ezkomment

# Orbital 2022 Proposal

Vu Van Dung Nguyen Viet Anh

14 March 2022

## Contents

1 Team Information 2

2 Motivation 2

3 Aim 4

4 User Stories 4

5 Features and Planned Timeline 4

6 Technical Information 5
- 6.1 Planned Tech Stack . . . 5
- 6.2 Qualifications . . . 6
  - 6.2.1 Vu Van Dung . . . 6
  - 6.2.2 Nguyen Viet Anh . . . 6

## 1 Team Information

- **Application/team name:** EzKomment
- **Proposed level of achievement:** Artemis

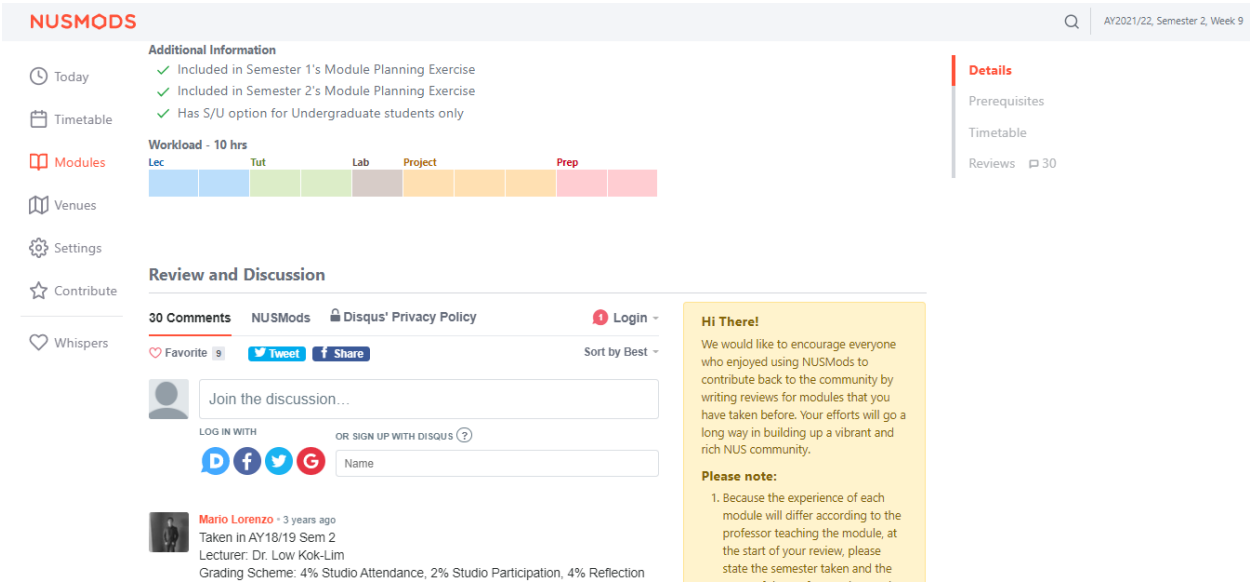
## 2 Motivation

Many programmers start learning web development by writing static sites. It can be for various purposes: self-introduction, product showcases or blogs. And we all want to get feedback on the products we built. However, since the sites we built, whether by pure HTML, with static generators such as Jekyll, or even more complex frameworks such as React, are all static, we cannot host

comments on it since they are dynamic contents and do require a backend with at least a database to store those contents.

I (Dung) was also once in the same position: I would like to add a form to the end of my blog so that readers can add comments to it, so that I (and other readers if I want to) can know what people think of my contents. However, I could only work on front-end development at the time, so I could not implement that feature. Best I could do was to do some tweaks on my Google account and the form such that whenever the reader submits the form, "I" send an email to myself. Of course it was not very practical: other readers cannot read those comments, and I would like to host the comments elsewhere so that my inbox does not get filled.

Many websites, for that reason and others, did not implement that, but instead opted to use more easy-to-use services. For example, NUSMods use Disqus. There is also fastfeedback.io (its main goal was not to provide a service but to teach a course, but it still does the work and can be used in any websites). However, there still remain many problems. Firstly, these services do not allow custom design for the form. Their own design is nice, but it is not customisable enough to be consistent with the main site design. For example, I do not think Disqus' design below can go along with NUSMods' design well. Some of the consistencies that anyone can notice with naked eye without inspecting the source, written by a person who has not even taken an official course in digital and UI/UX design:




- Fonts: NUSMods uses the standard system font stack (Segoe UI on Windows, San Francisco on macOS, etc.), while Disqus uses Helvetica Neue.
- Colours: While Disqus allows customisation of the primary colour (with the `--publisher-color` CSS variable), other colours do not have the same privilege. A naked eye can tell the body text colour of NUSMods and Disqus are completely different (NUSMods' is "greyer"). A source inspection shows NUSMods' body colour is #60707a, while Disqus uses all sorts of grey shades: #2a2e2e, #687a86, #657c7a, etc. Furthermore, the dropdown by Disqus does not follow `--publisher-color` at all, and still uses normal blue.
- Spacing: While NUSMods buttons are quite spacious and have large paddings, Disqus buttons

(such as the social share buttons or dropdown items) are narrower.

- Borders: NUSMods' borders are normally 1px wide and are pretty faded, Disqus borders are 2px wide and filled with a far stronger colour. etc. and etc.

Secondly, they are business services, so I need to pay to use them. While I can understand the reason behind that, up to ten comments per day for a project of not more than 100 pages should not cost me that much.

As a person making static sites myself, I would like a service which is customisable, fully-featured and (for relatively small sites) free.

## 3 Aim

A product-as-a-service (PaaS) where users can add dynamic comments to their static sites. We plan to create a separate page for each website the user needs, then the user can simply add an `<iframe>` to that page to display the dynamic contents. In that way, even super basic single-file HTML sites can have comments.

## 4 User Stories

- As a site author, I want to be able to add comments to my site without typing a single line of code. I would like to register for any number of sites, and for each site I have a readily-available `<iframe>` tag generated (or at least an URL that I can use for my `<iframe>`).
- As a site author, I want to be able to completely customise the comment section if I already have a custom design system at hand.
- As a site author, I want to see all comments on my site in a dashboard, and I would like to approve and delete them. Only approved comments will show up in the public comment section on my site.
- As a commenter, I want to use fully-featured Markdown to style my comments. I also want an editor I can use to style my comments in the case I do not know any Markdown or I forget some Markdown syntaxes.

## 5 Features and Planned Timeline

We plan to add the following features for our product to stand out compared to existing services on the market. We expect to implement all of them by the end of June.

- Fully customisable. In other words, users have complete control (within the security limitation) over the HTML/CSS of the page.

- However, we also plan to provide a nice built-in design for these pages so that normal users can still have a decent comment section design.
- Dark mode support for both the app and the page for users to `<iframe>` to.
- Support for Markdown inside comments, with additional plugins such as mathematics (with KaTeX).
- A WYSIWYG editor for those who do not know Markdown.
- Ability for users to approve and delete comments to prevent spams and toxicity.

The following are less prioritised features. We plan to work on them right after the above features are completed, and expect to complete these (except the features that turn out to be too complicated or impractical) by the end of July.

- A mechanism to host images, whether by a third-party service or by our own hosting server, so that users can paste images on the comments directly without having to upload them elsewhere (e.g. on Imgur) manually.
- Voting or reactions on comments.
- Reply to comments. However we do not plan to set up more "levels" of comments or threads (like Reddit).
- Publish a package on npm so that users can fully use the service to the highest level of customizability. For example, React users can import a custom React component from that package instead of having to use `<iframe>`.
- Since we have been generously offered several discounts and free plans by the GitHub Education Pack, we plan to make it completely free and open-source for the time being.

# 6 Technical Information

## 6.1 Planned Tech Stack

1. **Frontend:** Next.js, React, Tailwind
2. **Backend:** Express/Spring
3. **Authentication, database and storage:** Firebase
4. **Source hosting and collaboration:** GitHub
5. **Frontend deployment:** Vercel
6. **Backend deployment:** DigitalOcean/Render

## 6.2 Qualifications

This section has been modified to remove personal information.

### 6.2.1 Vu Van Dung

- HTML, CSS, Sass and DOM manipulation with JavaScript.
- React, Next.js and Firebase: I can build and deploy full stack apps with Next.js with a React-based frontend and REST API routes combined with Firebase SDK as backend.
- Git and GitHub: I can use git for version control and use GitHub as a git remote for my code. I also know how to work with Dependabot, continuous integration (GitHub Actions) and continuous deployment (GitHub Actions, Vercel, Netlify, Heroku, DigitalOcean, …).
- JavaScript and TypeScript: I am comfortable with JavaScript and JSX, and can use TypeScript/TSX (and prefer using TS) in my projects.
- Express: I have a basic understanding of how an Express app works, and can help maintain such apps.
- Web concepts: I have a basic understanding of HTTP requests, responses and status codes, and can apply it to my apps.

### 6.2.2 Nguyen Viet Anh

- Java, JavaScript, Python, TypeScript, Haskell: I have experiences with these languages, and I am comfortable with using Java/TypeScript/Python in my projects.
- Fundamental web technologies and concepts: I have basic knowledge about HTML, CSS and DOM manipulation with JavaScript. I also have a basic understanding about HTTP requests, responses and status codes.
- React, Express and Firebase: I can build web applications using React-based frontend with Express and Firebase SDK as backend.
- Spring: I am highly proficient with Java, and can use the Spring Framework for development combined with Junit for unit testing.
- Python: I mostly use Python to write automated scripts and its third libraries to implement extra features for my apps.
- Git and GitHub: I can use git for version control and working remotely. I also know how to work with GitHub Actions.
- Data Structures, Algorithms and Computer Theories: I have a solid understanding of multiple Data Structures, Algorithms and Computer Theories, and can apply suitable paradigm/tools to improve the performance of my apps.