



a blogging platform about photography

## Development Artifacts

### **REQUIREMENTS TO RELEASE: 1.0**

This document describes my basic approach, and techniques towards a web project. The aim is to have a good understanding of the project and to close any communication gap between myself and the customer.

### **Introduction**

This document provides a summary of the vision of the WebBlog project. The sections that follow provide an overview of the product, its stakeholders, its key features, and other requirements that form the foundation for the project.

#### **1. Background**

WebBlog is a fictitious company created by Foluso solely for Square1 assignment. This assignment is aimed towards understanding my approach towards a Web development project. Everything in this project is fictitious and are created from ground up, specifically for this project and not to be construed as representative of actual or existing project in the market.

## **2. Case Study**

This case study involves WebBlog, a fictitious company who is into web blogging. The WebBlog has approached Square1 on the need to develop a web blogging platform where users can create posts for viewing by visitors. WebBlog also emphasizes the need for the platform to interact with and retrieve new posts from another blogging platform through a provided api endpoint.

Any user on WebBlog can register in the platform, login to a private area to see the posts he created and, if they want, create new ones. However, they won't be able to edit or delete them.

Also, WebBlog have another blogging platform and want Square1 to create a way to auto import posts created on this other platform to the new blogging platform.

The expectation is that the website will be receiving quite a lot of traffic and one major goal is to minimize the strain put on the system during traffic peaks while also minimizing the strain we put on the feed server.

As a Web Developer my goal is to design and develop the web blogging platform that will meet the customer requirements and that can also withstand sudden spike in traffic. The focus will be on flexibility, clean code & good source code management, automated workflow, and easy deployment.

## Sprint 1 – Project Planning

### 3. Problem and Vision Statements

#### Problem Statement

The problem of	Interests in two different niche and writing about them on the same blogging platform
affects	The quality of contents on an existing platform
the impact of which is	that the blogging platform have started developing two different voices which can be confusing to the audience.
a successful solution would be	a new blogging platform that will communicate to the audience a single niche so that the old platform can maintain its original niche.

#### Product Vision statement

For	WebBlog audience
Who	want niche specific information from blogging platform
The product WebBlog	Is a niche focus blogging platform
That	Offers insightful information about what's trending in photography
Unlike	Our old blogging platform
Our new product	Integrates with top brands to also deliver contents to our audience.

#### 4. Requirements Elicitation and Specifications

##### Stakeholder descriptions

Table 1: Stakeholder summary

Name	Description	Product responsibilities	Project responsibilities
Visitors	Guests or platform visitors who read blog post on the website	Read posts and provide feedbacks.	If some are selected, they will provide inputs through interviews, or surveys.
WebBlog User	Users of WebBlog or Employees at WebBlog	Write, review, and create posts	Provide test and feedbacks on product releases
WebBlog project team	Employees at Square1 working on the project	Ensure intuitive design, ensure product scalability, test all features implementation.	Deliver the software based on the requirements by WebBlog. Keep WebBlog stakeholders informed on the project progress, as needed
Project Sponsor	Founder and CEO of WebBlog	Provide long-term vision and goals. Help prioritize product features in alignment with short and long-term goals.	Participate in project meetings and provide feedbacks

## Product Overview

Table 2: Product needs and features

Need	Feature	Priority
Manage Users	Register as a user	2
	Login as a user	
	Update Profile	2
Manage Blog Post	Create Blog post	1
	List only created blog posts	1
	Show Blog Posts to Audience	1
	Sort Blog Posts by publication date	1
Integrates with external API	Retrieve new post from API	2
Manage Application Development and Deployment	Create GitHub repository for source code management	1
	Automate Infrastructure creation	5
	Create Deployment Pipeline with GitHub action	8

## Use Case Diagram

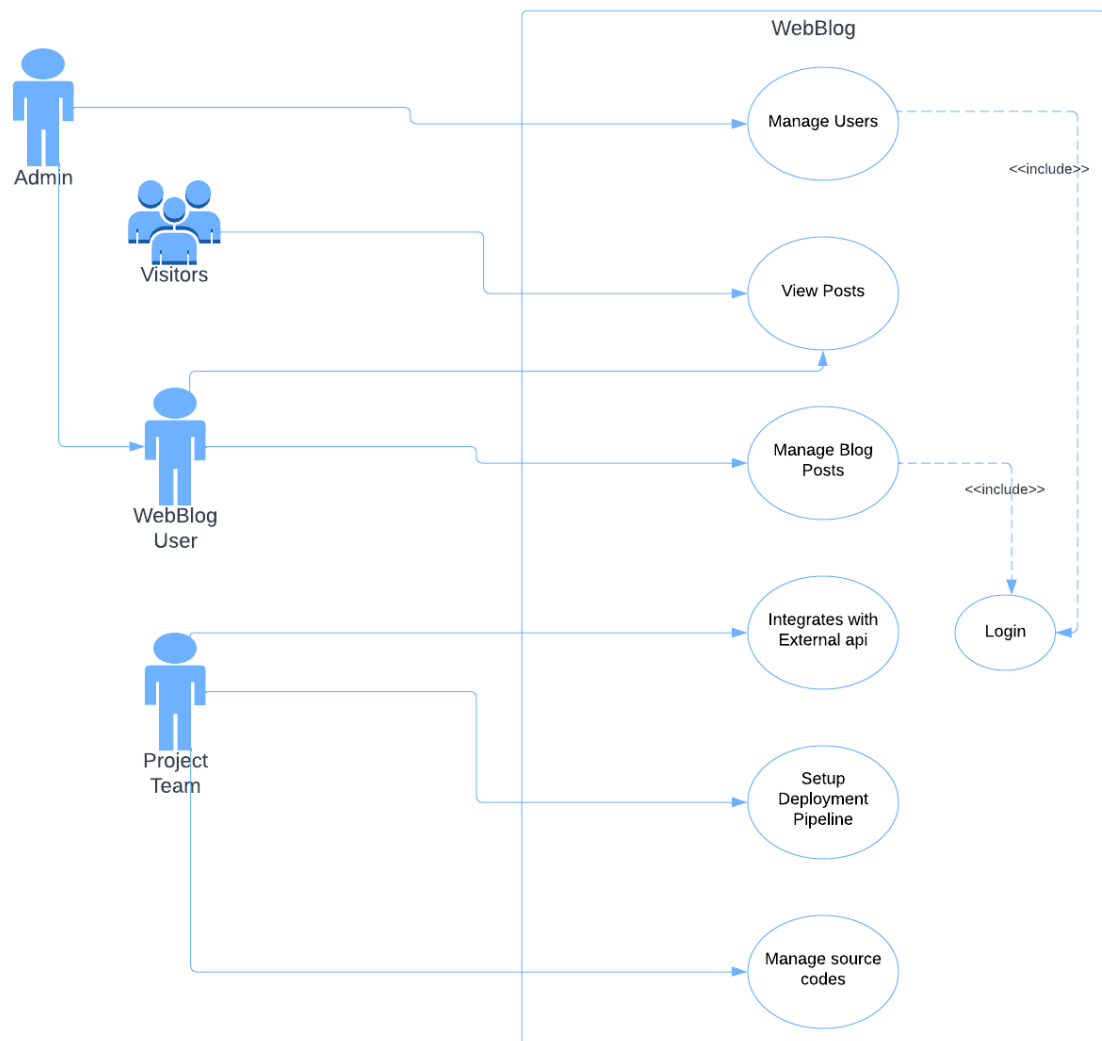


Figure 1: Use Case Diagram

## 5. Requirement Validation

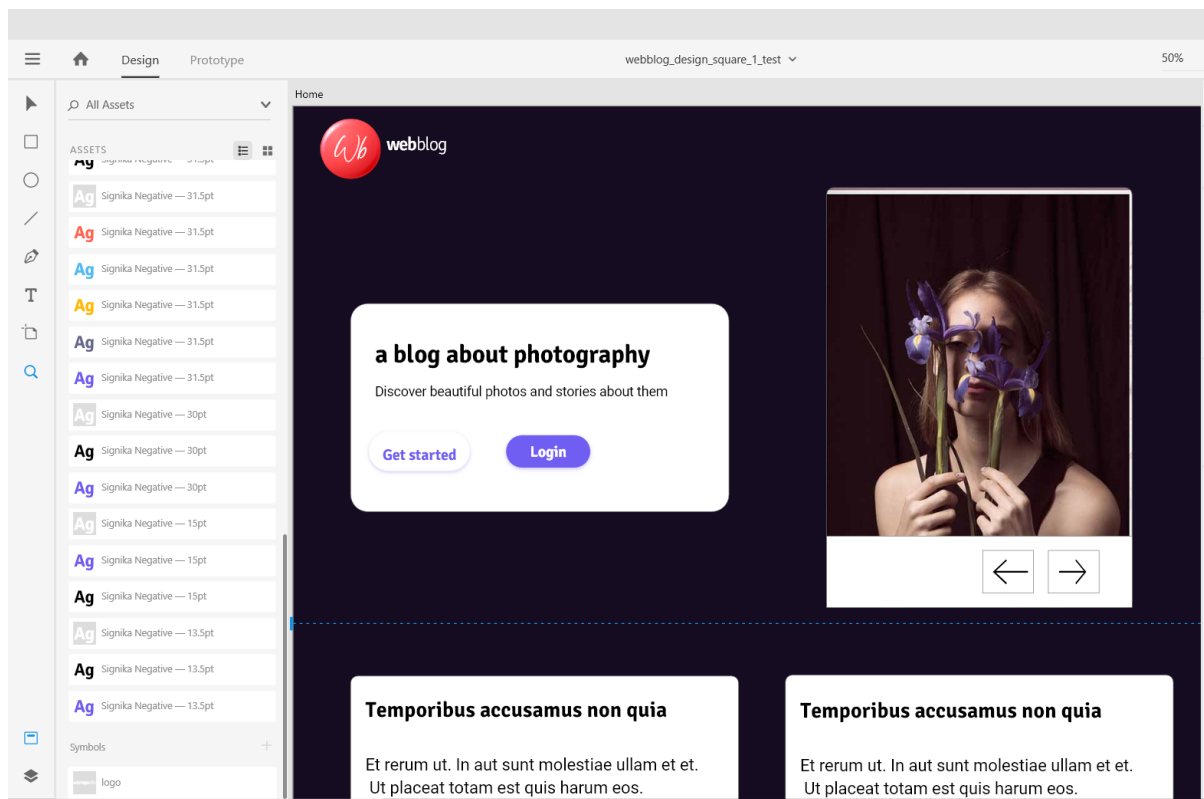


Figure 2: UI Mock up

## 6. Technology Validation

### Application Architecture

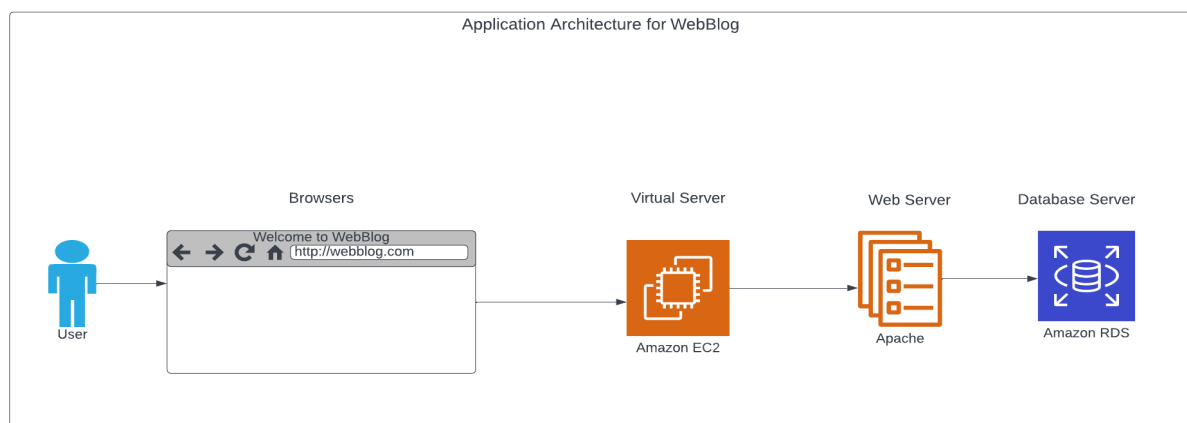


Figure 3: Application Layer

## Development Environment Architecture

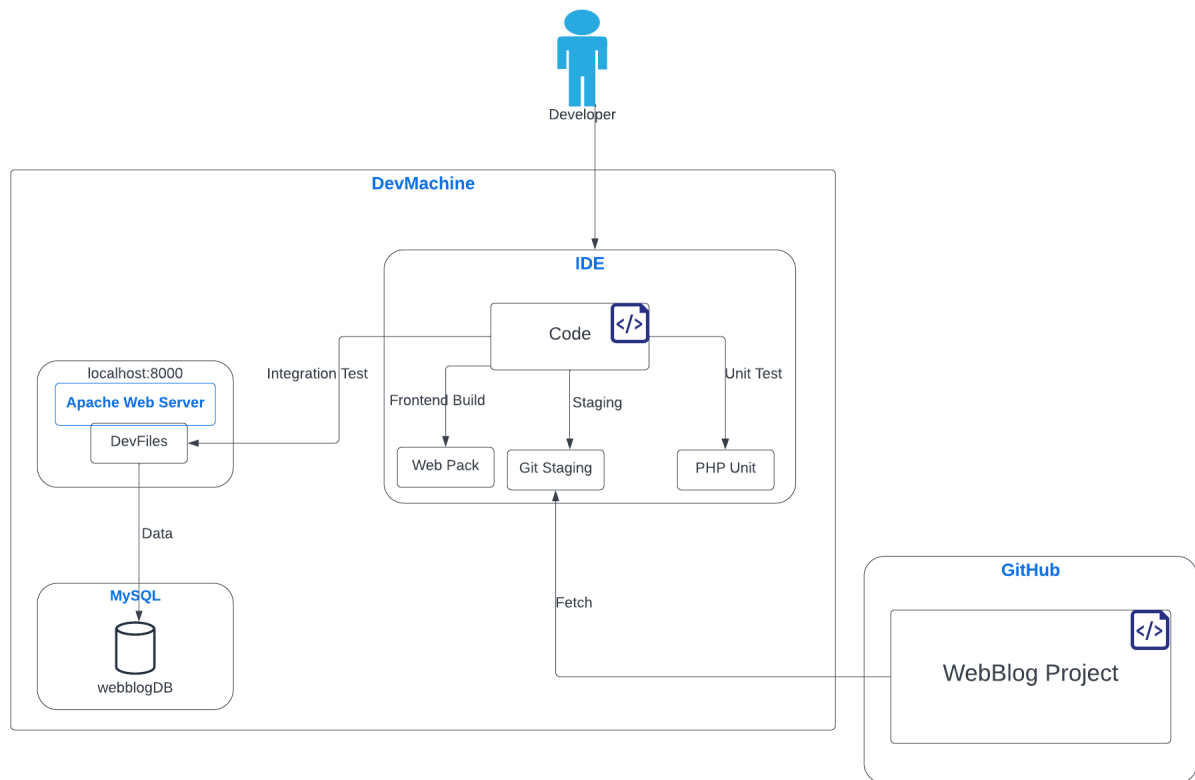


Figure 4: Development Environment Architecture

## Data Model

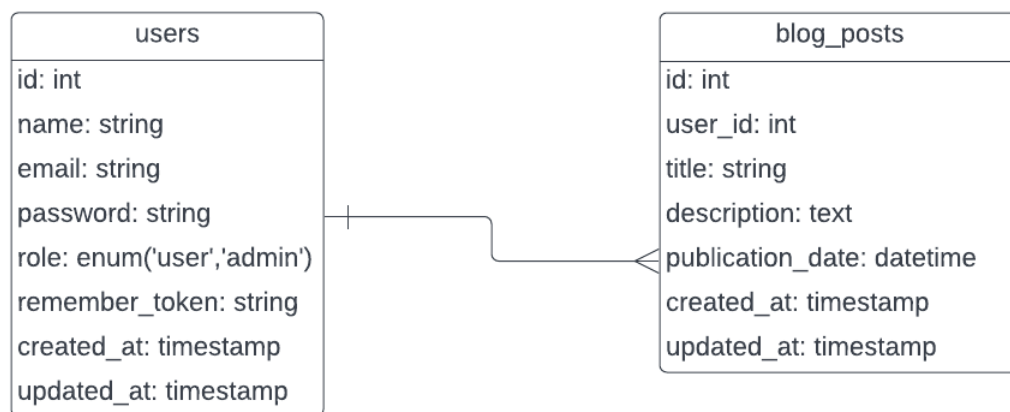
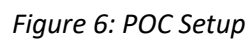


Figure 5: Data model



### Dev to Prod Deployment Diagram



## 8. Proof of Concept: Implementation

### Revised use case diagram

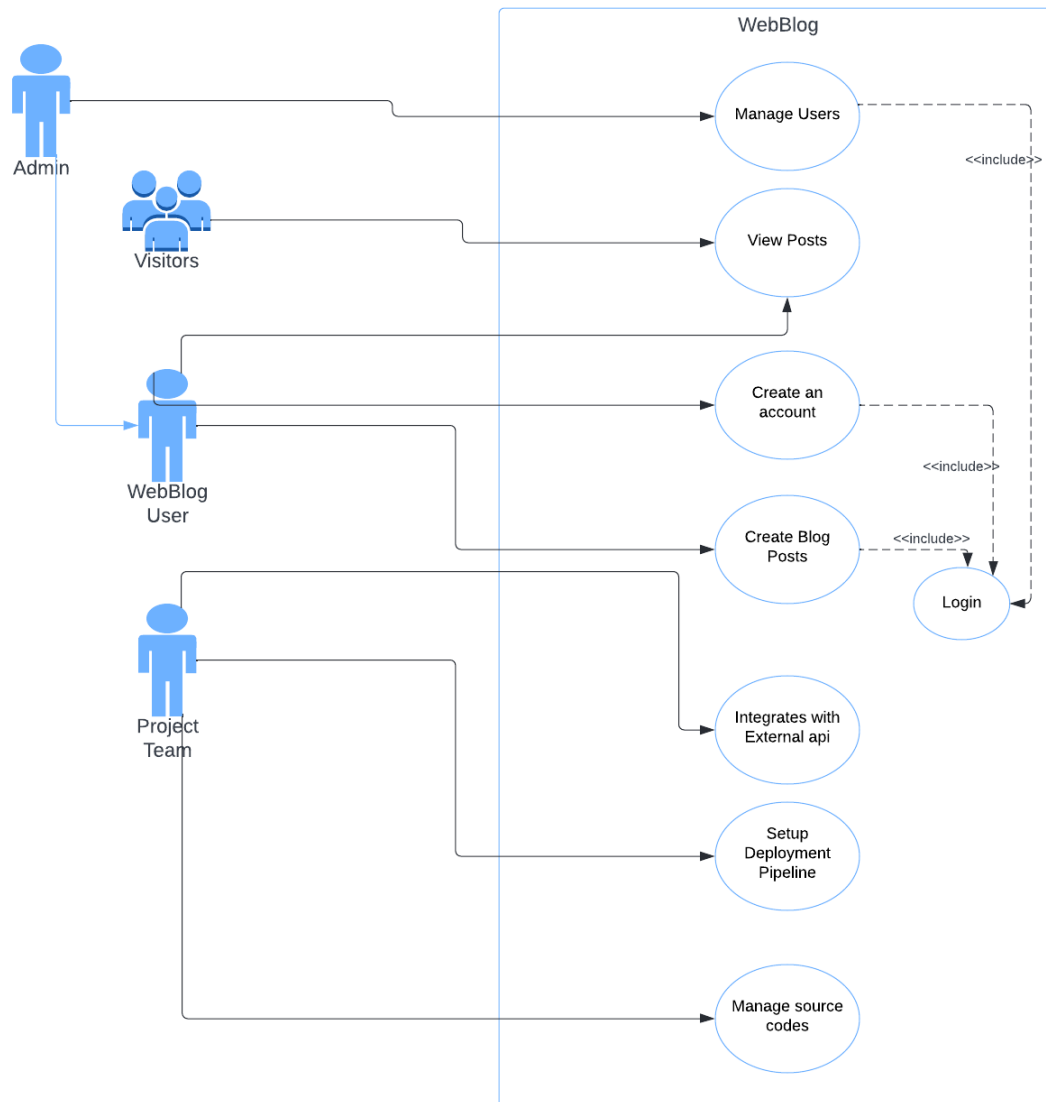


Figure 7: Revised use case diagram

## Sequence Diagram - POC

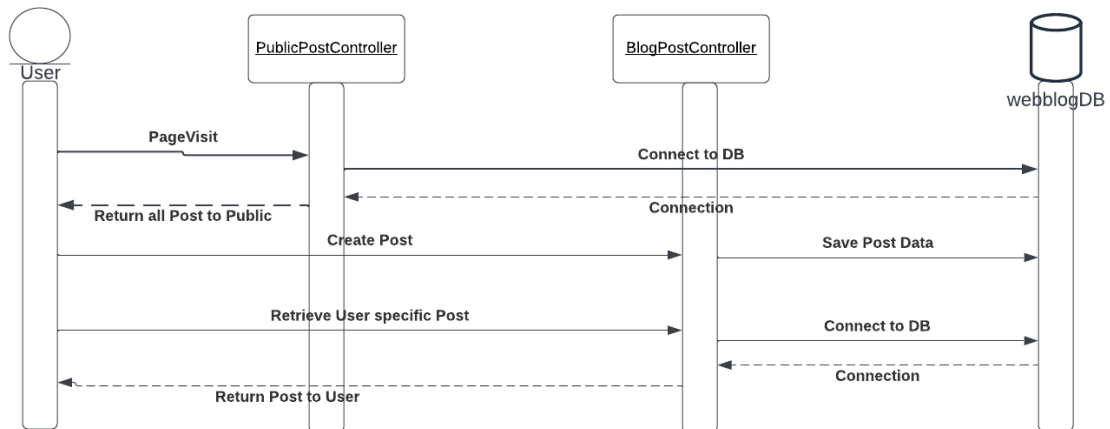
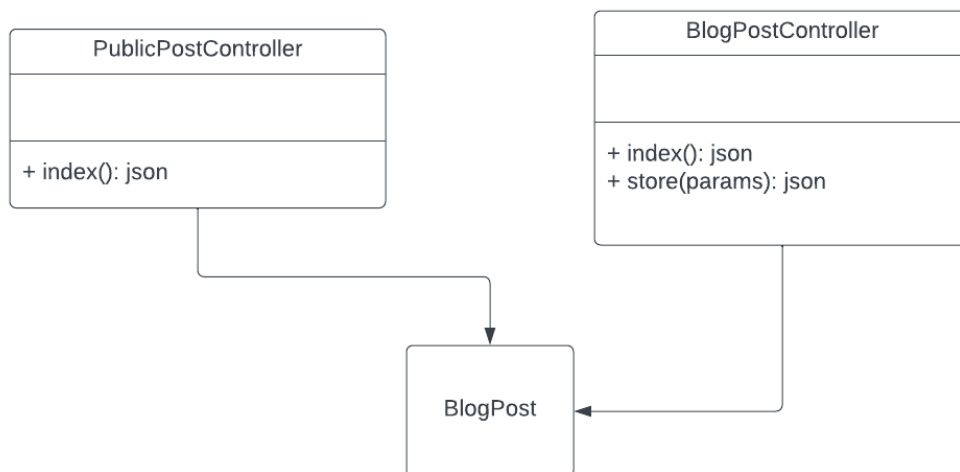


Figure 8: Sequence Diagram - POC

## Class Diagram - POC



## Sprint 2 – Initial Release

Table 3: Product backlogs and user stories

S/N	Backlog	User story
1.	Blog Home page	As a visitor, I must be able to visit blog home page so that I can view blog post.
2.	User Account	As a user, I must be able to create an account so that I can have access to the blogging platform
3.	User Login	As a user, I must be able to login to that I can create post
4.	Create Post	As a User, I must be able to create post so that visitors can view them.
5.	View Post	As a user, I want to be able to view only the post I created so that I can track my activity.
6.	Sort Post by Publication date	As a user, I want to be able to sort post by publication date so that I can identify the latest post.
7.	Retrieve New Post from API endpoint	As a developer, I must ensure the system automatically retrieve post from an API so that

## 9. challenge and solution:

Challenges	Solution
How secure is the endpoint how do we validate its records and ensure it is not sending any malicious information or any missing data for one or more fields?	Using validation object on each record.
Efficiently ensure we don't have duplicate posts especially when saving post from the endpoint.	Using Validation object and specifying uniqueness of the title field. This field was indexed to optimize speed. I also ensure efficient check within the retrieved records so that duplicate records are not saved to the db.
Ensuring the whole backend source codes is not bombarded with try catch blocks.	I implemented a global error handler, that catches errors, format them nicely and return to the front end.
How do we minimize visit to database during api calls.	One way I achieved this is by saving all posts at once to the db. This is a single request and executed after all the records have been validated.
How did you handle scaling of the system in response to high traffic	I implemented AWS load balancer to properly direct traffic within resources and with autoscaling group which can later be configured with scaling policy to scale the instances based on certain metrics like cpu utilization