**Blog with Ez**

DevOps Project Report

****

**Department of Computer Science and Engineering Jaypee Institute of Information Technology University, Noida**

**November 2020**

**Under the supervision of: Submitted by:**

**Dr. Sulabh Tyagi Rishabh Kejariwal 17103355**

**Akash Sharma 17103180**

**Ojas Bansal 17103316**

**Tanmay Agarwal 17103350**

**Shreesh Srivastava 17103351**

**TABLE OF CONTENTS**

* Project Description
* Main Components
* Languages/Technologies Used
* Project Features
* Project Management – JIRA
* Version control system – Git
* CI/CD Pipeline - Travis CI

1. PROJECT DESCRIPTION

Blog with Ez is a blogging web application where anyone can read or write articles under different categories. A user must login/signup before creating a new article. Each user can update/edit its own articles. An additional feature is added i.e. there will be admin(s) who can remove any other user if found the content inappropriate.  
Link to Access: [BlogWithEz](https://blog-with-ez.herokuapp.com/)

1. MAIN COMPONENTS

The blogging web application is divided into following main components:

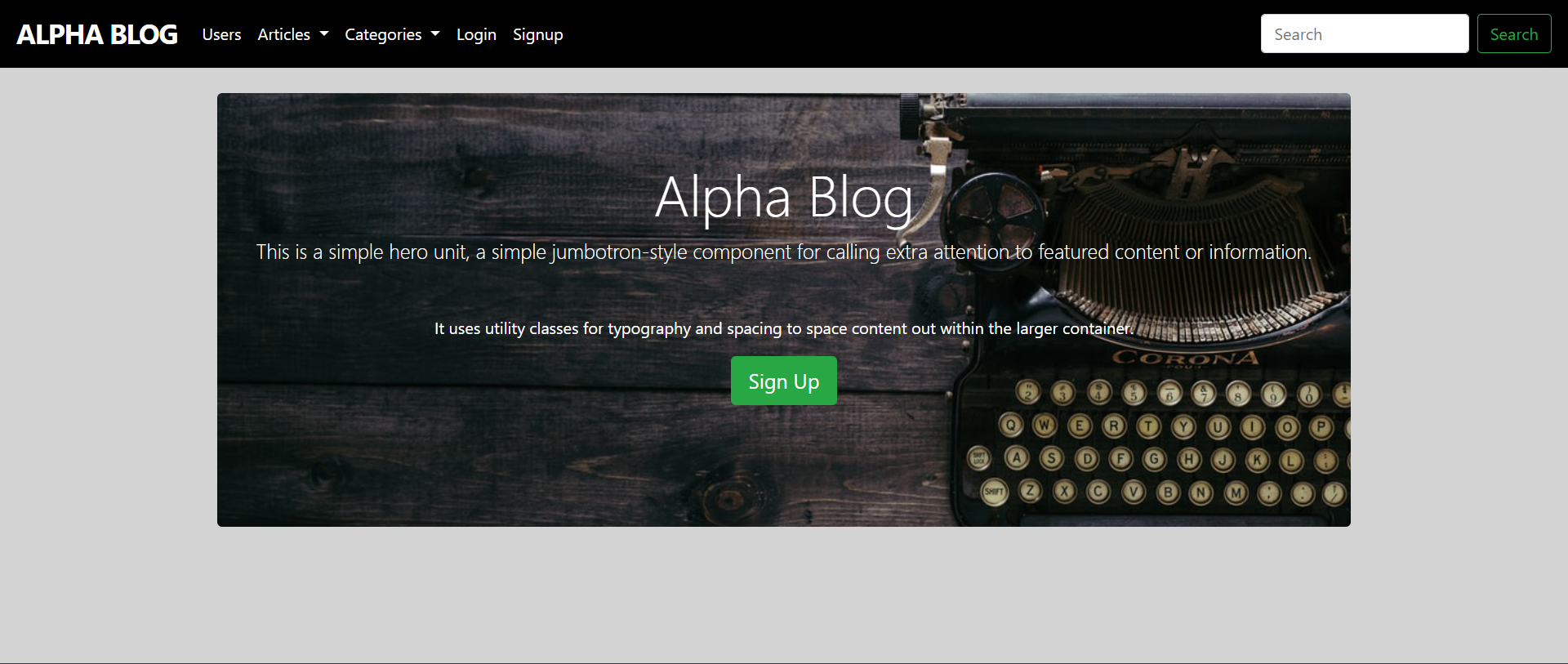
* **Authentication** :
* Login Page
* Sign Up Page
* **Admin:**
* Profile Page
* View all users
* View all articles
* Create new category
* **User View:**
* DashBoard
* Profile
* Write new article
* View all users and their articles

1. LANGUAGES/TECHNOLOGIES USED

* **FrontEnd Interface:** HTML, CSS, Ruby, JavaScript
* **BackEnd Interface:** Rails
* **Database:** SQLite (Relational Database)
* **Project Management Tool:** JIRA
* **Version Control System:** Git
* **CI/CD Pipeline**: Travis CI
* **Deployment Tool:** Docker

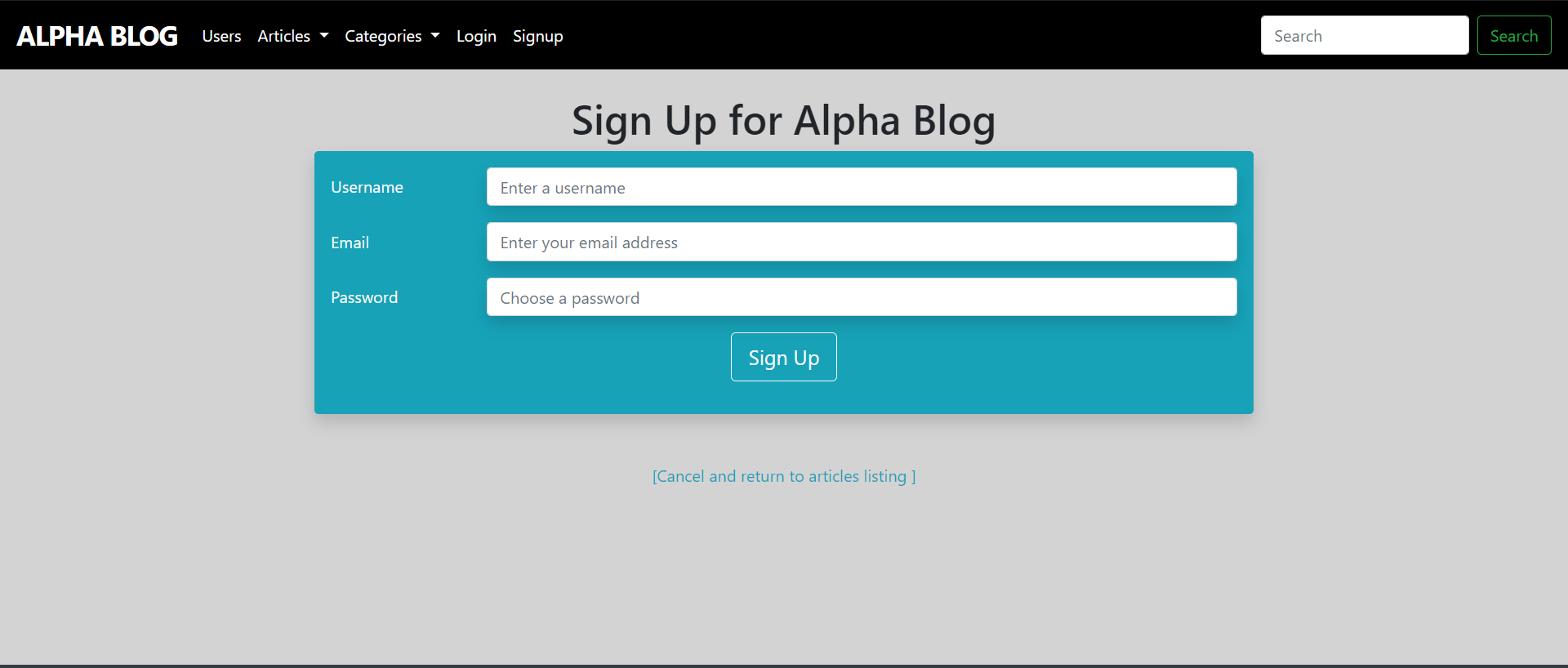
1. PROJECT FEATURES

**WELCOME PAGE:**

****

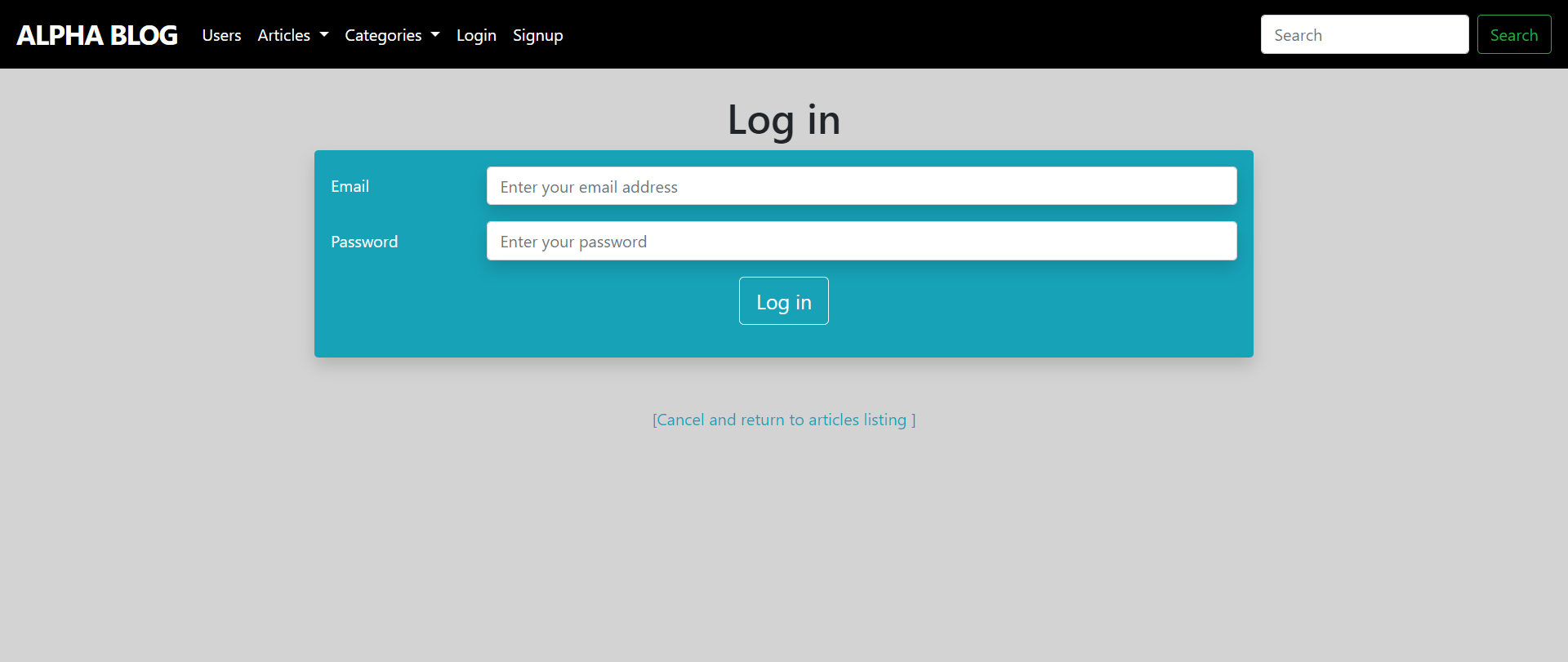
**REGISTER PORTAL:**

Users can register themselves to create blogs .



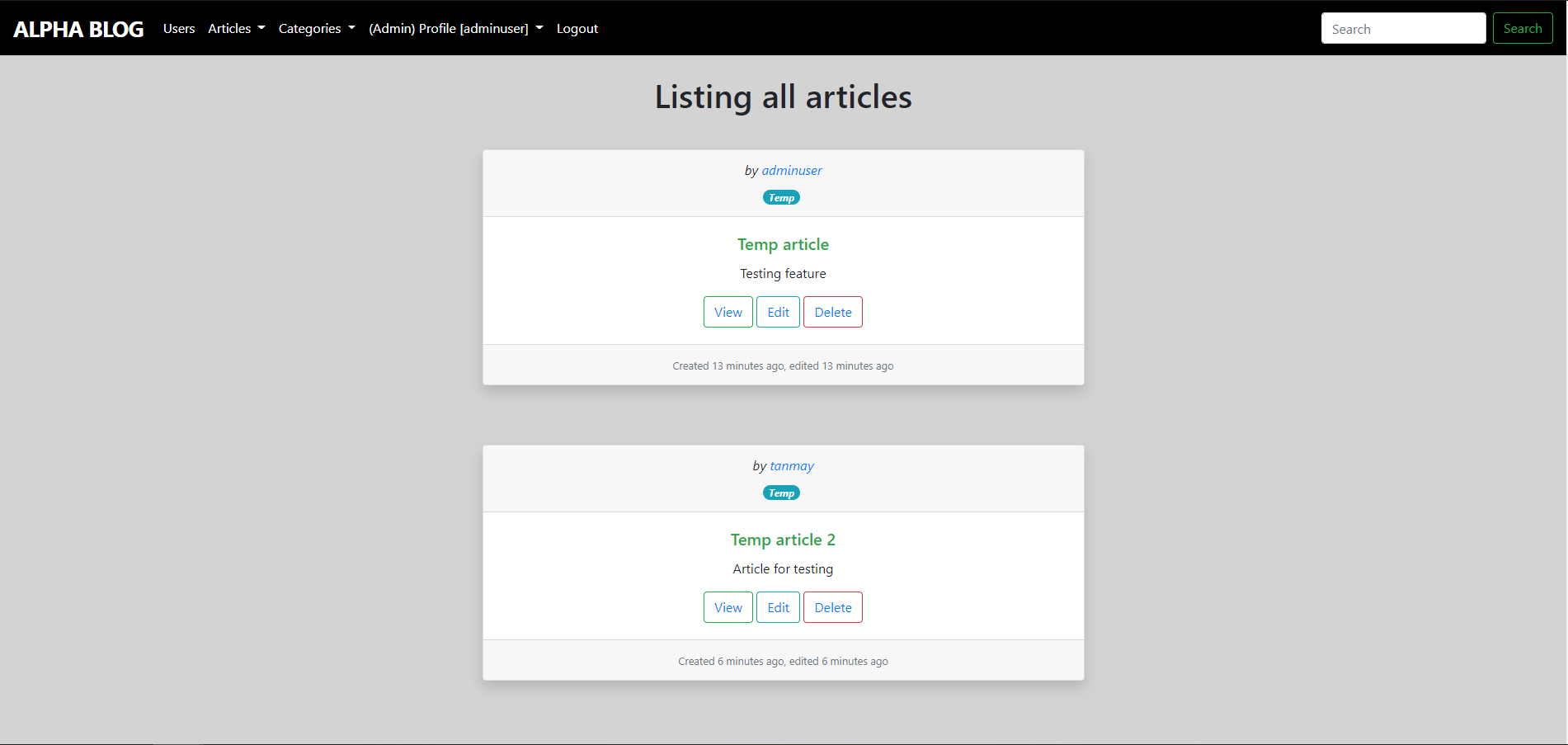
**LOGIN PORTAL:**

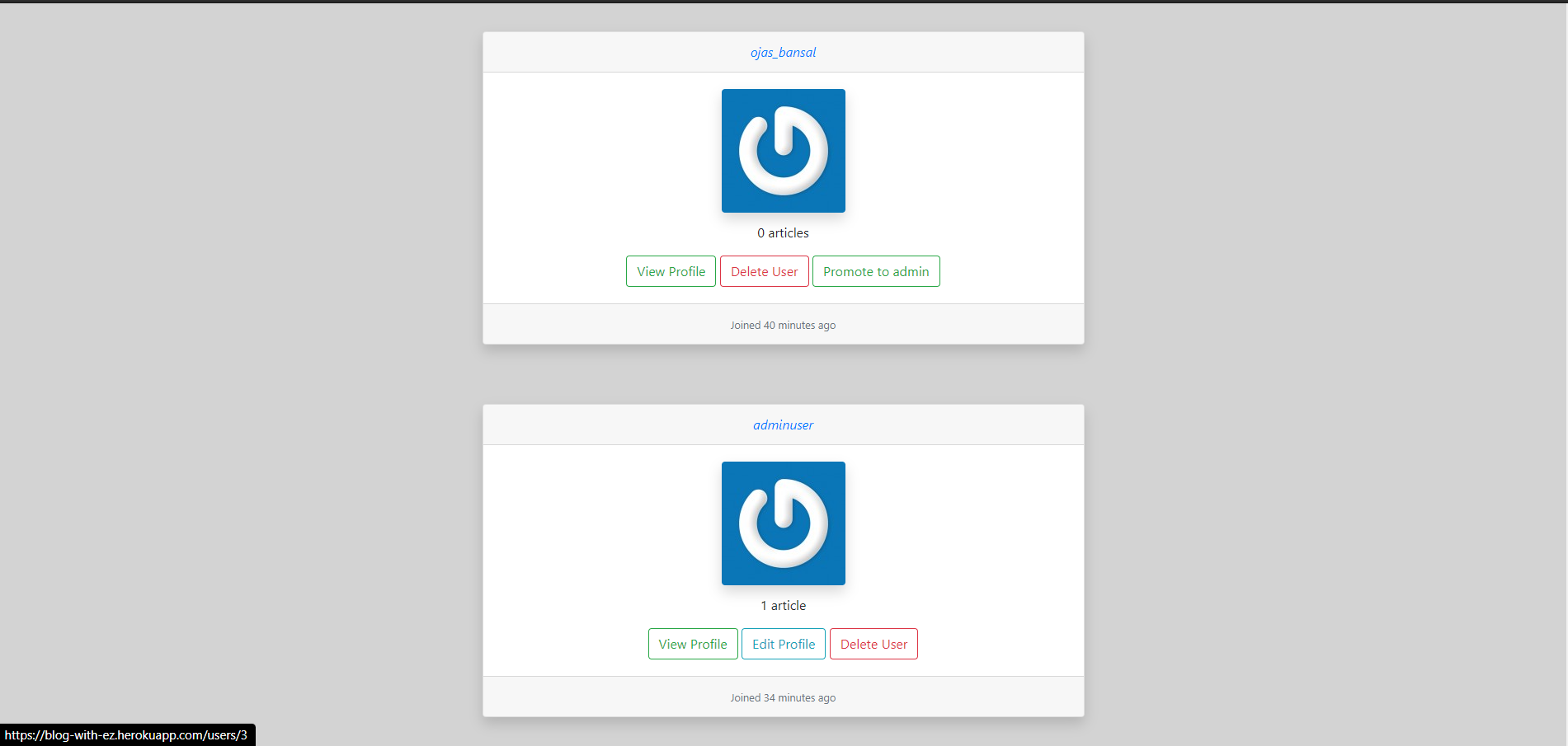
Users can login themselves as a Normal User or admin.



**ADMIN DASHBOARD:**

User logged in as admin can delete any user, promote an user to admin, edit/delete any article and create new categories.





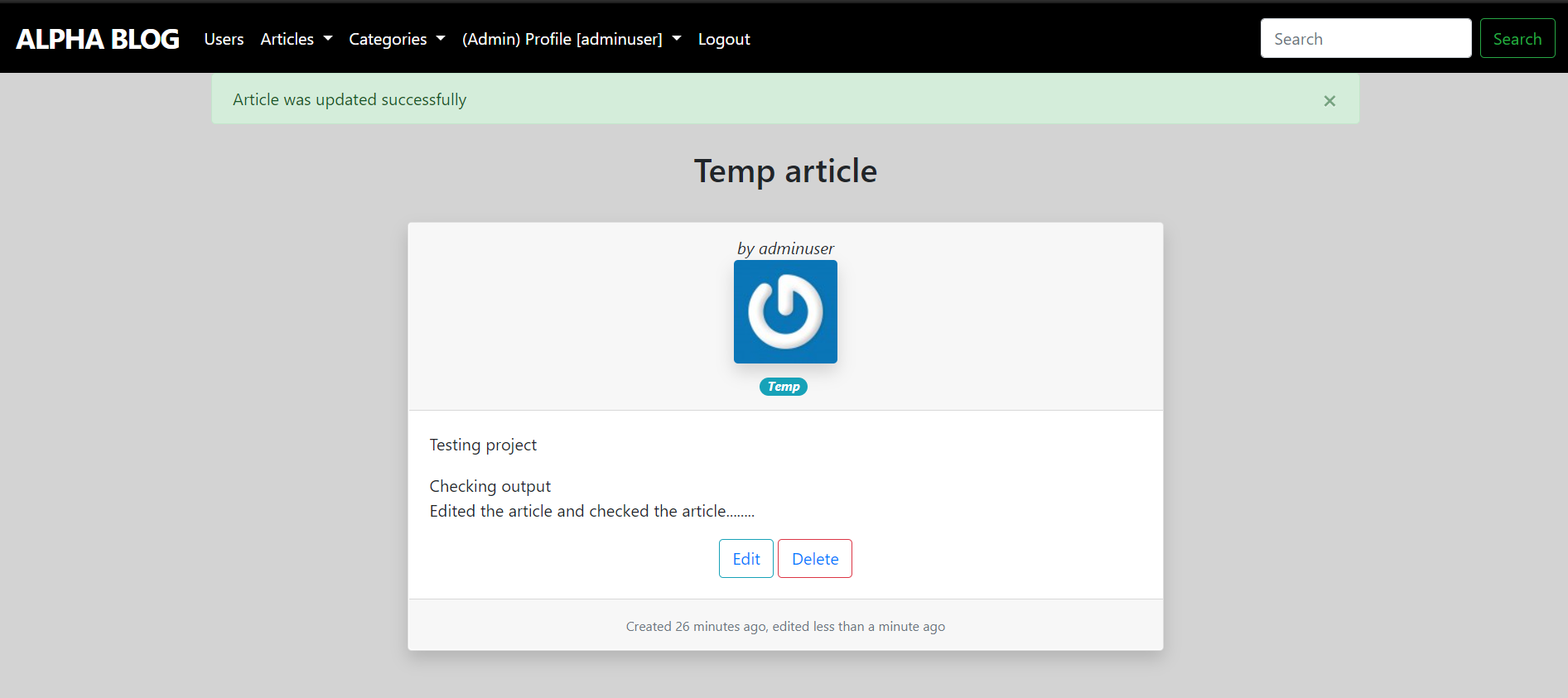
**USER DASHBOARD:**

Normal User can view any article and edit/delete articles created by the respective user, edit/delete their profile.

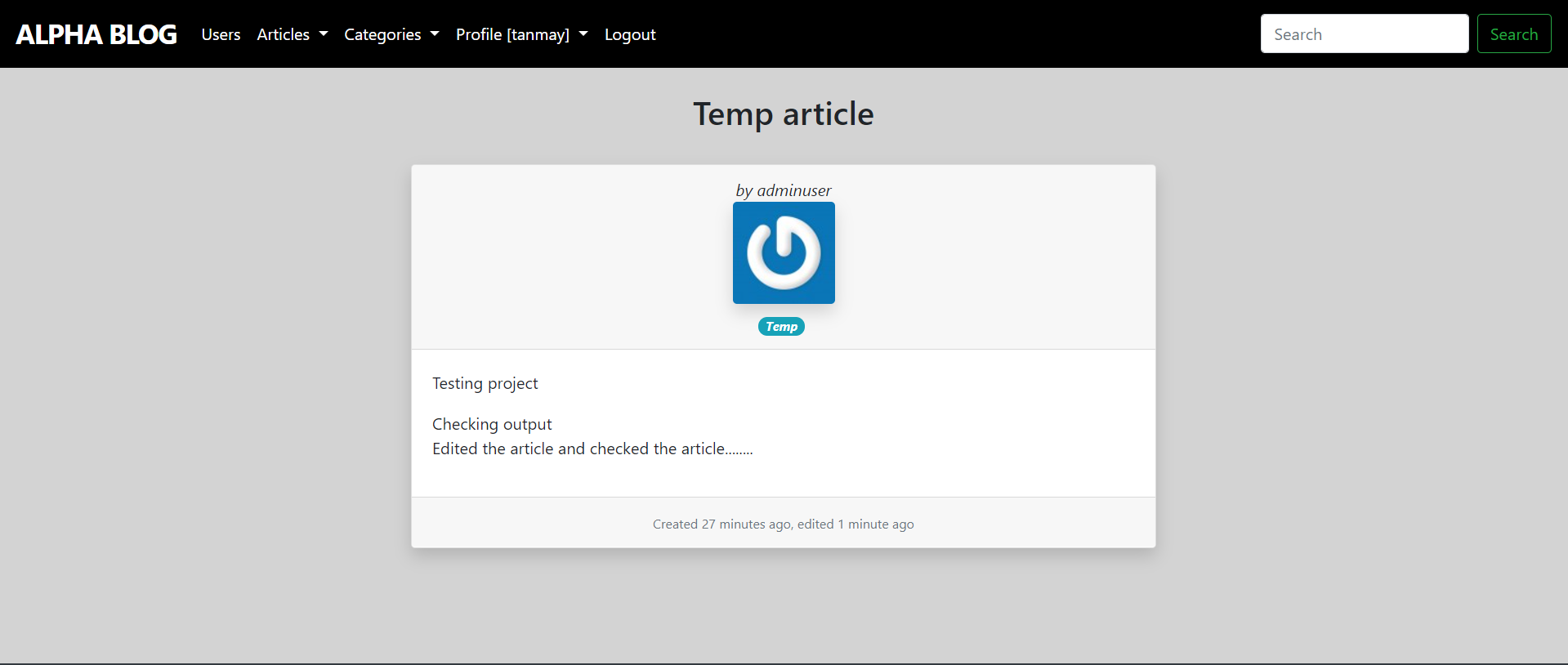


**ARTICLES:**

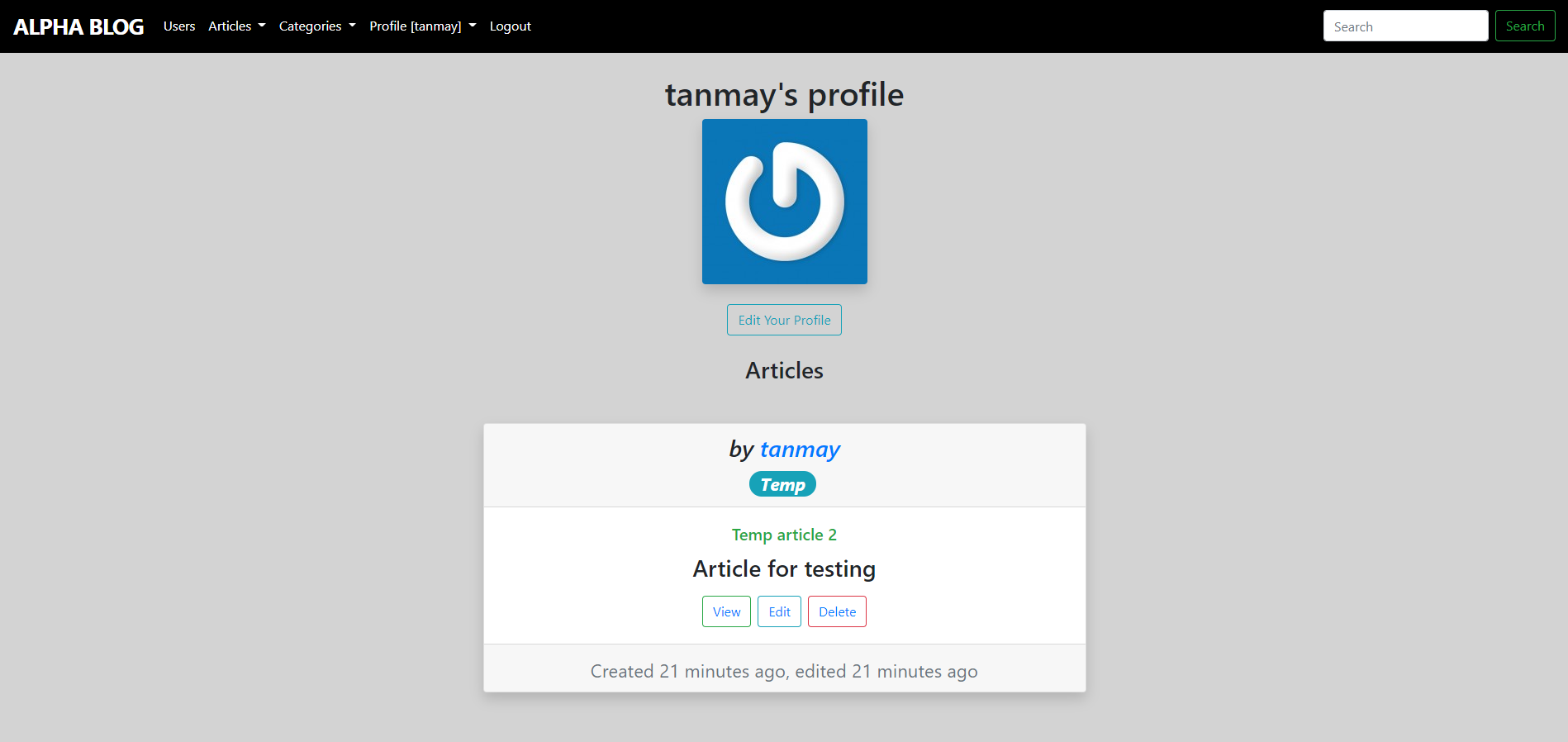
Admin/Article Owner view:

****

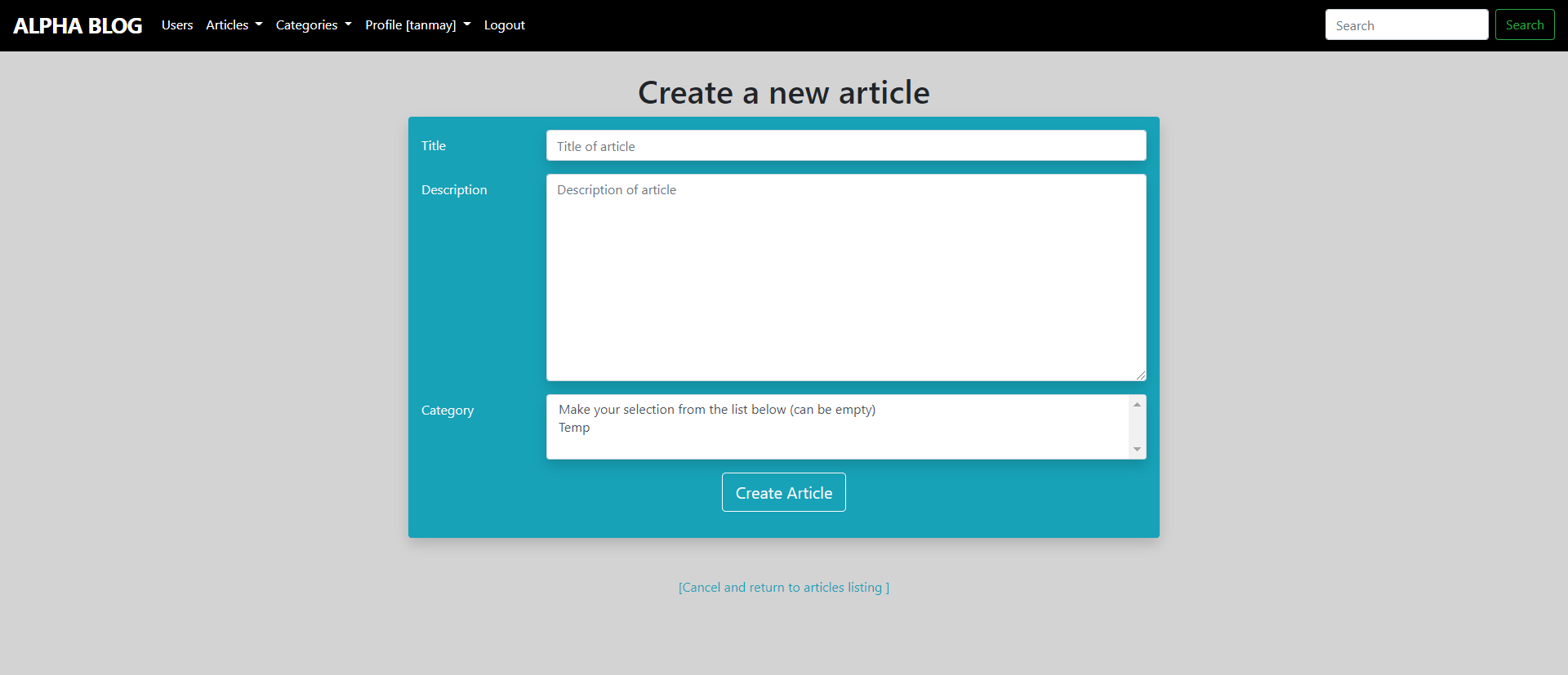
Other User view:

****

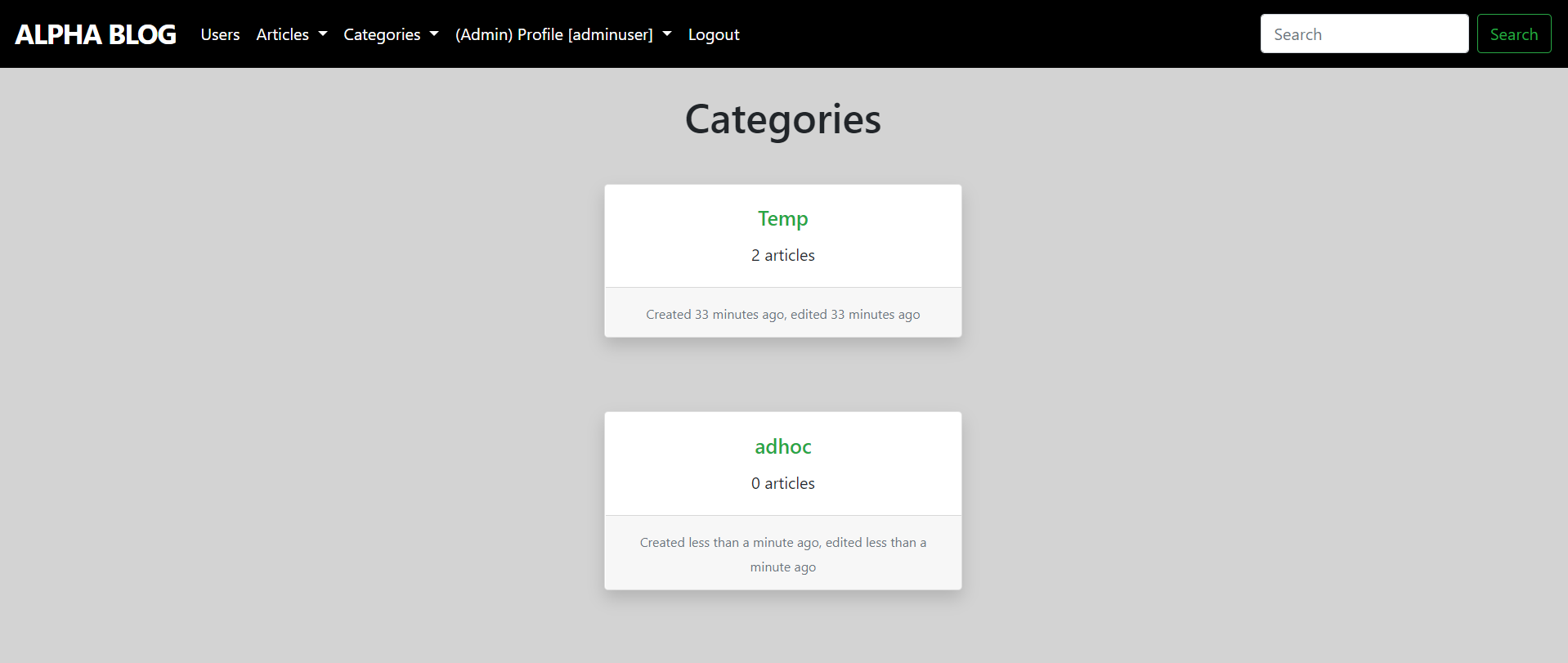
**PROFILE VIEW:**



**Creating Articles:**



**CATEGORY VIEW:**

****

1. Project Management

**JIRA :**

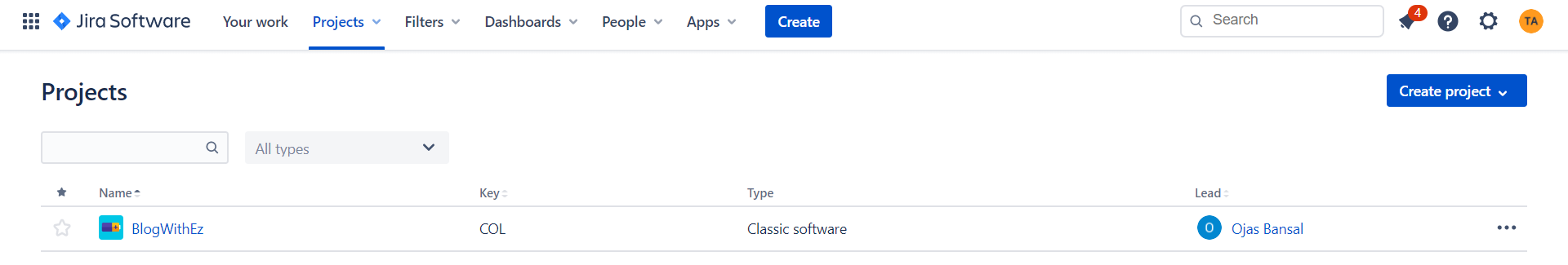
JIRA is a centralized, highly customizable project management system. All projects are logged into a central database and each one goes through a number of workflows (processes). Workflows control the status of the project as well as the rules by which it transitions to other statuses.

**User Stories:**

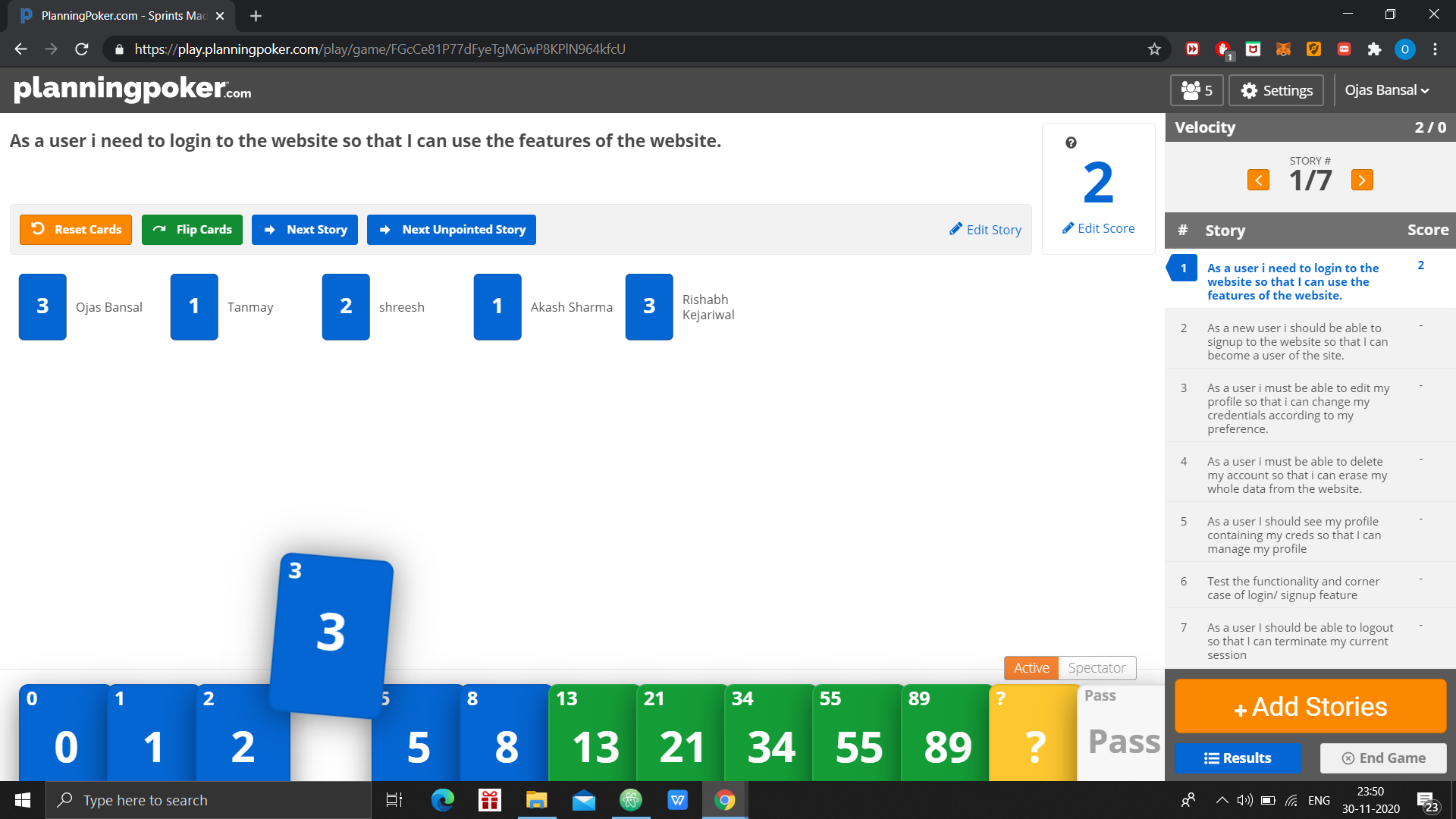
1. As a user i need to login to the website so that I can use the features of the website.
2. As a new user i should be able to sign up to the website so that I can become a user of the site.
3. As a user i must be able to edit my profile so that i can change my credentials according to my preference.
4. As a user i must be able to delete my account so that i can erase my whole data from the website.
5. As a user I should see my profile containing my creds so that I can manage my profile.
6. Test the functionality and corner case of login/ signup feature
7. As a user I should be able to logout so that I can terminate my current session.
8. As a user i should be able to write and post article.
9. As a admin i must be able to delete any article present on the website so that i can make sure that content on the website follows the general rules and regulations.
10. As a user i must be able to edit my pre-written article so that i can update the changes as required.
11. As a user i must be able to edit my pre-written article so that i can update the changes as required.
12. As a admin i must be able to add new category so that users can divide their articles in a more vast way.
13. As a admin i must be able to delete or edit any category so that i can enhance the content of the website.
14. As a new user i should be able to surf the website without logging in so that i can read articles written by other users.
15. As a user i must be able to see my profile page or other users profile pages so that i can specifically look for an article written by any particular user or myself.
16. As a user i must be able to see the different categories present on the website and articles present under them so that i can easily search for whatever topic i want to read.
17. As an admin i must be able to promote other users to admin so that i can divide my work in a broader team.
18. As a user i should be able to see the name of all registered users and articles written by them without logging in so that i check the popularity of the website.
19. As a user i should be recommended the categories while posting new articles so that i can tell other users that what category my article belongs to.

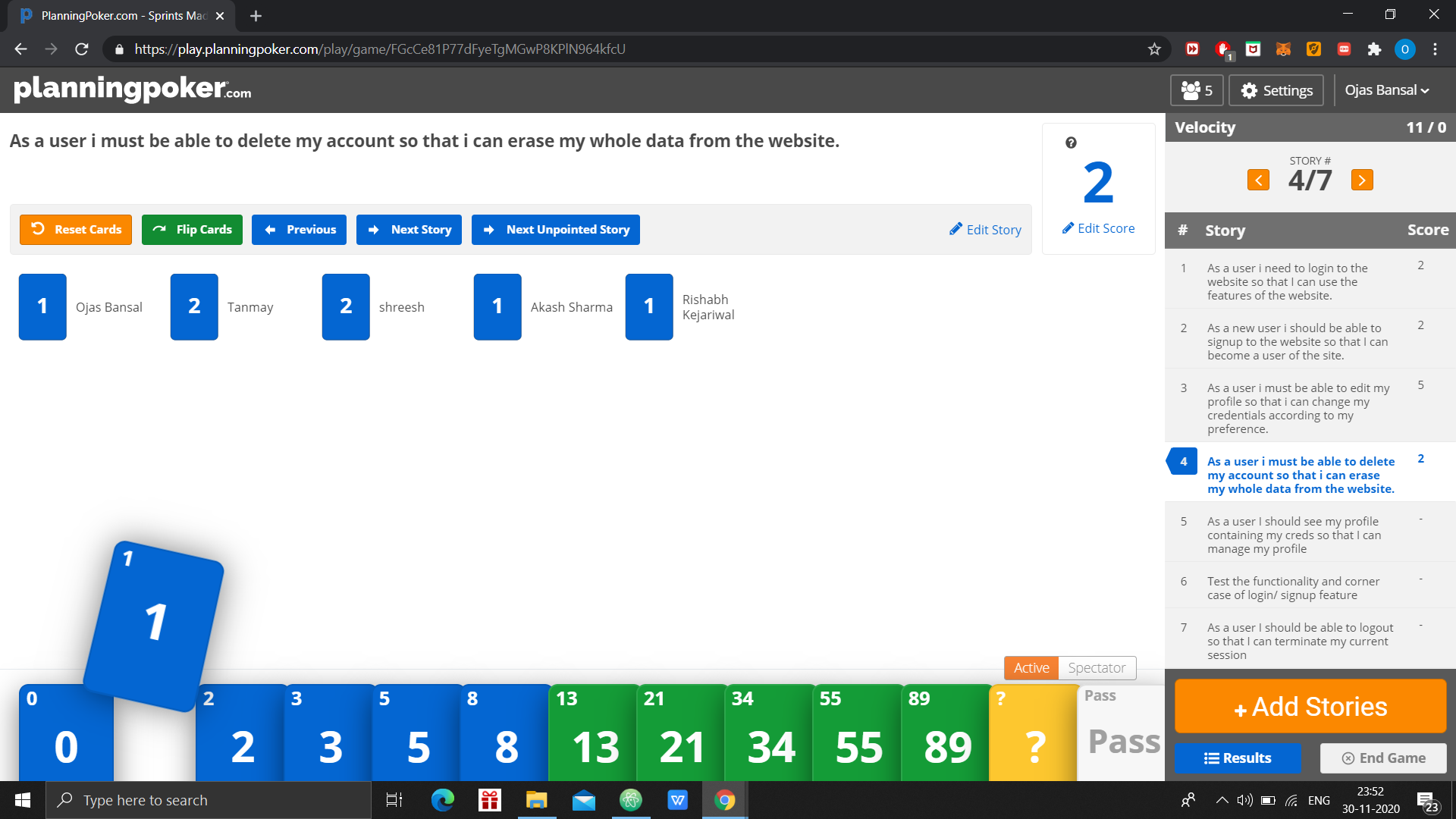
**Screen Shots :**

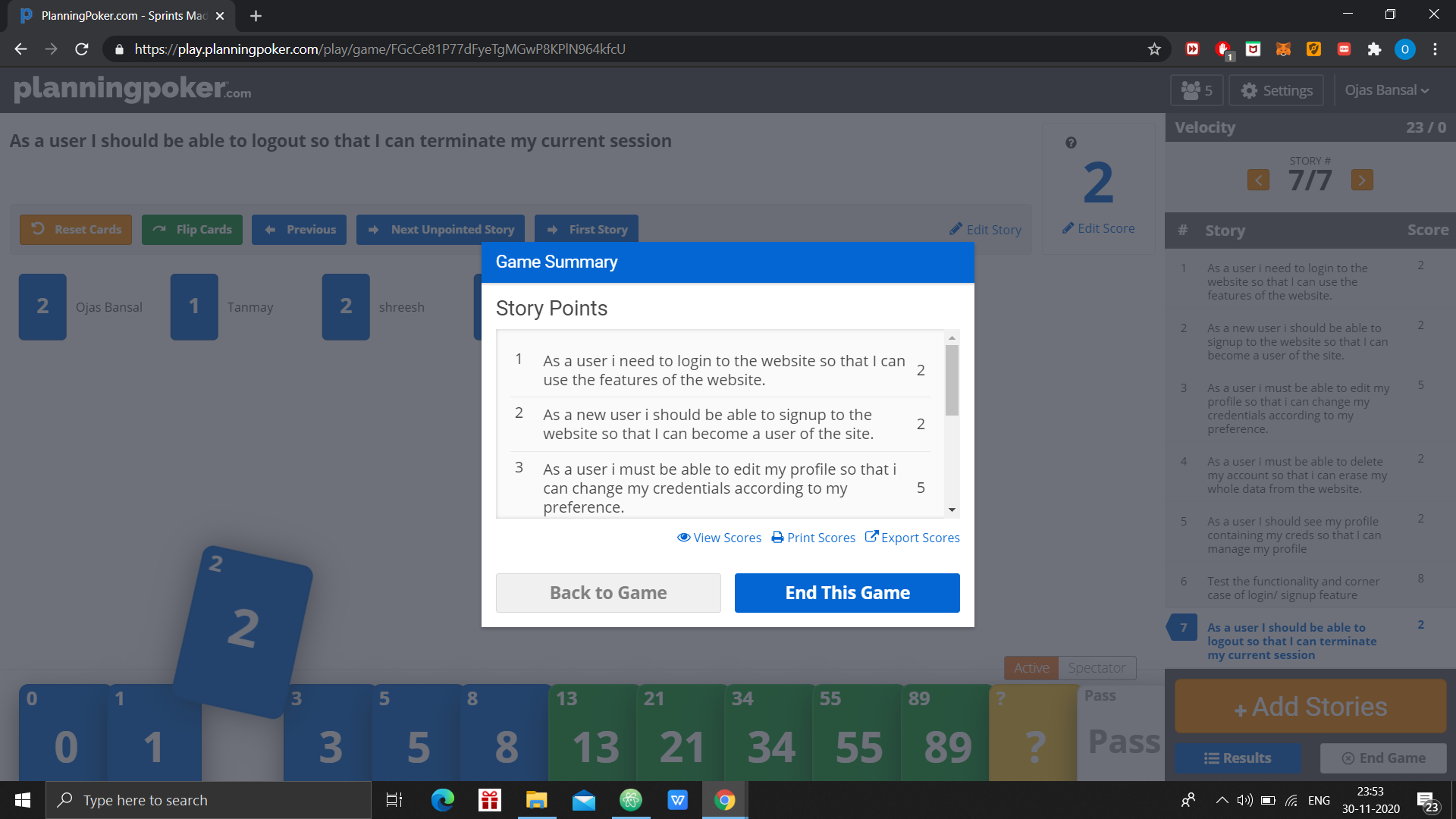
* **DashBoard :**

****

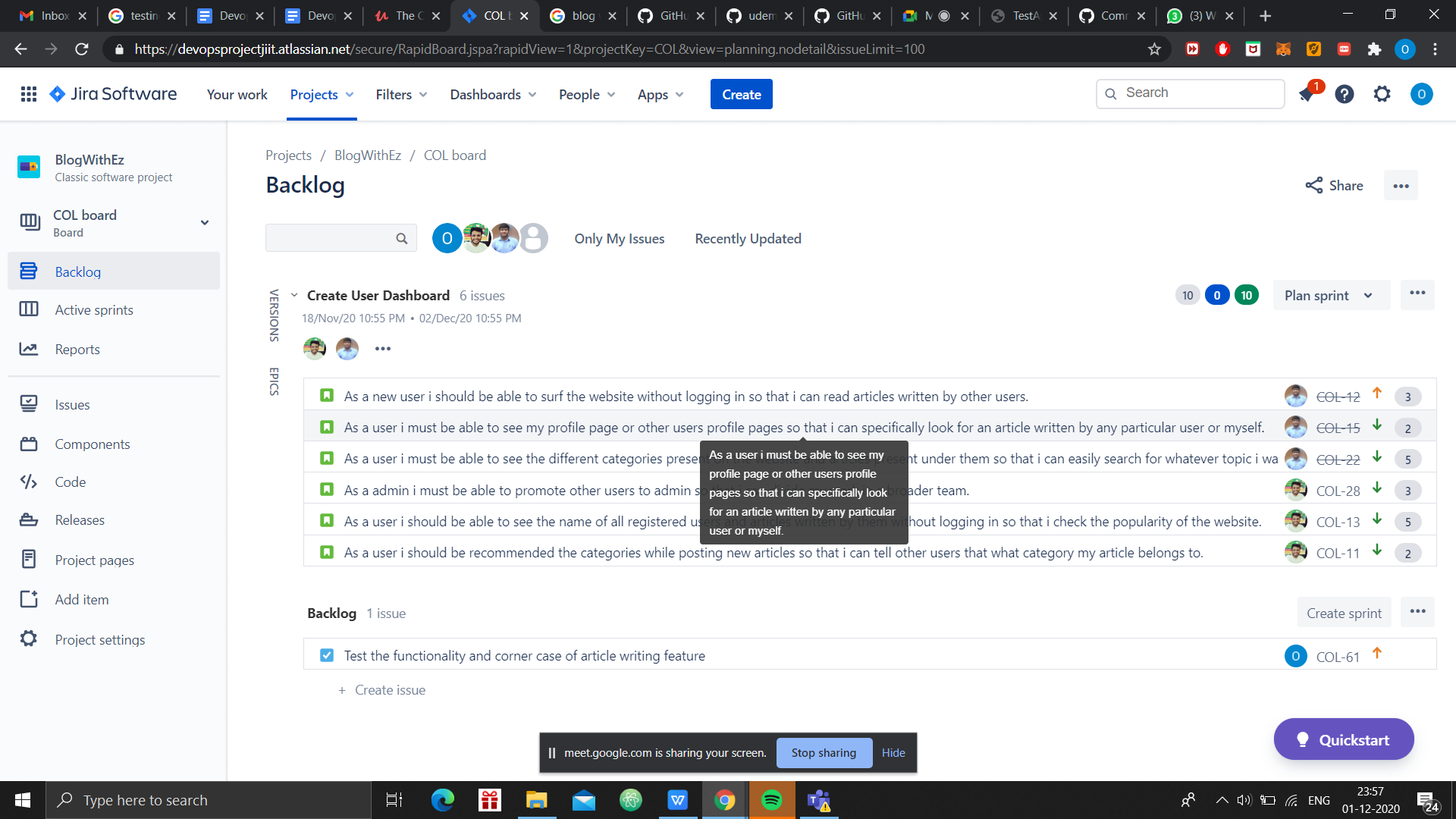
* **Estimation using Planning Poker:**

****

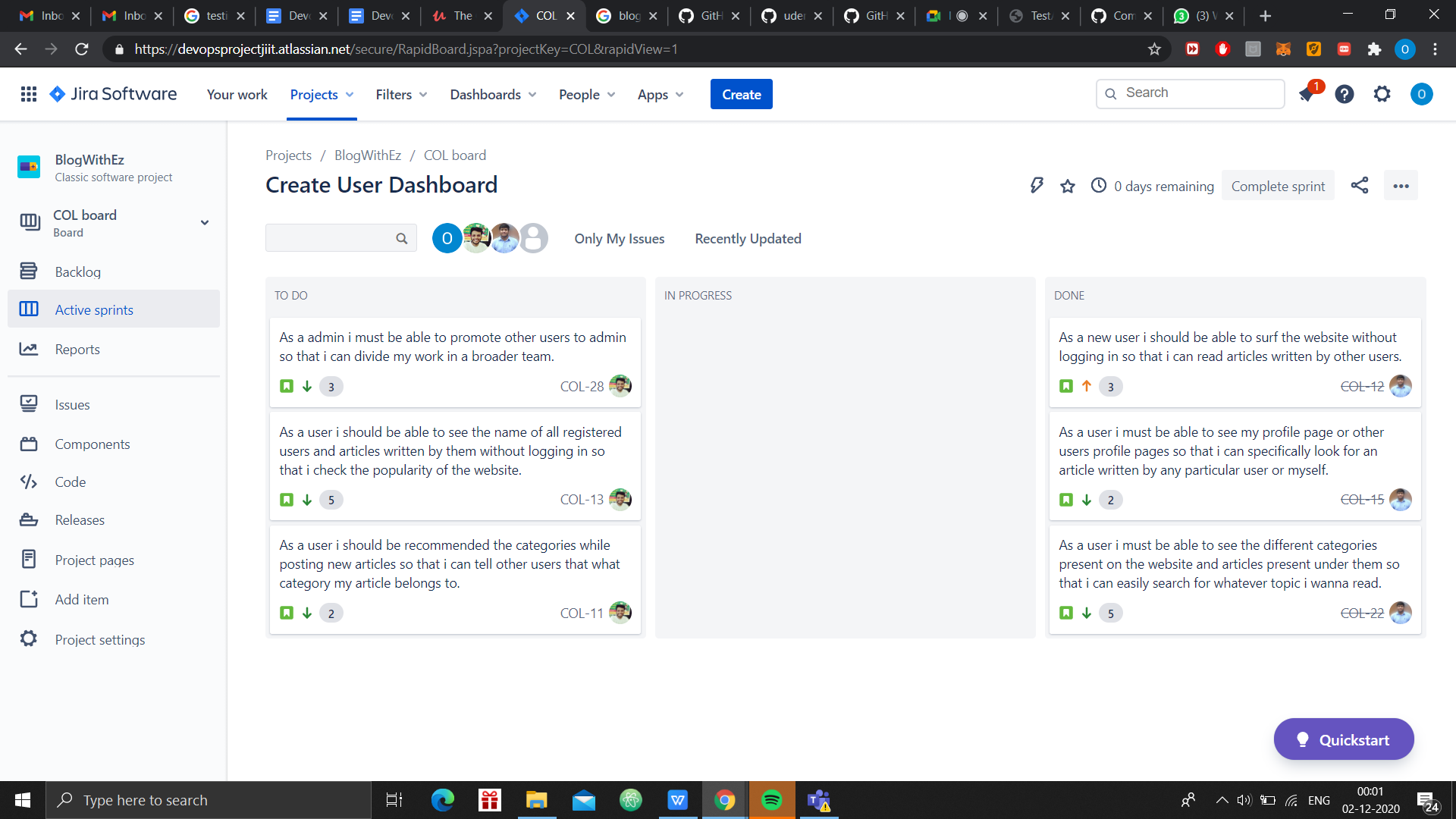
****

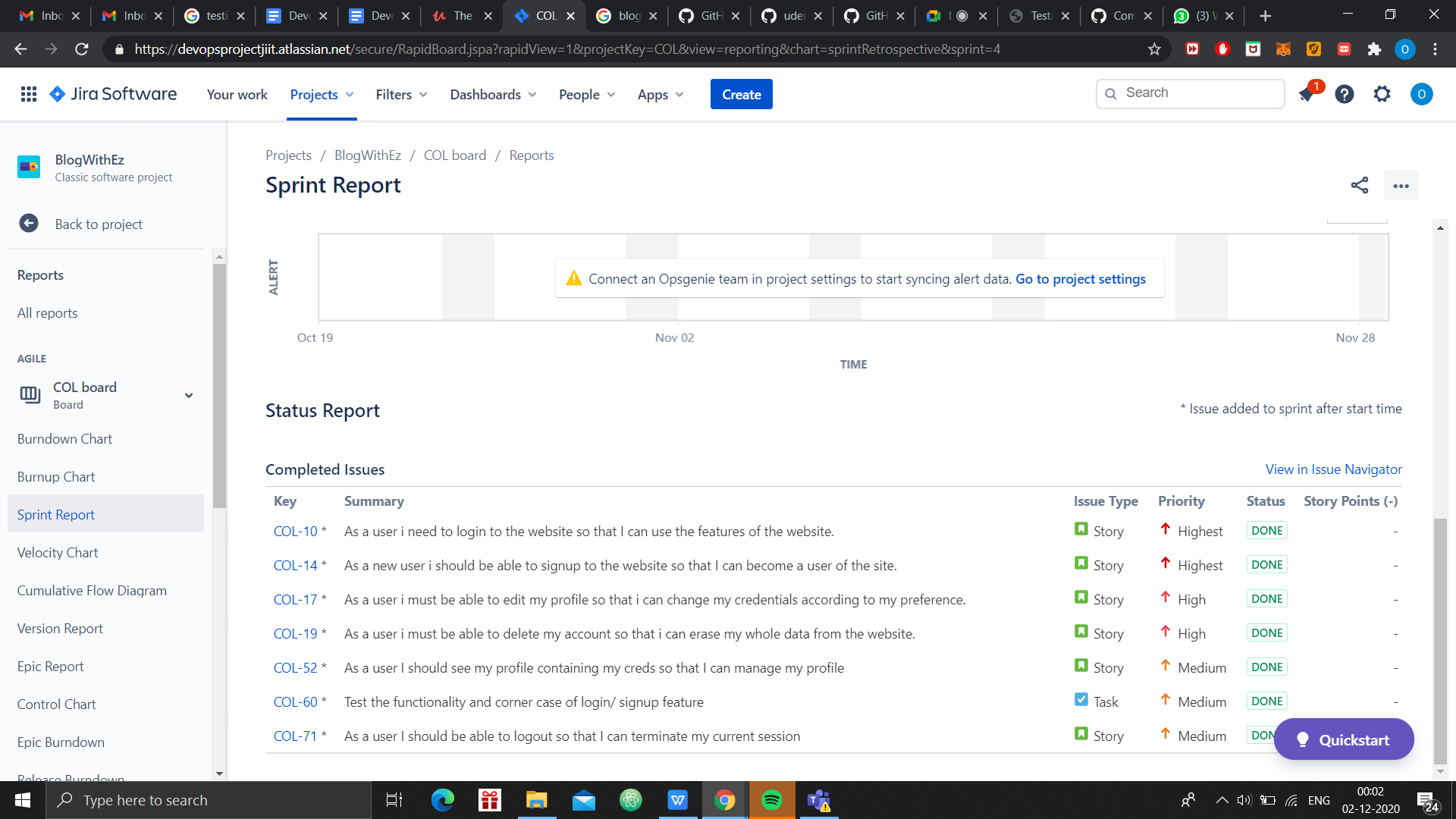
****

* **BackLog:**

****

* **Sprints**

****

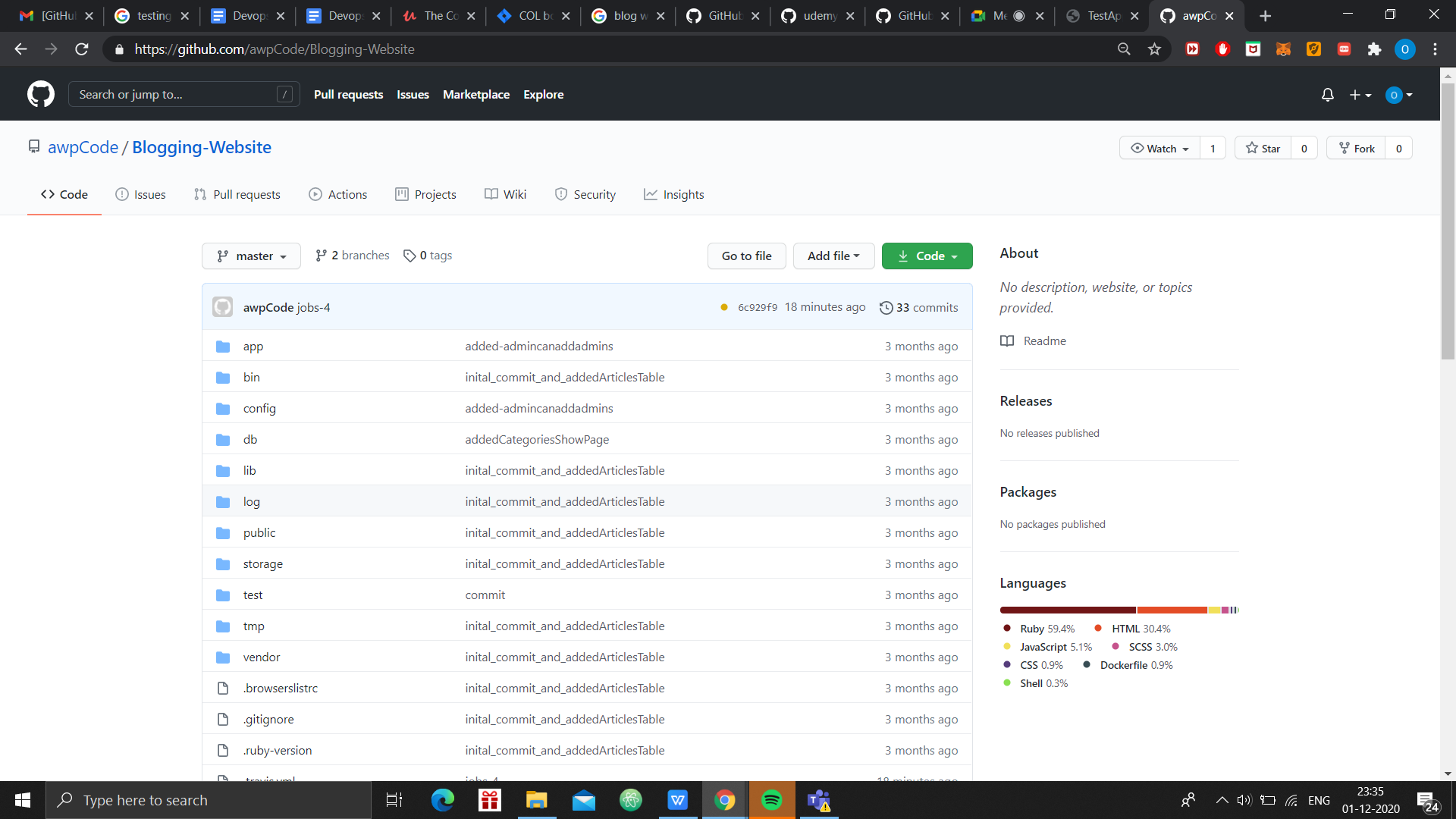
****

1. Version Control System-Git

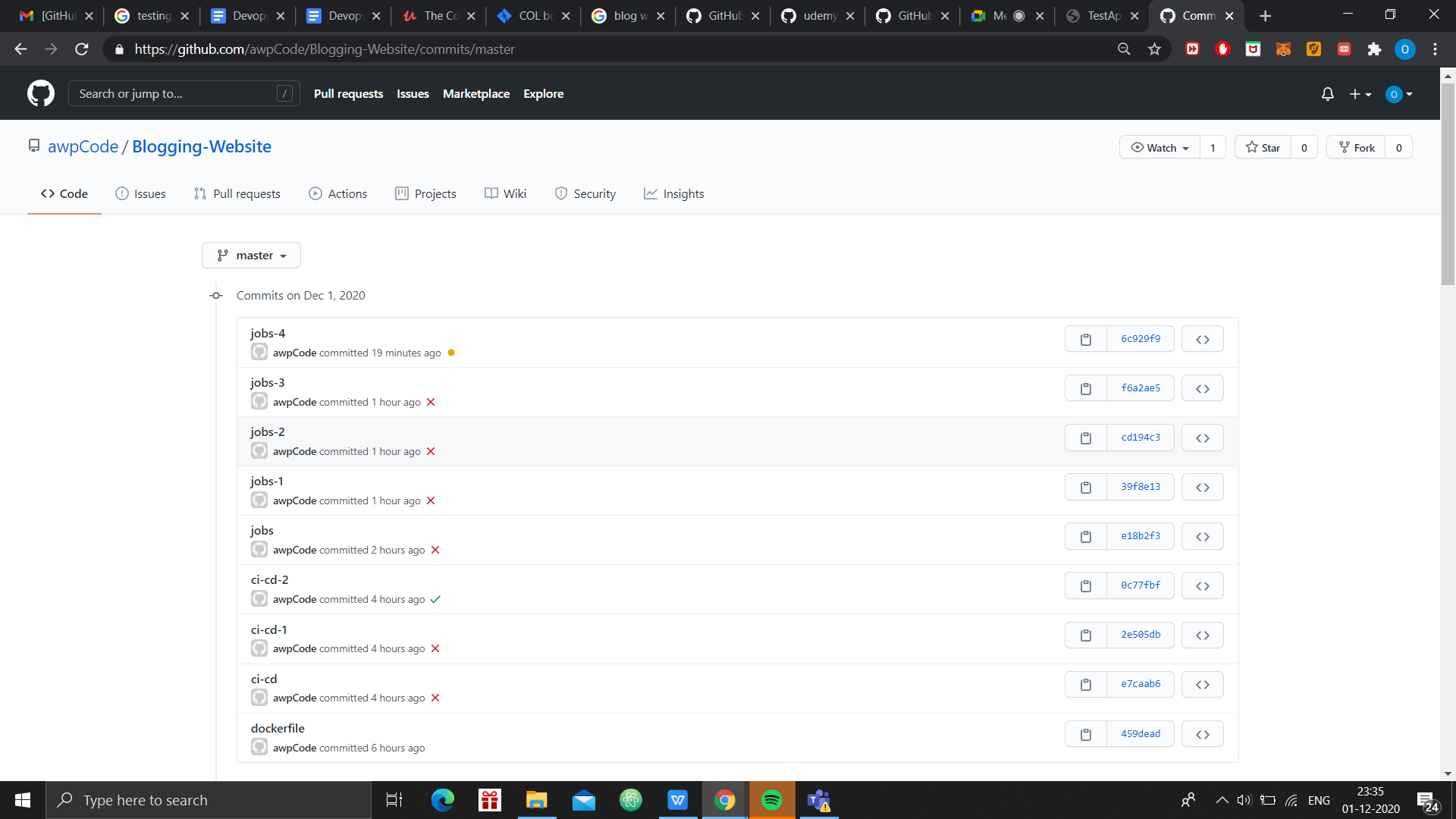
The term Version control refers to a system that records changes to a file or set of files over time called the ‘versions’. In other words, these versions help us in tracking the changes in your codes/project and if necessary, undo those changes as well. This feature of being able to compare, differentiate and revert changes between two versions of a particular project becomes really helpful when working on a larger project.Larger projects mean more people working on the same code which increases the chances of conflicts. Using version control we can easily prevent these conflicts.Git is a version control system which tracks the changes when working with computer codes while GitHub is a Web-based Git version control repository hosting service. It provides all of the distributed version control and source code management (SCM) functionalities of Git while topping it with a few of its own features.

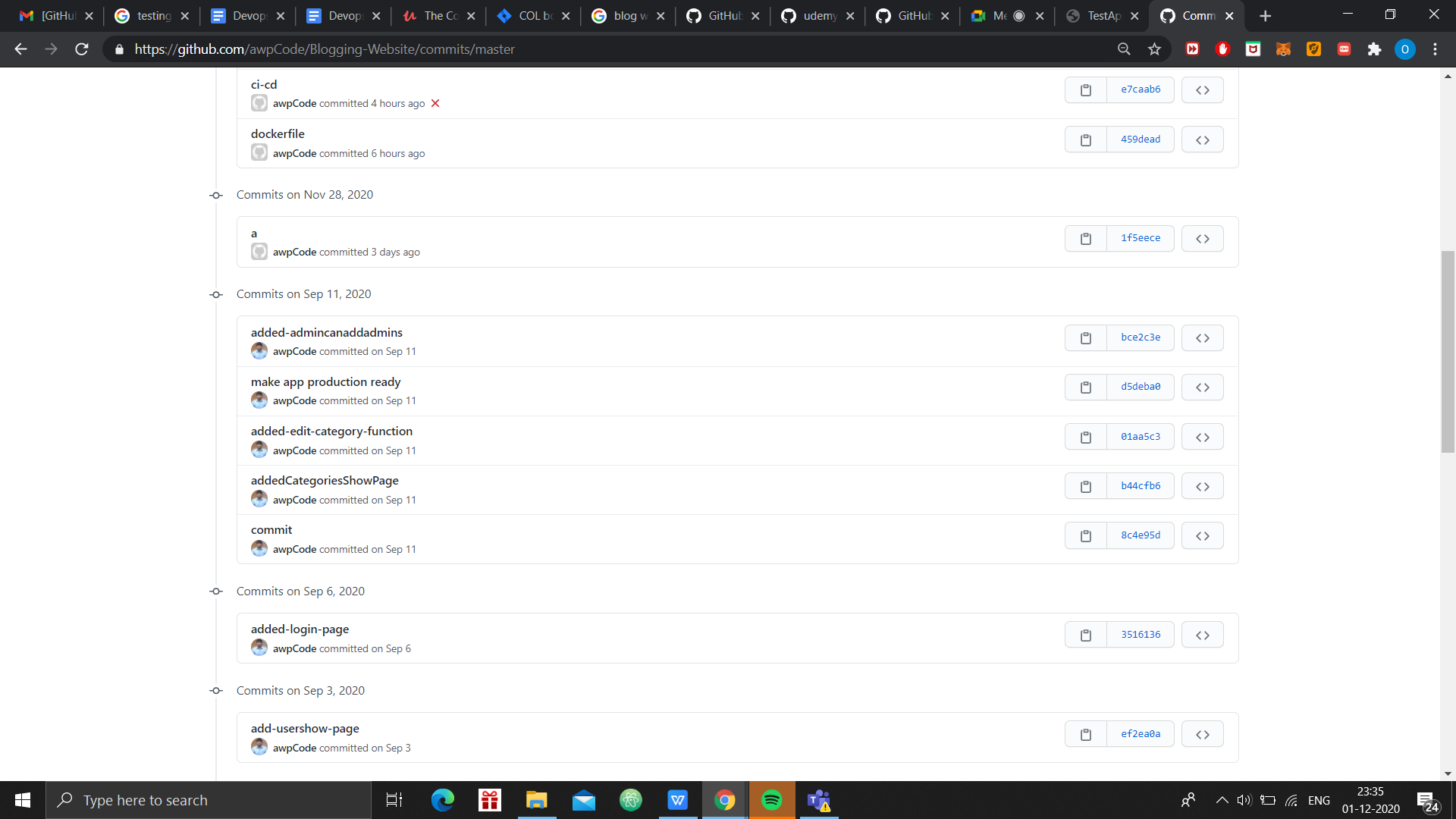
**Project Repository Link** [**Click Here**](https://github.com/awpCode/Blogging-Website)

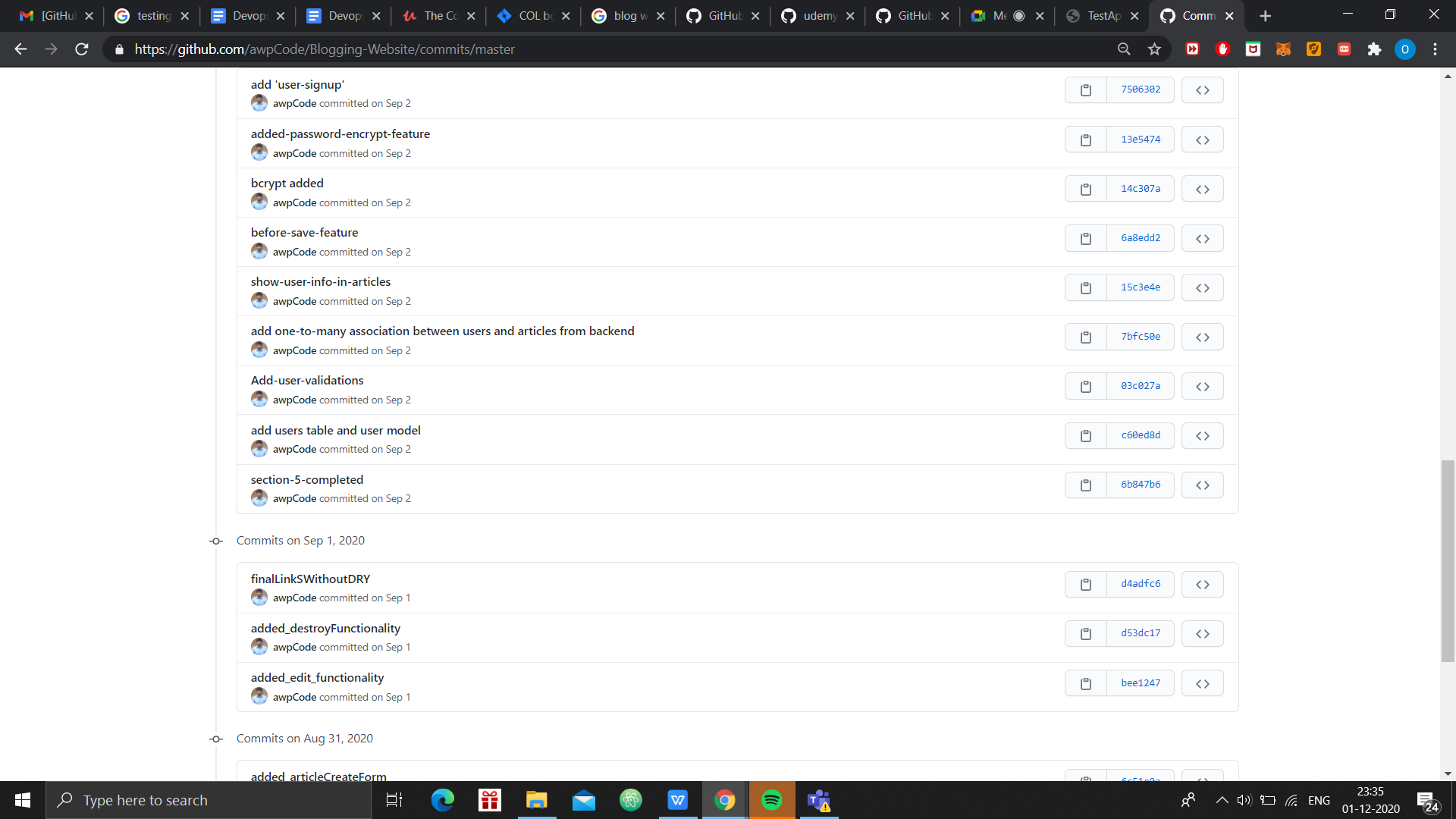
**Github Repository:**

****

**Commit History**

****

****

****

1. CI/CD PIPELINE- Travis CI

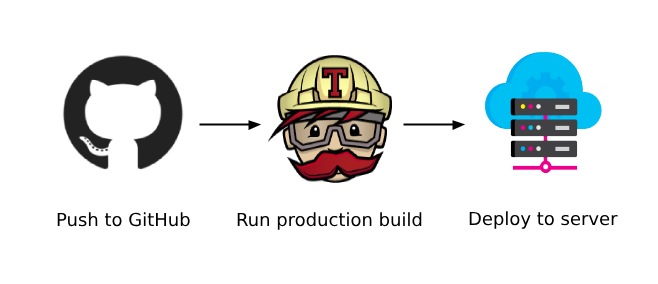
In software engineering, CI/CD or CICD generally refers to the combined practices of continuous integration and either continuous delivery or continuous deployment. CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications.

Travis CI is a hosted [continuous integration](https://en.wikipedia.org/wiki/Continuous_integration) service used to build and test software projects hosted at [GitHub](https://en.wikipedia.org/wiki/GitHub) and [Bitbucket](https://en.wikipedia.org/wiki/Bitbucket).

Travis CI provides various paid plans for private projects, and a free plan for open source. TravisPro provides custom deployments of a proprietary version on the customer's own hardware.

The source is technically [free software](https://en.wikipedia.org/wiki/Free_software) and available piecemeal on GitHub under permissive licenses. The company notes, however, that the large number of tasks that a user needs to monitor and perform can make it difficult for some users to successfully integrate the Enterprise version with their own infrastructure.

Travis CI enables your team to test and ship your apps with confidence. Easily sync your projects with Travis CI and you'll be testing your code in minutes.

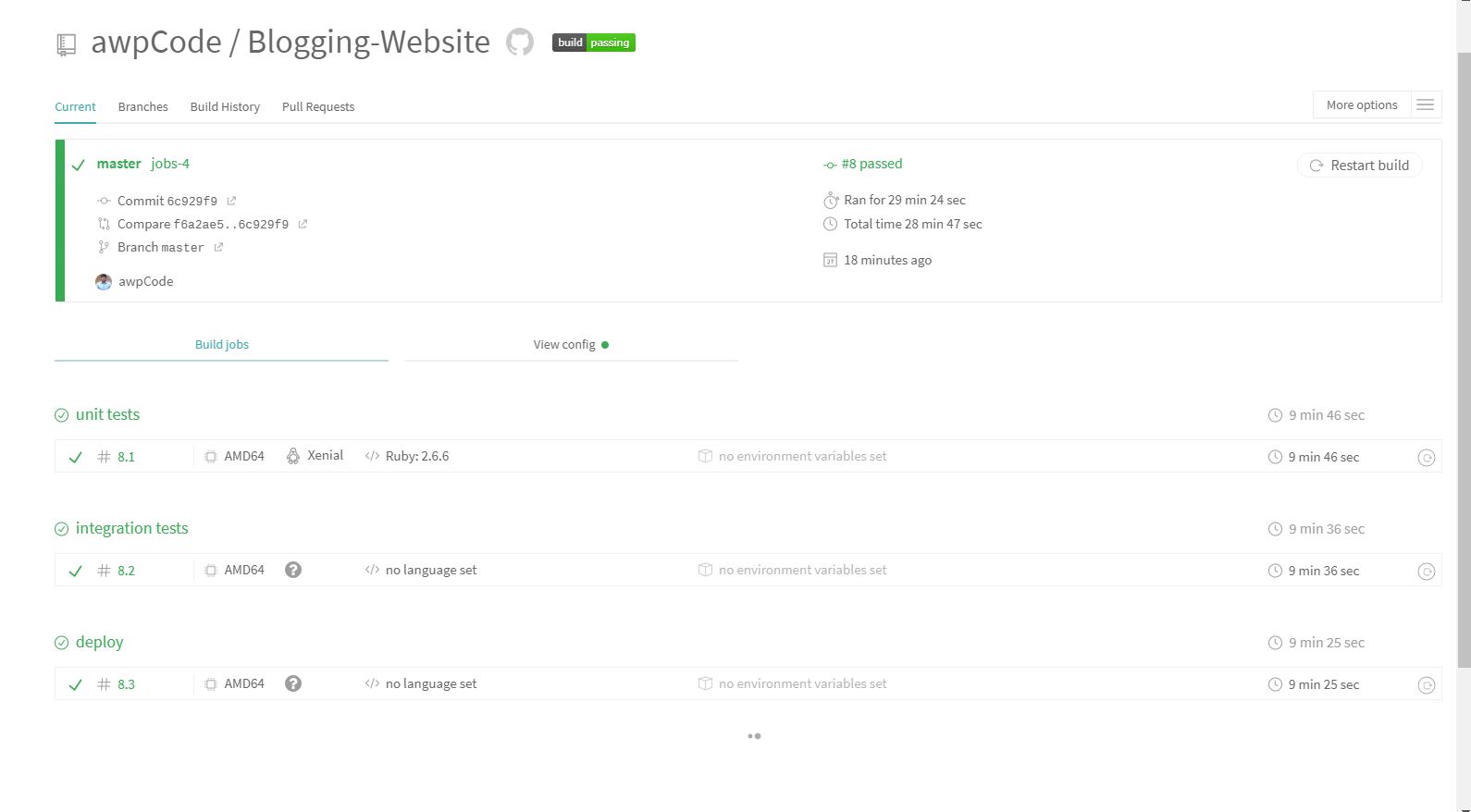


## **CI Builds and Automation: Building, Testing, Deploying**

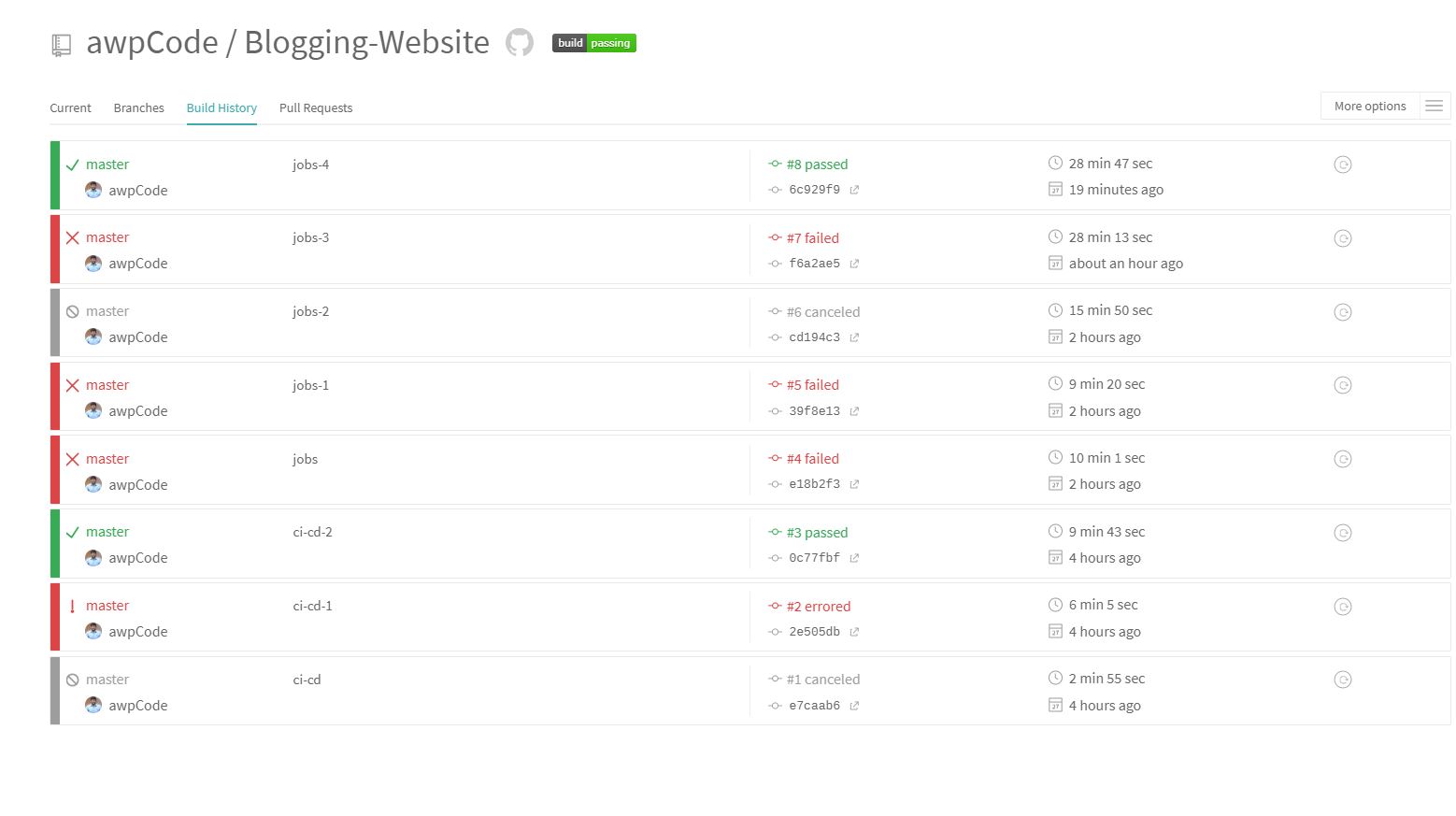
When you run a build, Travis CI clones your GitHub repository into a brand-new virtual environment, and carries out a series of tasks to build and test your code. If one or more of those tasks fail, the build is considered [broken](https://docs.travis-ci.com/user/for-beginners/#breaking-the-build). If none of the tasks fail, the build is considered [passed](https://docs.travis-ci.com/user/for-beginners/#breaking-the-build) and Travis CI can deploy your code to a web server or application host.

CI builds can also automate other parts of your delivery workflow. This means you can have jobs depend on each other with [Build Stages](https://docs.travis-ci.com/user/build-stages/), set up [notifications](https://docs.travis-ci.com/user/notifications/), prepare [deployments](https://docs.travis-ci.com/user/deployment/) after builds and many other tasks.

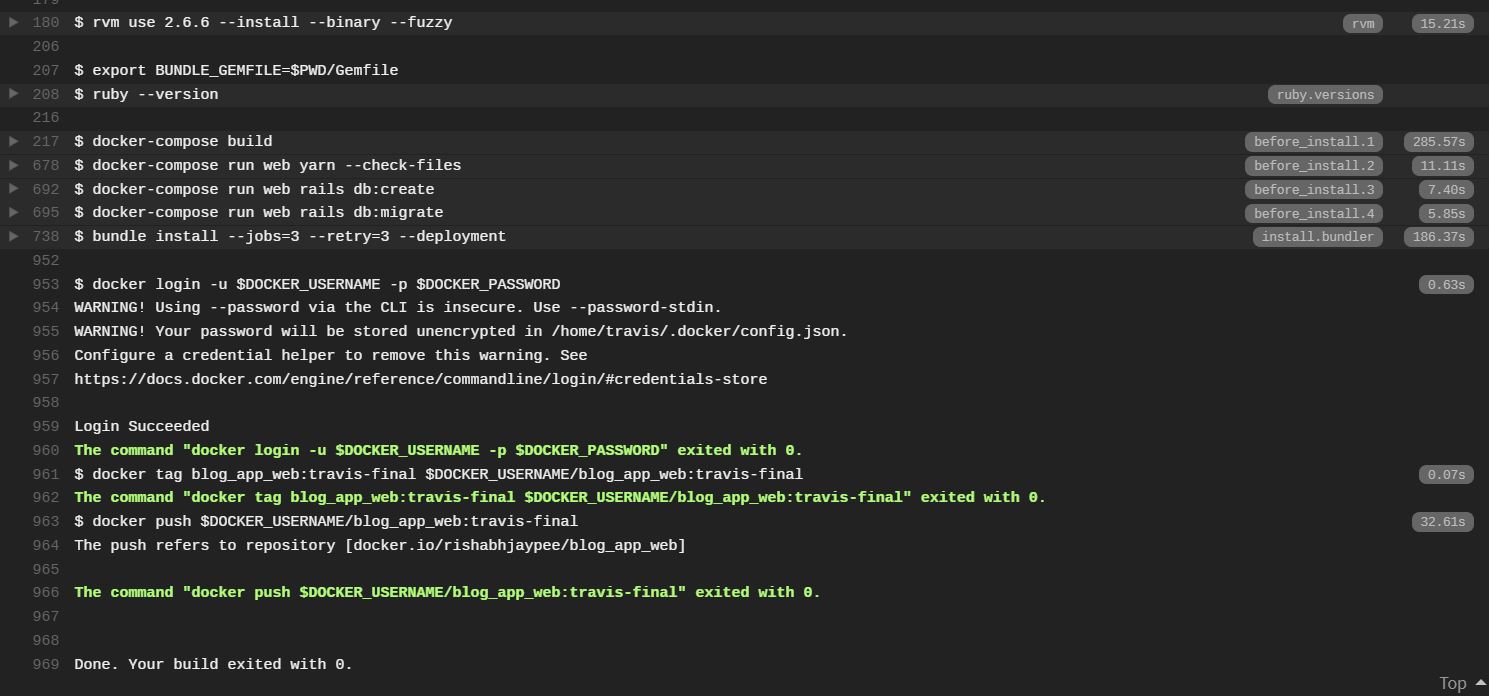
**TRAVIS CI DASHBOARD**

****

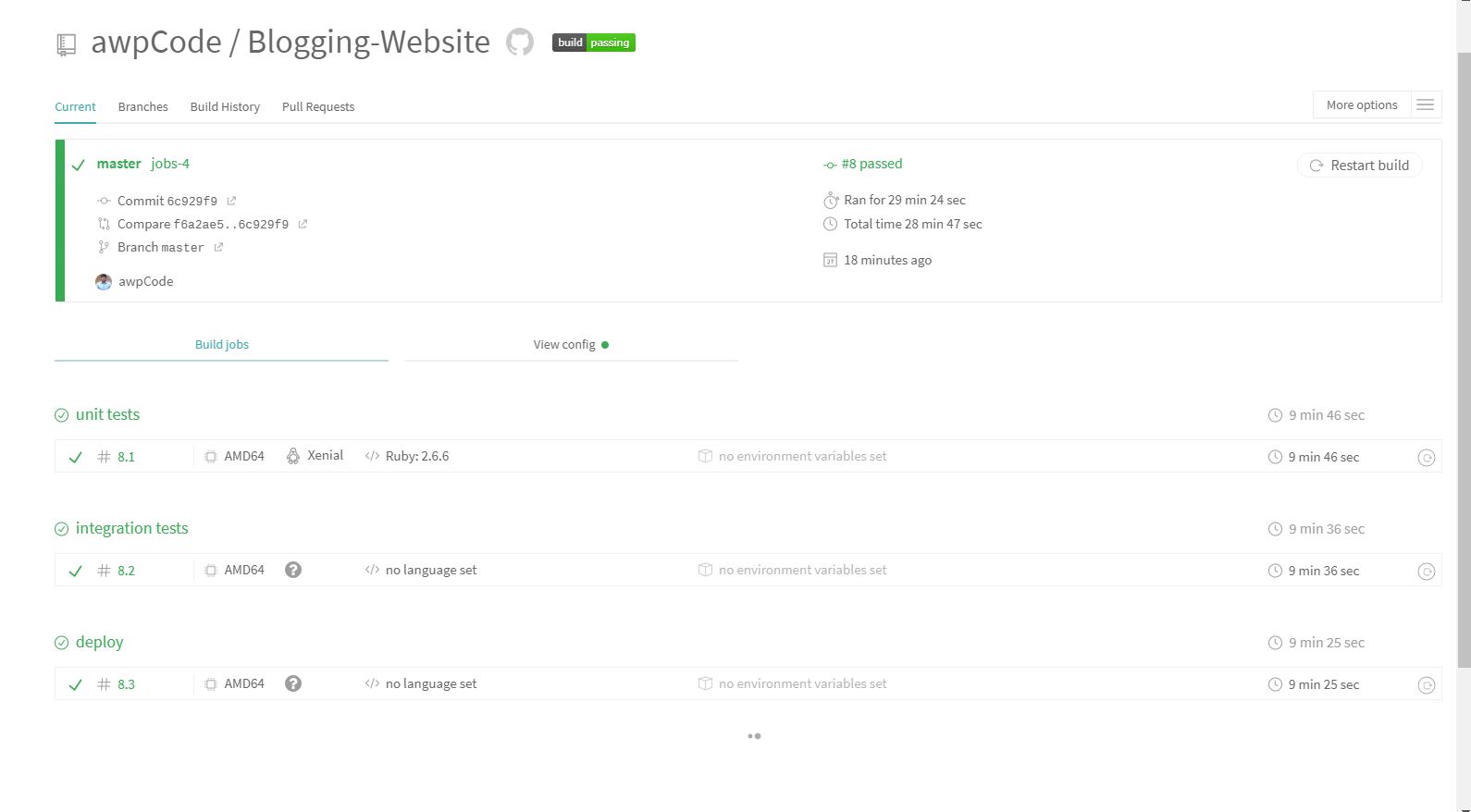
**PROJECT BUILD HISTORY**

****

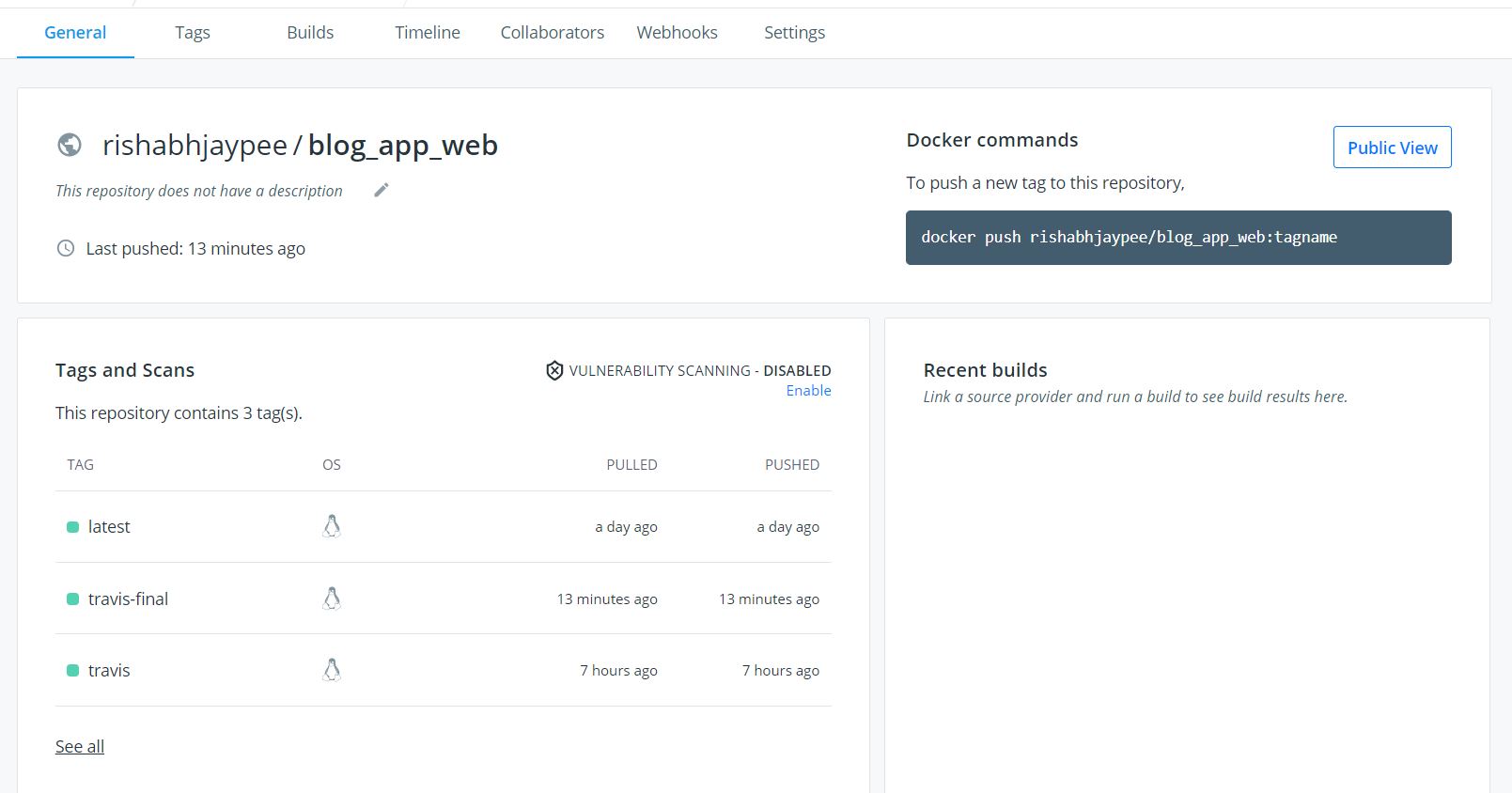
**CONSOLE OUTPUT**

****

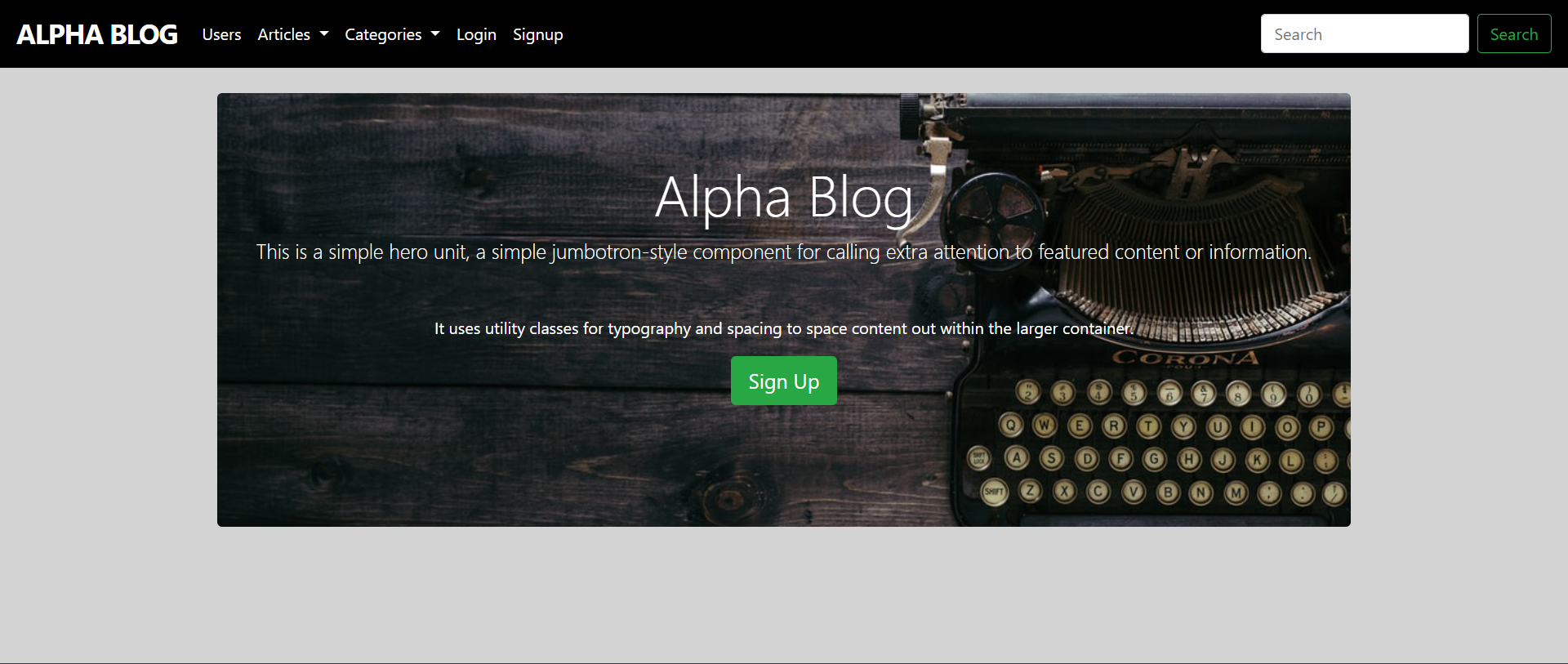
**Travis Pipeline View:**

****

**Deploy on DockerHub**

****

**Output**

****

### **Conclusion:**

We can conclude by building this project that a very pivotal element in adopting DevOps is team empowerment which has to be developed in every individual, resulting in more motivated and trustworthy employees. Optimization and streamlining of process by creating pipelines to check the end to end flow from source code check in to deployment. Improved productivity is a factor but the most important one is saving time and money with efficient use of resources. At last, we can say that adopting DevOps help business start making lots of money with more quicker data. All businesses are nowadays more of data and cloud business thus providing more profit in addition to automation is the appropriate thing all organizations are craving for.

Thank You