

---

# **Software Requirements Specification**

**for**

## **MicroBlog - Blogging Website**

**Version 1.0 approved**

**Krishnaswaroop Varati**

**Ansh Goel**

**Arjun Prashant**

**Vellore Institute of Technology, Chennai.**

**02-09-2021**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>Revision History</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
<b>3. External Interface Requirements</b>	<b>3</b>
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
<b>4. System Features</b>	<b>4</b>
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
<b>5. Other Nonfunctional Requirements</b>	<b>4</b>
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
<b>6. Other Requirements</b>	<b>5</b>
<b>Appendix A: Glossary</b>	<b>5</b>
<b>Appendix B: Analysis Models</b>	<b>5</b>
<b>Appendix C: To Be Determined List</b>	<b>6</b>

## Revision History

Name	Date	Reason For Changes	Version

--	--	--	--

# **1. Introduction**

## **1.1 Purpose**

This SRS specifies the software requirements of “MicroBlog - A Mini Blogging Website”. This is the first version of SRS for the given project. This SRS covers the entire project, including all dependencies, functional requirements and non-functional requirements.

## **1.2 Document Conventions**

This document adheres to the IEEE standard of SRS. The font used in this document is Times New Roman for topics and subtopics and Arial for the body. The font size also varies and is 18 for topics , 14 for subtopics and 11 for body. The text color is also as per standard with black text on white background.

## **1.3 Intended Audience and Reading Suggestions**

This document is intended for the developers, users and the testing staff. The following parts of the document contain brief information about the overall description of the project which includes brief information about each module such as user/admin/database. It also contains information about the design of the website. Various interfaces and functional dependencies that are required for the optimal functioning of the project are also mentioned in this document. The language used for creating the interface and the logic that allows the functioning of this webpage is also mentioned. I would like all readers to go through the document in the same order in which the document is written and request the developers to give more emphasis to modules 2,3 and 4 of this document.

## **1.4 Product Scope**

MicroBlog lets users post their life experiences or stories for the world to read. It allows users to also read the blogs posted by other users from around the world . The goal of this project is to create a healthy and respectful environment of people that can share their thoughts and stories over a global platform that is accessible to other people.

## **2. Overall Description**

### **2.1 Product Perspective i/p-task-o/p**

This is a completely new and self-contained project which is partially inspired from existing products but the design and structure of this product is unique. This SRS defines the overall project which includes every module of the project.

### **2.2 Product Functions**

- The website allows people to sign up and become a user of the webpage.
- A user is required to provide his Name, E-mail address, and password for creating his account.
- A user can create his own mini-blog or can read other users' mini-blogs.
- For creating a mini-blog the user is to provide a suitable title and description for his/her mini-blog.
- The user can create multiple mini-blogs.
- Users can access their mini-blogs from the past too.

### **2.3 User Classes and Characteristics**

There is only one user class that uses this webpage, they are the “standard users” who can write or read mini-blogs. There is an admin who can monitor and access the data of any user to ensure proper functioning of the webpage. The admin can flag inappropriate posts and delete such blogs and posts violating the rules.

### **2.4 Operating Environment**

The server is hosted on a Heroku Dyno which runs on Unix. It has 512 MB of RAM.

### **2.5 Design and Implementation Constraints**

1. MongoDB collection limit for free tier plan (512 MB)
2. Heroku Uptime Constraint of 550-1000 Hrs per month
3. Heroku Ram Constraint of 512 MB
4. Heroku Sleep property, program sleeps after 30 minutes of inactivity.
5. NoSQL commands for queries.

## **2.6 User Documentation**

A help section will be available on the webpage.

## **2.7 Assumptions and Dependencies**

### **2.7.1 Assumptions:**

1. Less than 100 users
2. Less than 2000 Words in a post
3. Blog title and description is together less than 100 characters

### **2.7.2 Dependencies:**

1. Python 3.9 or later
2. Flask 2.0.1
3. gunicorn 20.1.0
4. Werkzeug 2.0.1
5. Jinja2 3.0.1
6. Flask-Bcrypt 0.7.1
7. Flask-Login 0.5.0
8. python-dotenv 0.19.0

## 3. External Interface Requirements

### 3.1 User Interfaces

The header of all the pages contains a navigation bar. The footer contains the details and other contacts.

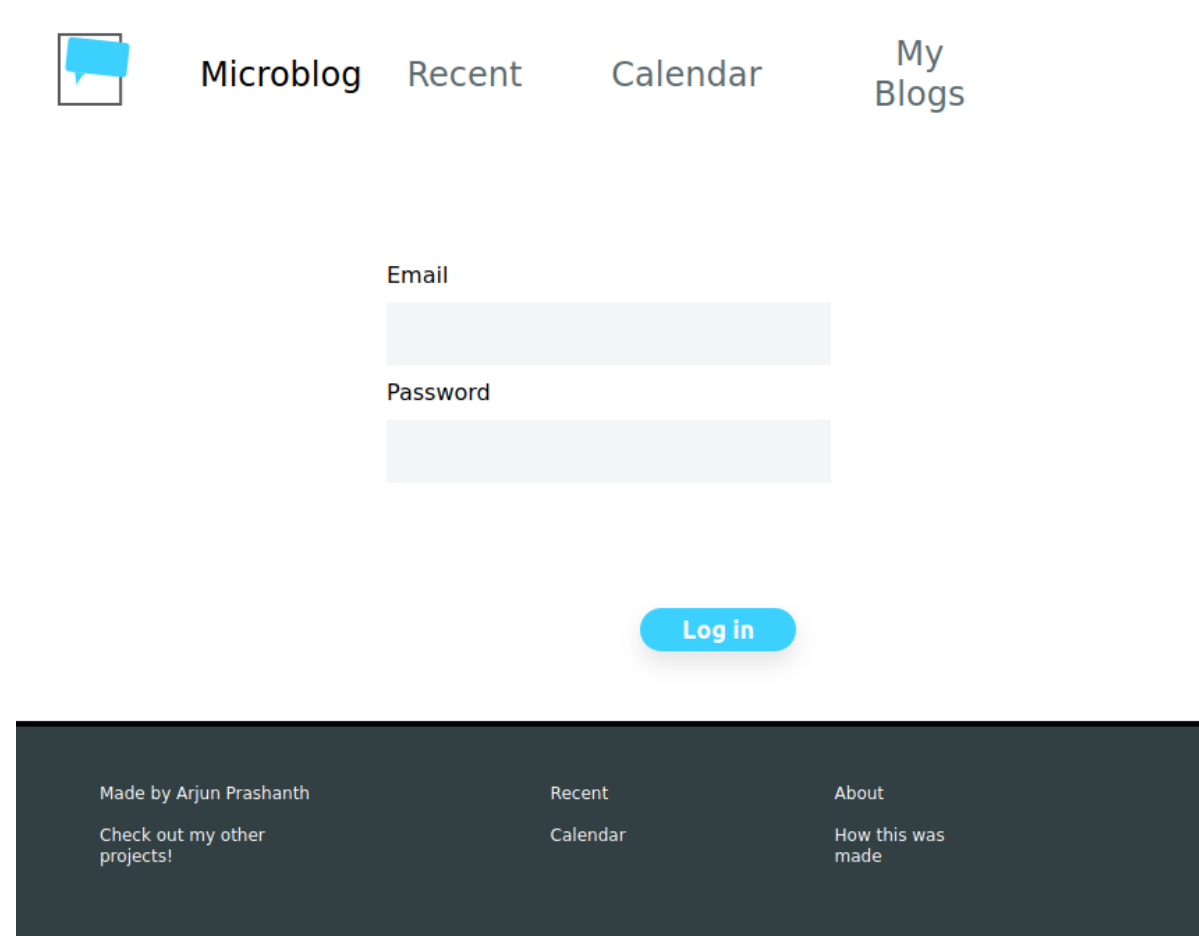
The primary theme is light and the primary color of the webpage is blue.

The website requires the user to sign up to use ;



The sign-up page has 4 fields taking in email, password, first name and last name :

Login:



The image shows a login form with a navigation bar at the top. The navigation bar contains a blue speech bubble icon, the text 'Microblog', 'Recent', 'Calendar', and 'My Blogs'. Below the navigation bar, there are two input fields: 'Email' and 'Password'. Below the 'Password' field is a blue 'Log in' button. At the bottom of the form is a dark grey footer bar with three columns of text: 'Made by Arjun Prashanth' with a link 'Check out my other projects!', 'Recent' with a link 'Calendar', and 'About' with a link 'How this was made'.

Microblog Recent Calendar My Blogs

Email

Password

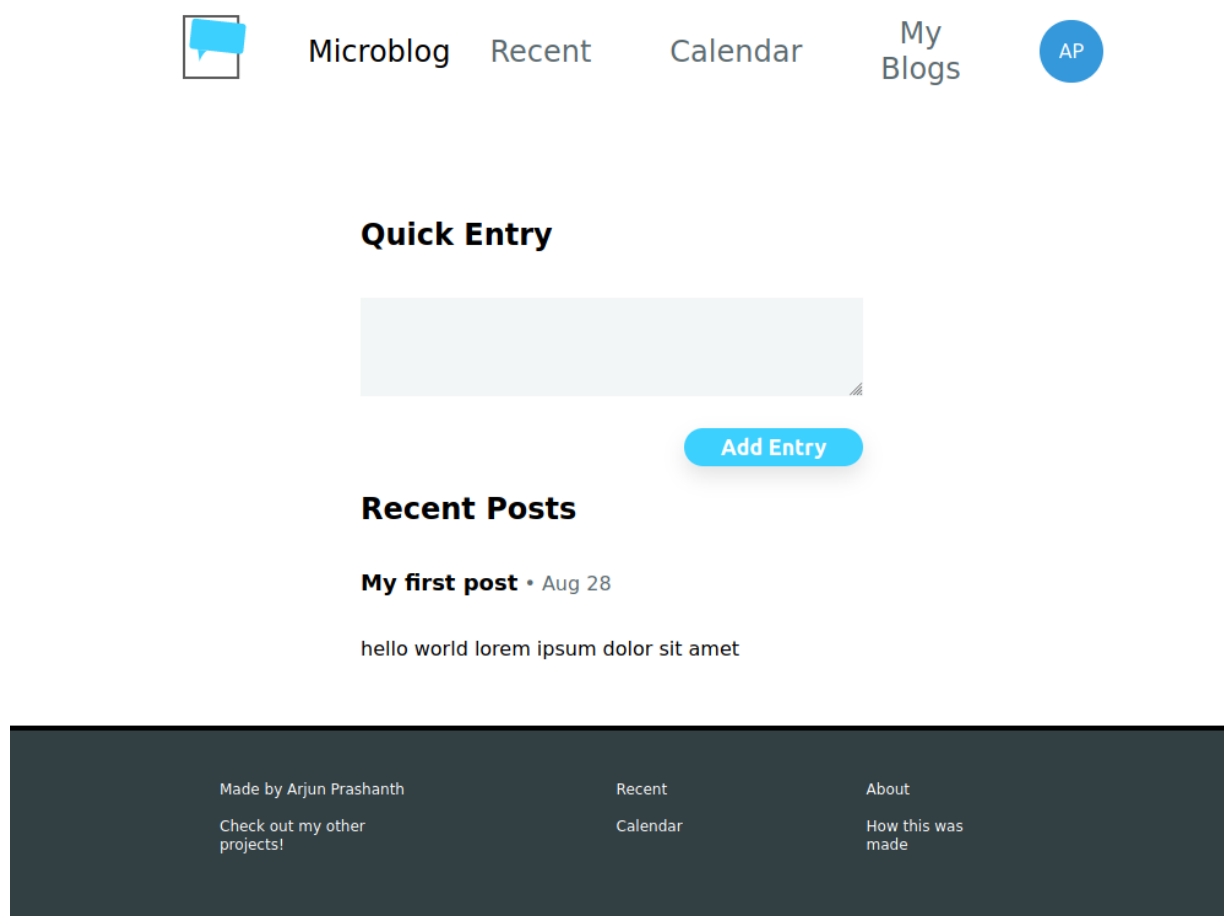
Log in

Made by Arjun Prashanth  
Check out my other projects!

Recent  
Calendar

About  
How this was made

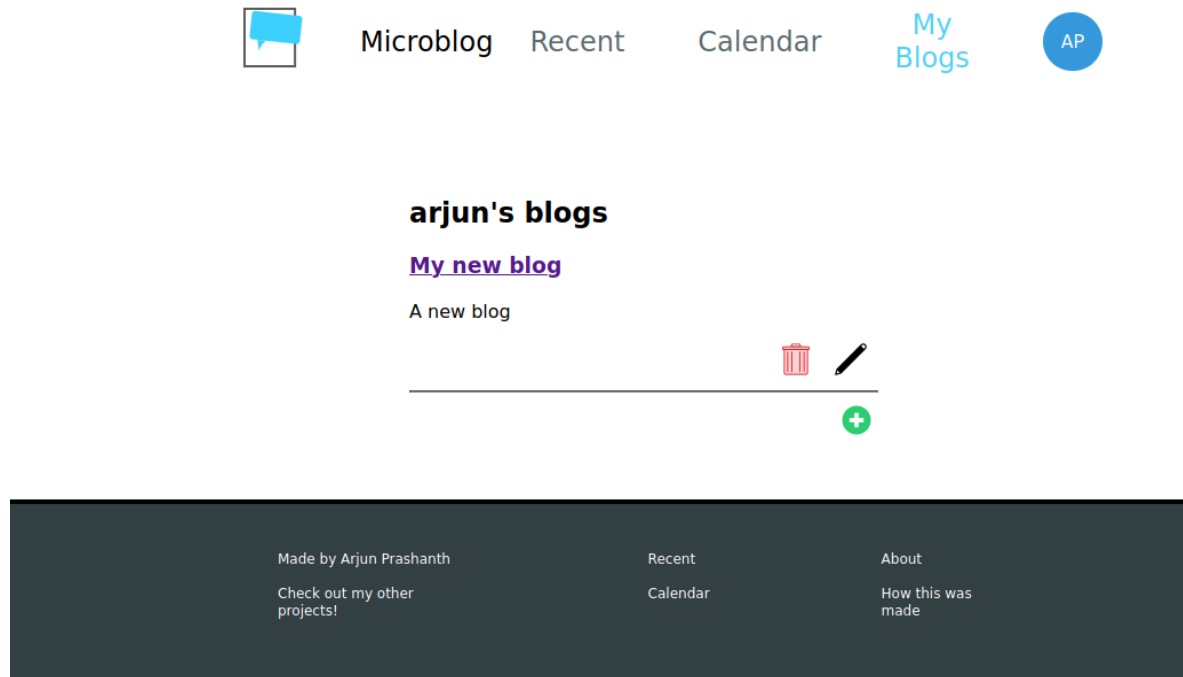
Once logged in the user can create a new blog instantly with the *Quick Entry* section. The top 5 recent posts of the user are also displayed below and clicking on it leads to the respective blog page and the post.



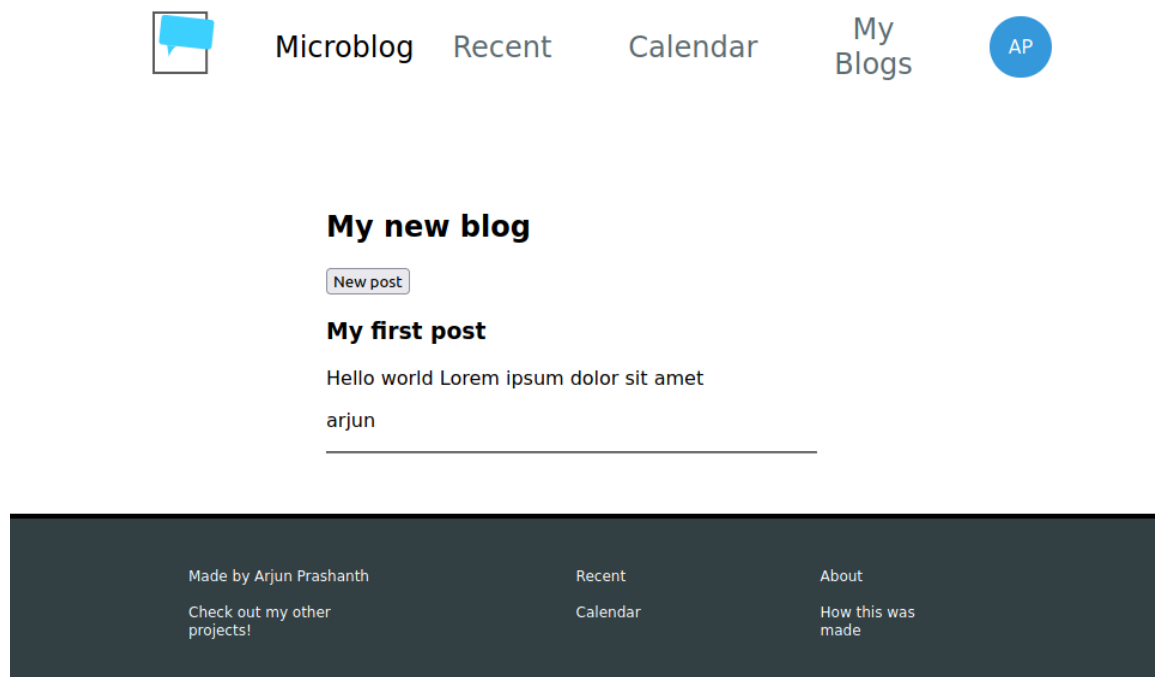
Clicking on the *My Blogs* navigation it takes the user to a page showing his list of blogs along with the title and description.

The user also has the option to edit and delete the blogs. New blogs can be created here using the *plus* icon.





Further, clicking on the title of the blog leads to the post page where all the posts by the author are shown for that blog and new posts can be added here too.



The new post button allows the user to create a new post and takes in a post title and content .

The page also shows the posts on the particular blog page.

### 3.2 Hardware Interfaces

Hardware components required are just a keyboard and a mouse and any device which has connection to the internet.

### 3.3 Software Interfaces

Software used	Description
Operating system	Linux
Database	MongoDB cloud database
Python	to connect to the Database and update the web pages with server data Python has been used.
HTML	To create the structure of the WebPage HTML is used.
CSS/JS	To style the web page and add items such as dropdowns

### 3.4 Communications Interfaces

This project supports all web browsers. It uses HTTPS protocol for network communication.

## 4. System Features

### 4.1 Blog & Post Saving/Retrieval

#### 4.1.1 Description and Priority

The blogs and posts made on blogs by a user will be saved and retrieved from a MongoDB cloud database. It has high priority as it is the core of the website

#### 4.1.2 Stimulus/Response Sequences

This is triggered when a user makes a new blog, makes a new post, edits a blog, edits a post, deletes a blog, deletes a post, loads the blogs page and loads the posts page.

#### 4.1.3 Functional Requirements

REQ-1: *pymongo library in python.*

## 4.2 User Signup and Login

### 4.2.1 Description and Priority

The user should be able to register themselves with an email and create a password to access the account. The user data is stored in the MongoDB cloud database. It has high priority as it is the core of the website.

### 4.2.2 Stimulus/Response Sequences:

This feature is triggered when a user tries to log in to his account or tries to signup and create an account for this website.

### 4.2.3 Functional Requirements

This feature needs to store the user password in the database and to prevent password storage insecurity, the password is hashed using bcrypt.

REQ 1: flask-bcrypt module

REQ 2: pymongo

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

Data from the backend must be retrieved quickly else it will bottleneck the application, therefore it is preferable to use mongoDB aggregation pipeline over returning all documents and then performing operations on it.

### 5.2 Safety Requirements

If any database shard becomes corrupted or goes down, the other shards must store redundant data so as to provide data seamlessly without interruptions.

### 5.3 Security Requirements

Database Vendor must be chosen carefully as sensitive data such as email and password should not be leaked

### 5.4 Software Quality Attributes

The software must be scalable and available.

## 5.5 Business Rules

Under specific circumstances such as misuse of the website, Admins may remove posts or blogs which violate Rules and regulations.

## 6. Other Requirements

## Appendix A: Glossary

Definitions, Acronyms and Abbreviation:

SRS: Software Requirement Specification

Client/User: The entity who will be using the microblog website

Server: A system that runs in Linux that monitors the User Blogs.

RAM: Random Access Memory

Dyno : A Heroku server machine

OS: Operating System (LINUX)

MongoDB: ANoSQL database management system

HTTP: HyperText Transfer Protocol

Username: Unique name given to each account of digital library

Password: Unique word given to each user as a secret code

*Flask-Bcrypt* is a *Flask* extension that provides *bcrypt* hashing utilities for your application

- HTML
- Cascading Style Sheets (CSS)
- Javascript (JS)
- Python