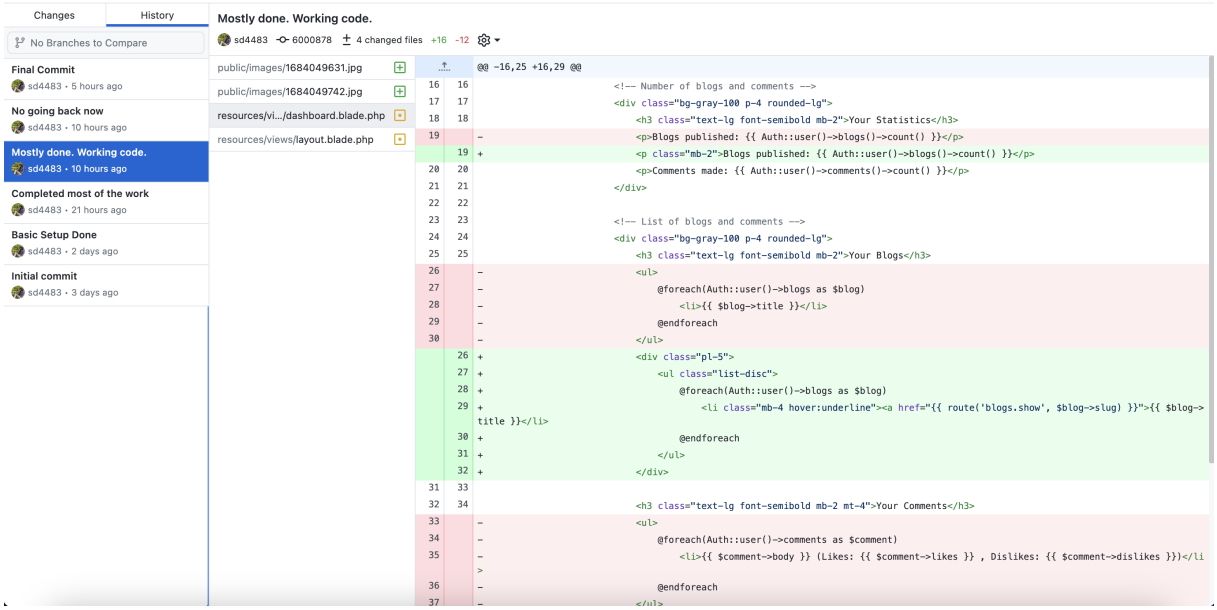
```
        <!-- Sidebar -->
        <div class="w-full md:w-1/4 mt-8 md:mt-0">
            <div class="bg-blue-100 p-4 rounded-md">
                <h2 class="text-lg font-bold text-gray-900 mb-4 underline decoration-dotted">Top categories</h2>
                <ul>
                    @foreach($topCategories as $category)
                        <li class="mb-4"><a href="{{ route('categories.show', $category) }}" class="text-gray-900">{{ $category->name }
                    @endforeach
                </ul>
            </div>
        </div>
    </div>
</x-app-layout>
@endsection
```




```
@extends('layout')

@section('content')
<x-app-layout>
    <div class="container mx-auto py-8 flex">
        <div class="w-3/4 pr-4">
            <h1 class="text-4xl font-bold mb-4">{{ $blog->title }}</h1>
            <p class="text-lg mb-2 text-gray-700"><strong>By</strong> {{ $blog->user->name }}</p>

            <p class="mb-4 text-lg">
                <strong>Category:</strong>
                @foreach($blog->categories as $category)
                    <a href="{{ route('categories.show', $category) }}" class="text-gray-700 underline">{{ $category->name }}</a>{{ !$l
                @endforeach
            </p>
```

```
            @if ($blog->featured_image)
                <img src="/images/{{ $blog->featured_image }}" alt="{{ $blog->title }}" class="mb-4">
            @endif

            <div class="blog-content prose prose-lg w-full max-w-none mb-12 mt-12">
                {!! $blog->content !!}
            </div>

            <!-- Comments -->
            <div class="comments-content">
                @include('comments.show', ['blog' => $blog])
            </div>
        </div>
        <div class="w-1/4 pl-4">
            <div class="bg-blue-100 p-4 rounded-md mb-4">
                <h2 class="text-lg font-bold text-gray-900 mb-4 underline decoration-dotted">Latest Blogs </h2>
                <ul>
                    @foreach($latestBlogs as $latestBlog)
                        <li class="mb-4"><a href="{{ route('blogs.show', $latestBlog->slug) }}" class="text-gray-900">{{ $latestBlog->t
                    @endforeach
                </ul>
            </div>

            <div class="bg-blue-100 p-4 rounded-md">
                <h2 class="text-lg font-bold text-gray-900 mb-4 underline decoration-dotted">Top Categories</h2>
                <ul>
                    @foreach($topCategories as $category)
                        <li class="mb-4"><a href="{{ route('categories.show', $category) }}" class="text-gray-900">{{ $category->name }
                    @endforeach
                </ul>
            </div>
        </div>
    </div>
</x-app-layout>
@endsection
```

2. **Provide a screenshot of your git commit history here**



3. **Explain the concept of MVC referencing source files in your implementation by providing screenshots**

   MVC stands for Model-View-Controller. It is a design pattern that separates an application into three interconnected components. This separation helps manage complexity when building an application and improves flexibility and control.

Model: This corresponds to all data-related logic. Models interact with the database, retrieve data, insert data, update data, and delete data. They represent the knowledge and the data of an application.

In this case, the Blog, Comment, Category, and User models are all interacting with their respective tables in the database.

View: This corresponds to the user interface logic of the application. It's responsible for presenting data to the user in a readable and understandable format.

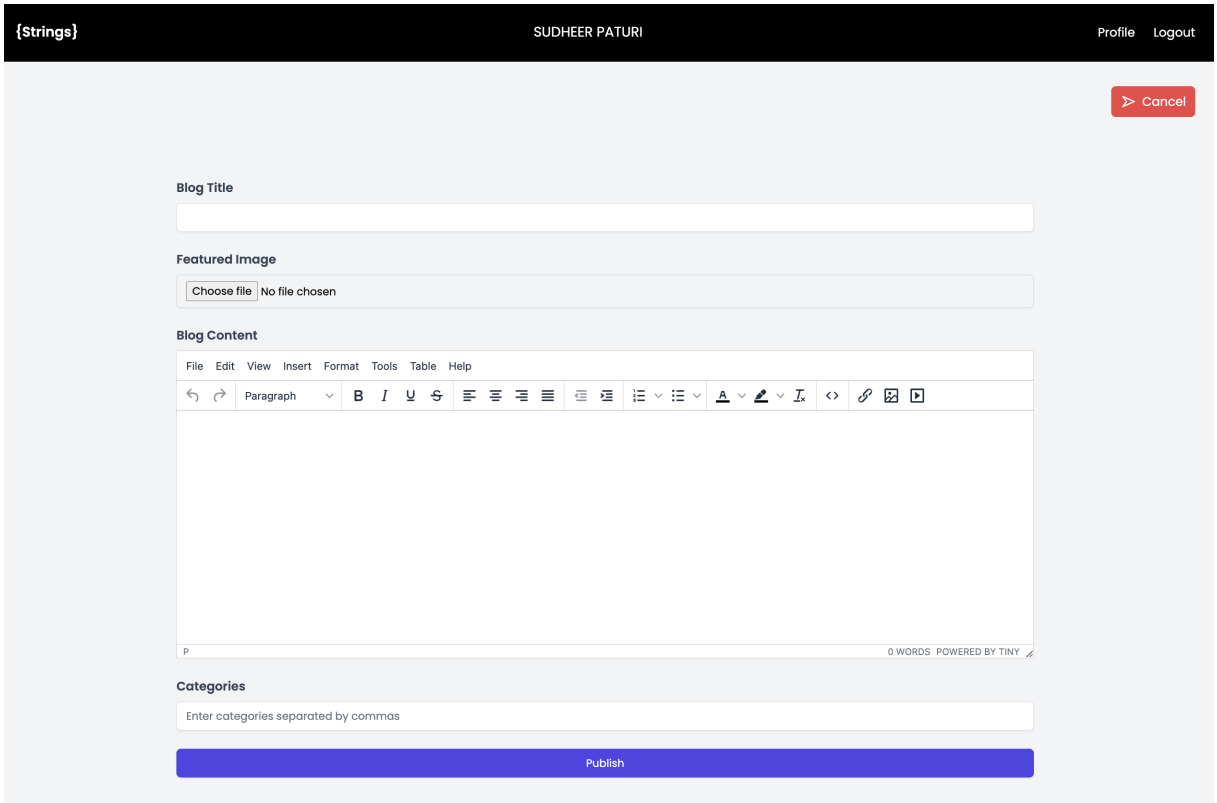
In this case, the layout view, welcome view, dashboard view, categories.show view, blogs.show view, and comments.show view are all various interfaces the user interacts with. They receive data from the controllers (which get their data from the models) and render it for the user.

Controller: This corresponds to the business logic of your application. Controllers handle user requests, interact with the model to get the data, and pass this data to the views for presentation.

In this case, the BlogController, CommentController, CategoryController, and UserController are all controllers. They handle different types of user requests, such as creating a new blog post, adding a comment, showing a category, and managing user profiles.

Explanation using Blog views, Blog Controller and Blog Model in a typical flow:

The user interacts with the view (e.g., submits a form to create a new blog post)



```
@extends('layout')

@section('content')
<x-app-layout>
    <div class="container mx-auto px-6 py-8">
        <div class="w-full md:w-3/4 mx-auto">

            @if ($errors->any())
                <div class="alert alert-danger">
                    <ul>
                        @foreach ($errors->all() as $error)
                            <li>{{ $error }}</li>
                        @endforeach
                    </ul>
                </div>
            @endif


            <form action="{{ route('blogs.store') }}" method="post" enctype="multipart/form-data" class="space-y-6">
                @csrf

                <div>
                    <label for="title" class="block text-base font-bold text-gray-700 mb-2">Blog Title</label>
                    <input type="text" id="title" name="title" class="mt-1 block w-full py-2 px-3 border border-gray-300 rounded-md sha
                </div>

                <div>
                    <label for="featured_image" class="block text-base font-bold text-gray-700 mb-2">Featured Image</label>
                    <input type="file" id="image" name="image" class="mt-1 block w-full py-2 px-3 border border-gray-300 rounded-md sha
                </div>

                <div>
                    <label for="content" class="block text-base font-bold text-gray-700 mb-2">Blog Content</label>
                    <textarea id="content" name="content" class="mt-1 block w-full py-2 px-3 border border-gray-300 rounded-md shadow-s
                    <script>
                        tinymce.init({
                          selector: '#content',
                          height: 400, // Specify the desired height for the editor
                          plugins: 'advlist autolink lists link image charmap print preview anchor searchreplace visualblocks code full
                          toolbar: 'undo redo | formatselect | bold italic underline strikethrough | alignleft aligncenter alignright a
                          menubar: 'file edit view insert format tools table help', // Customize the menubar with desired options
                            // Optionally, link a custom CSS file for further styling
                        });
                    </script>
                </div>

                <div>
                    <label for="categories" class="block text-base font-bold text-gray-700 mb-2">Categories</label>
                    <input type="text" id="categories" name="categories" class="mt-1 block w-full py-2 px-3 border border-gray-300 roun
                </div>

                <button type="submit" class="w-full flex justify-center py-2 px-4 border border-transparent rounded-md shadow-sm text-s
            </form>
        </div>

        @if(isset($blog) && !$blog->published)
            <a href="{{ route('welcome') }}" class="px-3 py-2 bg-red-600 text-white rounded-md flex items-center space-x-2">
                <svg class="h-5 w-5" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke="currentColor">
                    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M19 7l-.867 12.142A2 2 0 0116.138 21H7.862
                </svg>
                <span>Delete Draft</span>
            </a>
        @endif

    </div>

</x-app-layout>
@endsection
```

The view sends this request to a controller (e.g., the BlogController)

```
<?php

namespace App\Http\Controllers;
```

```
use Illuminate\Support\Facades\Auth;
use Illuminate\Http\Request;
use Illuminate\Support\Str;
use App\Models\Blog;
use App\Models\Category;
use App\Models\Comment;

class BlogController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $blogs = Blog::where('published', true)->paginate(9);
        return view('welcome', compact('blogs'));
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        return view('create');
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'title' => 'required',
            'content' => 'required',
            'categories' => 'required',
            'featured_image' => 'nullable|image',
        ]);

        $blog = new Blog();
        $blog->title = $validatedData['title'];
        $blog->slug = Str::slug($validatedData['title']); // Generate slug from title
        $blog->content = $validatedData['content'];

        // Handling the image upload
        if ($request->hasFile('image')) {
            $imageName = time().'.'.$request->image->extension();
            $request->image->move(public_path('images'), $imageName);
            $blog->featured_image = $imageName;
        }

        $blog->status = $request->filled('published') ? 'published' : 'draft'; // Check if "published" field is filled
        $blog->user_id = Auth::id();

        // Save the blog to the database
        $blog->save();

        // Split the categories string by commas
        $categoryNames = explode(',', $request->input('categories'));

        // Loop through each category name
        foreach ($categoryNames as $categoryName) {
            // Trim whitespace
            $categoryName = strtolower(trim($categoryName));

            // Find the category by its name or create a new one if it does not exist
            $category = Category::firstOrCreate(['name' => $categoryName]);

            // Attach the category to the blog
            $blog->categories()->attach($category);
        }

        return redirect()->route('welcome');
    }

    /**
     * Display the specified resource.
     */


     public function show(string $slug)
     {
```

```
        $blog = Blog::where('slug', $slug)->with(['comments' => function ($query) {
            $query->orderBy('created_at', 'desc');
        }])->firstOrFail();

        $latestBlogs = Blog::latest()->limit(5)->get();
        $topCategories = Category::getTopCategories(5);

        return view('blogs.show', [
            'blog' => $blog,
            'latestBlogs' => $latestBlogs,
            'topCategories' => $topCategories,
        ]);
    }


    /**
     * Show the form for editing the specified resource.
     */
    public function edit(string $id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     */
    public function update(Request $request, string $id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy(string $id)
    {
        //
    }

}
```

The controller interacts with the model (e.g., the Blog model) to process data (e.g., save the new blog post in the database)

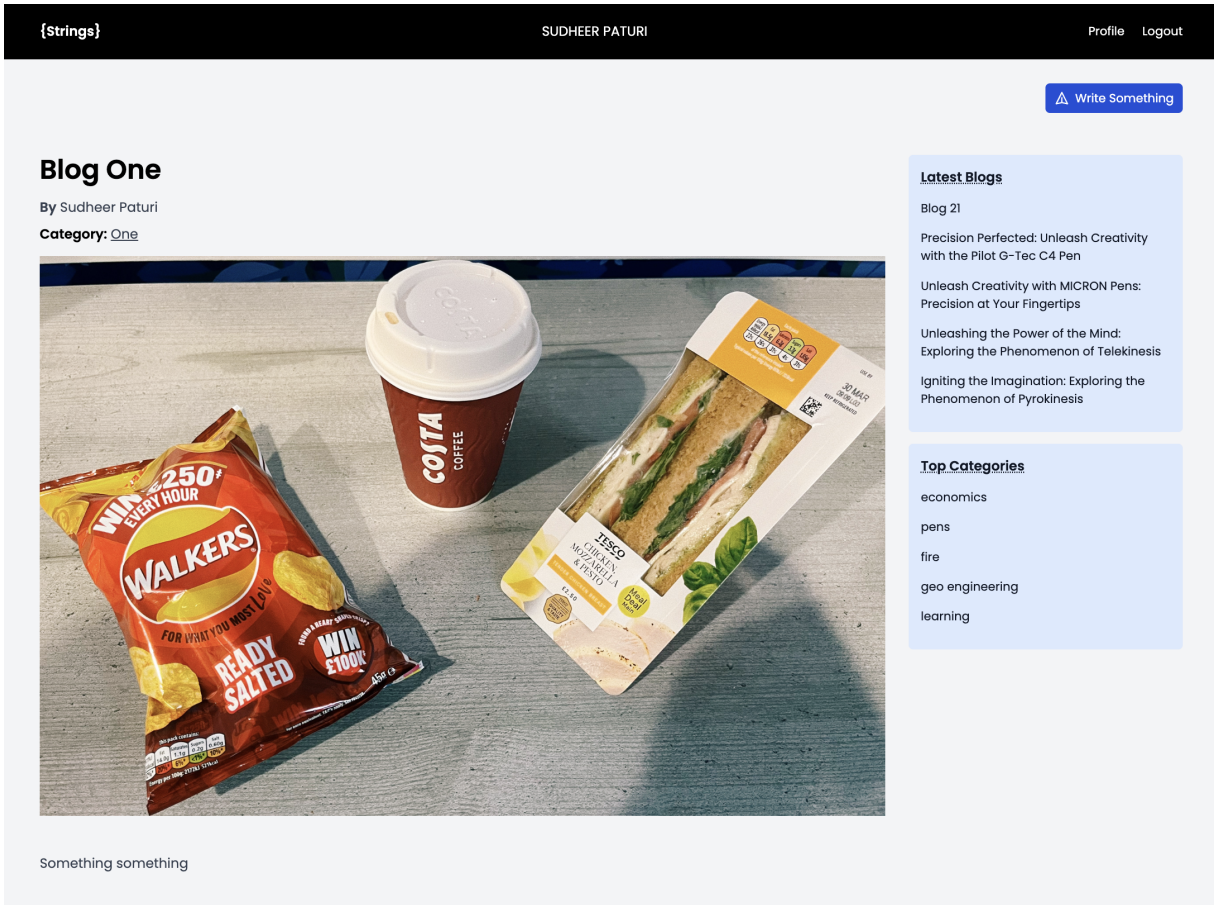
```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Support\Str;
use Illuminate\Database\Eloquent\Model;


class Blog extends Model
{
    use HasFactory;

    public function user()
    {
        return $this->belongsTo(User::class);
    }

    public function categories()
    {
        return $this->belongsToMany(Category::class);
    }

    public function comments()
    {
        return $this->hasMany(Comment::class);
    }



    public static function boot()
    {
        parent::boot();
```

```
        static::creating(function ($blog) {
            $blog->slug = Str::slug($blog->title);
        });

        static::updating(function ($blog) {
            $blog->slug = Str::slug($blog->title);
        });
    }
}
```

The controller then receives data from the model (e.g., the newly created blog post) and sends it to the blogs.show view

```
@extends('layout')

@section('content')
<x-app-layout>
    <div class="container mx-auto py-8 flex">
        <div class="w-3/4 pr-4">
            <h1 class="text-4xl font-bold mb-4">{{ $blog->title }}</h1>
            <p class="text-lg mb-2 text-gray-700"><strong>By</strong> {{ $blog->user->name }}</p>

            <p class="mb-4 text-lg">
                <strong>Category:</strong>
                @foreach($blog->categories as $category)
                    <a href="{{ route('categories.show', $category) }}" class="text-gray-700 underline">{{ $category->name }}</a>{{ !$l
                @endforeach
            </p>

            @if ($blog->featured_image)
                <img src="/images/{{ $blog->featured_image }}" alt="{{ $blog->title }}" class="mb-4">
            @endif

            <div class="blog-content prose prose-lg w-full max-w-none mb-12 mt-12">
                {!! $blog->content !!}
            </div>

            <!-- Comments -->
            <div class="comments-content">
                @include('comments.show', ['blog' => $blog])
            </div>
        </div>
        <div class="w-1/4 pl-4">
            <div class="bg-blue-100 p-4 rounded-md mb-4">
                <h2 class="text-lg font-bold text-gray-900 mb-4 underline decoration-dotted">Latest Blogs </h2>
                <ul>
                    @foreach($latestBlogs as $latestBlog)
                        <li class="mb-4"><a href="{{ route('blogs.show', $latestBlog->slug) }}" class="text-gray-900">{{ $latestBlog->t
                    @endforeach
                </ul>
            </div>

            <div class="bg-blue-100 p-4 rounded-md">
                <h2 class="text-lg font-bold text-gray-900 mb-4 underline decoration-dotted">Top Categories</h2>
                <ul>
                    @foreach($topCategories as $category)
                        <li class="mb-4"><a href="{{ route('categories.show', $category) }}" class="text-gray-900">{{ $category->name }
                    @endforeach
                </ul>
            </div>
        </div>
    </div>
</x-app-layout>
@endsection
```

The blogs.show view then presents this data to the user



This separation of objects allows developers to work on individual components without affecting others, improving maintainability and scalability.

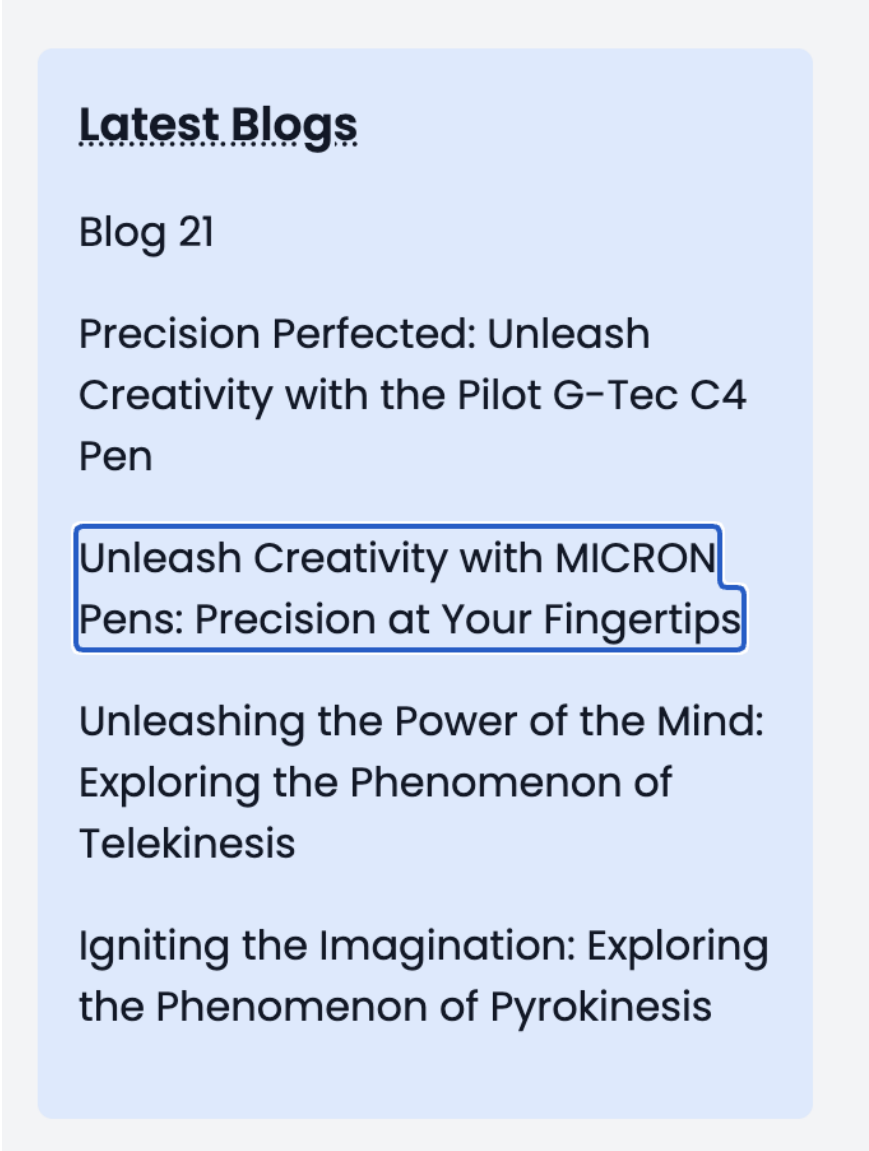
4. **Write a reflection on how well your web application meets the Web Content Accessibility Guidelines 2 requirements, include screenshots**

   My application meets some of the requirements mentioned in the Web Content Accessibility Guidelines 2. Here's an explanation.

   Text Alternatives: I'm providing alt attributes for images, which means screen readers will be able to interpret the images and provide a description for users who can't see them.

```
@if ($blog->featured_image)
     <img src="/images/{{ $blog->featured_image }}" alt="{{ $blog->title }}" class="mb-4">
@endif
```

   Keyboard Accessibility: People can navigate some parts of the website using keyboard although not the entire site.



Understandable Text: All the text on the site is readable and understandable. I'm using a minimum font size of 16px so that none of the font appears small for any reader.

Welcome, curious minds, to a captivating journey into the fascinating world of microeconomics! Have you ever wondered how individual decisions shape the intricate fabric of our economy? Or how businesses determine prices and output levels? If so, you're in for a treat. In this blog post, we will unravel the mysteries of microeconomics, shedding light on its core principles and illustrating its profound impact on our daily lives. Get ready to embark on a captivating exploration that will forever change the way you perceive the economy!

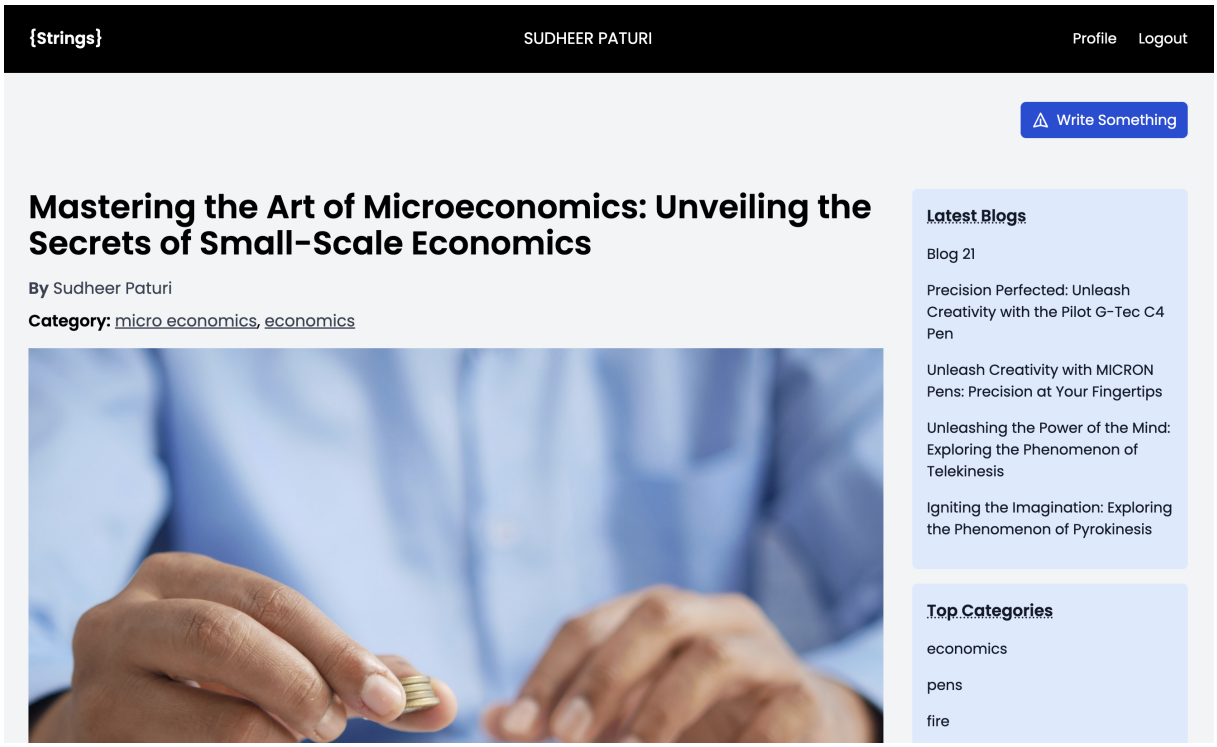
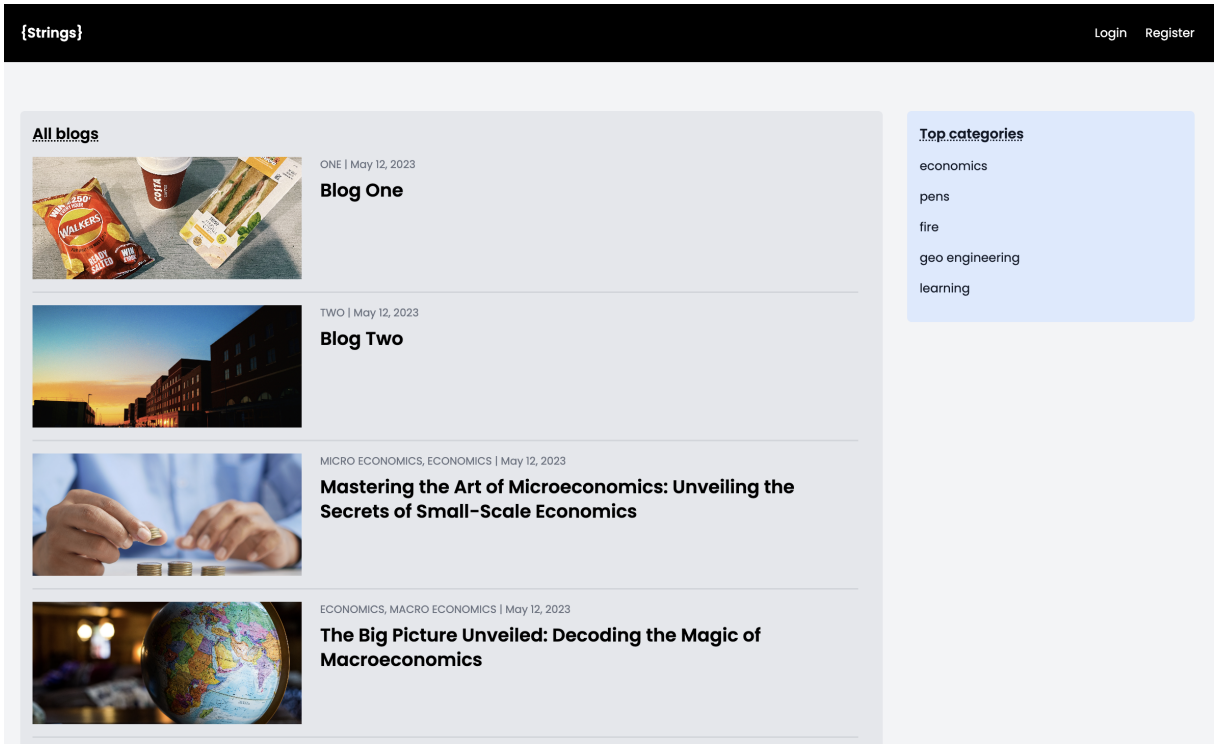
The Essence of Microeconomics

At its core, microeconomics delves into the study of economic behavior at the individual, household, and firm level. It analyzes how people make decisions regarding the allocation of limited resources to satisfy their needs and wants. From understanding consumer preferences to unraveling the intricate workings of supply and demand, microeconomics serves as a powerful tool for comprehending the building blocks of the economy.

Demand and Supply: The Forces of the Marketplace

No discussion of microeconomics would be complete without examining the dynamics of demand and supply. Picture a bustling marketplace where buyers and sellers interact, shaping prices and determining the quantity of goods and services exchanged. Microeconomics unveils the delicate equilibrium between consumer desires and producer capabilities, revealing how these forces drive the allocation of resources and ultimately influence market outcomes.

Distinguishable Content: I’m using contrasting colors and different font sizes to make the content as distinguishable as possible.



Predictability: Everything on the site is predictable I believe, there are clear headings everywhere and easy to follow navigation across the site.