Building a Social Media Platform Capable of Scaling to a Million Users Overnight

Harry Day

School of Computer Science

University of Manchester

Supervisor: Dr. Gareth Henshall

2025

Abstract

This is a brief summary of your dissertation. It should outline the research question, methodology, results, and conclusions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In today's increasingly digital-centred landscape, social media can transform lives and connect us in ways our ancestors could never have imagined. Social media enables friends and family to share their lives, connects customers to businesses directly, allows fans to interact with their sports teams and players, and helps employees find new jobs. Social media has allowed us to connect more and "play a positive role in strengthening the relationship between friends" (Chen et al. 2016).

One such social network that has seen consistent growth is LinkedIn. This professional social network connects professionals across all industries with their peers and colleagues. Whilst it is the gold standard, it would be impossible for the network to cater to the needs of each industry individually. This project aims to create a new social media platform enabling computer science students and software engineers to show off their personal projects

to their network, giving them a single place to direct friends, colleagues and recruiters to display their skills and engineering and creative skills. In the software industry, personal projects and past achievements are more valuable than academic grades as they show a willingness to learn and interest in software outside of work, leading to a more overall picture of a candidate than just an assessment centre.

New social media platforms often start slow but eventually hit a critical mass where growth becomes exponential, as with BeReal. In just two years, this pandemic-born platform exploded to an active user base of over 70 million (Curry 2025). One of the key focuses of this project, then, should be to create a highly scalable social media platform that can cope with the demands of millions of users overnight.

1.2 Aims and Objectives

This project aims to build a highly scalable web application, Omni, which enables users to share their personal projects online in a single space, creating an online portfolio they can share with friends, family, colleagues, and recruiters. Formally:

- Create a highly-rated (in user feedback) web application that is both easy to use and pleasant to view, which enables users to see projects that others have posted.
- To serve this web application, create a highly scalable backend using a microservices architecture capable of scaling from 0 to millions of

requests per minute.

- Create an API that enables third parties to interact with the platform, backed by scalable microservices.
- Create a database and schema allowing sharding to enable horizontal data storage scaling.
- Follow industry standards for microservices, system design, Kubernetes, and software engineering.

1.3 Scope and Limitations

As this project primarily focuses on creating a highly scalable backend for a web platform, the majority of the focus will be on this section of the project. The front-end website shown to users will contain the minimum viable product to display the backend features but will not have much front-end 'magic' to enhance the user experience. Additionally, the platform does not contain the attributes commonly associated with social media, such as likes, follows and comments. This is an effort to combat "fakeness" on social networks: following someone to boost your social status, posting low-effort and topical content to receive vast numbers of likes. This social network intends to function closer to an online blog or portfolio, allowing software engineers to show off the cool projects they are working on without the fear of 'creators' dominating the platform.

The platform will also not have any sort of algorithmic recommendation algorithm, although this would be interesting to explore in the future. Much

of the success of modern social media can be associated with the algorithmic suggestion of content for users to consume, as it "directly impacts user satisfaction, engagement, and retention" (Chen & Huang 2024). Despite the benefits, this type of machine learning algorithm could be a project within itself and falls outside the scope of building a scalable web platform.

1.4 Dissertation Structure

Chapter 2

Background

Lorem ipsum dolar

Chapter 3

Design

3.1 High-Level System Design

Omni will be primarily designed using the microservices architecture. This enables a highly distributed and scalable system, which is able to respond to changes in incoming requests dynamically. The approach should make use of common industry standards and technology as well as be portable across cloud providers. The aim is to build Omni cloud-natively so that it could scale exponentially in the case of a "viral" moment.

3.1.1 Kubernetes and Containerisation

Kubernetes is an orchestration tool which enables the management of pods across clusters. Clusters are defined as one or more nodes (physical machines) connected together, possibly across different data centres or even regions, which run containers known as workloads. Kubernetes is designed to manage the lifecycle of pods using the industry standard healthcheck end-

points `/healthz`, `/readyz` and `/livez`. This endpoints for an application tell Kubernetes whether a pod is running (`/livez`), ready to respond to requests, possibly checking dependencies like a database (`readyz`), and the general health of a pod through `/healthz`.

Using these endpoints, Kubernetes deployments are able to moniter the pods (a pod is a running unit containing usually one but sometimes more containers) they have spawned and if any of them die, for example if they crash, auto-heal and spawn a replacement. Deployments can also be configured to scale up the number of pods they are running when certain limits are reached. The possibilities are endless but some common examples could be CPU usage on a particular node or number of requests being received per time period.

This is what makes Kubernetes super powerful. Combining the autohealing and management of ephemeral pods with autoscaling leads to solutions which are highly adaptable to any scenario. For this reason, I will be utilising Kubernetes in the design of Omni. This means that I will need to create containerised applications to run the platform.

Containerisation is the process of modifying an application to run within a container. A container is a self-contained module which can be run without needing to worry about its dependencies or setup. For example if an application has a dependency on a particular version of Linux with a certain library installed, containerisation will mean that the user running the container can execute the application on any supported OS, without needing to worry about the dependencies.

The other added benefit of containers is that only the compiled binary is

needed in the execution environment. Rather than the host machine needing to compile the binary, or a CI/CD pipeline creating many binaries for various different operating systems, we can use one container, which contains only the compiled binary for the OS of the container.

So for Omni, the application(s) should be containerised and ready for deployment to a Kubernetes cluster in any of the major cloud providers (as well as bare-metal solutions).

3.1.2 Backend

Chapter 4

Implementation

Lorem ipsum dolar

Chapter 5

Testing and User Feedback

Lorem ipsum dolar

Chapter 6

Reflection and Evaluation

Lorem ipsum dolar

Bibliography

Chen, L., Zhu, J., Tang, Y., Fung, G., Wong, W. & Li, Z. (2016), The strength of social networks - connecting people and enhancing relationship, *in* '2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)', pp. 470–475.

Chen, Y. & Huang, J. (2024), 'Effective content recommendation in new media: Leveraging algorithmic approaches', *IEEE Access* **12**, 90561–90570.

Curry, D. (2025), 'Bereal revenue and usage statistics (2025)', `https://www.businessofapps.com/data/bereal-statistics/`. Accessed: 2025-04-09.

Chapter 7

Appendix

Lorem ipsum dolar