# Project Title

A Blogging Platform

# Project Outline

Develop a blogging platform where users can sign up for accounts, publish blog posts, read blog posts and comment on these posts.

# Functional Requirements

1. Home Page:
   1. This Page should display recent posts in a chronological order (latest to oldest).
   2. Only the Title, Author, Creation Time, and a short summary of the posts should be shown on this page.
   3. The Post Title should link to the Post.
   4. The Home page should be accessible by all, i.e., even unauthenticated users can access the home page.
   5. The Home Page as well as all pages should have a nav bar
2. Nav Bar: The Nav Bar should have links to
   1. The Home Page
   2. Signup Page
   3. Login/Logout Page
   4. Create Post Page
   5. Update Post Page
3. The Login/Logout Pages:
   1. These pages can be the default Spring Security Pages
   2. If you want you can create your own views for Login and Logout, but its optional
   3. The login page should allow users to sign in if their credentials are stored in the database.
4. Access Control
   1. There should be two types of users: Admin & Blogger
   2. Admins should be allowed to create posts, and update/delete all the posts in the blog.
   3. Bloggers should be allowed to create posts, but they should only be allowed to update/delete their own posts.
5. The Signup Page
   1. The signup page should allow a user to sign up as a blogger
   2. Admin accounts should be created directly in the database
   3. Once signed up these users should be able to login using the login page
6. Create Post Page
   1. This page should contain a form with fields to add a new post.
   2. It should have the following fields
      1. Title
      2. Summary
      3. Body
   3. The entity used to store this data can have many other fields. It is up to you to analyse and add fields that will be helpful to implement the functionality required for this project.
7. Post Page
   1. Every post created should have a unique url
   2. The posts visible on the home page should link to the url of their respective post.
   3. This page should display the data of all fields related to a post.
   4. If the logged in user is the author of the post, then a delete button should be visible. When this delete button is pressed, the post should be deleted.
   5. The post should be followed by a form for adding comments
   6. This form should only be visible for authenticated users
   7. On submitting the form, the comment should be added to the post, and should be visible below the post.
   8. The comment should show the content, creation date, and the author of the comment.
   9. If the logged in user is the author of the comment, there should be a delete button that can be used to delete the comment.
8. Update Post Page
   1. This page should first take an id
   2. Upon passing the id a prefilled form should be visible that contains the data for all the fields of post with that id
   3. After modifying the data using the form, and performing proper validations, the post data should be updated.
   4. Make sure not to make the id field updatable.

# Creative Freedom

Anything not mentioned in the functional requirements, but necessary to develop the complete application can be added in by the student.

The student is supposed to make educated decisions on how to place different elements for different functionalities so as to display a better UI design.

# Marking Scheme

You will be marked on the following parameters:

1. The complete application should be developed using the Microservices Architectural Style. You will be scored on the domains that you select for your Microservices.
2. Package names should confirm to the Layered Architecture or Domain Driven Design. You will be scored on your choice of package names.
3. You will be given marks for following Industry Best Practices.
4. Your APIs should be RESTful and should implement proper mappings for CRUD Operations. You will be scored for proper mappings.
5. You should use Authentication and Authorization in your application. You will be scored for proper implementation.
6. You will be scored for proper implementation of Zuul Proxy and Eureka Discovery Service.
7. Finally, you will be scored for all the functionalities defined in this document.

# Submission

1. Create a new STS Workspace
2. Develop all the projects in this workspace
3. Compress the workspace folder as a .zip file
4. Rename the folder to **p1\_yourFullName.zip** (If you don’t follow this naming convention, then it will become hard to score your project)
5. Submit it to the appropriate person