



GSoD'22 statement of interest

Palisadoes

Project 2 - Create How-To Guides and Tutorials

Personal information

Name:

Course:

University:

Current Year:

Email:

Github: <https://github.com/vedant-jain03>

LinkedIn: <https://www.linkedin.com/in/vedant-jain-781006145/>

Phone:

Current Country:

Link to Resume:

Professional information

Skills

- Languages: C, C++, JavaScript, HTML5/CSS3, Java.
- Frameworks and Libraries: Node.js, Express.js, React.js, Socket.io, Ruby on Rails, Bootstrap
- Databases: MongoDB, PostgreSQL, SQL
- Developer Tools: Git, VS Code, Github
- Technical Writing Tools: Markdown, Github gist, google docs
- Core Skills: Data Structure and Algorithms, Problem Solving, Operating System, Computer Networks

Technical Writing Samples:

- Some of my public repositories and files:
 1. [Bit Manipulation](#)
 2. [DBMS Cheat Sheet](#)
 3. [Quick cheat sheet for Backend\(Node.js + Express.js\)](#)
- [Dev Blogs](#)
- Open-source documentation contribution
 1. Made [CONTRIBUTING.md](#) [PR] for the main repository of CircuitVerse[GSoC] organization.

Open source contribution:

- CircuitVerse[GSoC, GSoD] organisation:
 1. CircuitVerse Repository[Rails and Javascript]: [PRs](#).
 2. Interactive-Book Repository[Markdown / Jekyll]: [PRs](#).
- Girlsscript Summer of Code 2021: [PRs](#).
- I am mentoring an open-source contributor in Girls-script Summer of Codes 2022.
- I participated as a Maintainer in HacktoberFest 21.

Project Statement

Project Title:

Create How-To Guides and Tutorials for getting started with Talawa, Talawa API, and Talawa-Admin

Mentors:

Shannika Jackson, Dominic Mills, Michael Lue

Abstract:

User documentation is designed to assist end-users to use a product or service. Good user documentation is very important because it provides an avenue for users to learn how to use software, its features, tips, and tricks, and also resolve common problems encountered when using the software. It also reduces support costs and is part of the corporate

identity of the product: good user documentation is a sign of the healthiness of the product, and the developer team.

Currently, there exists no truly detailed guide on how to get set up and become productive within the basic ecosystems of Talawa, Talawa-API, and Talawa-Admin. There are a number of resources available, such as videos demonstrating features from merged pull requests in the respective repositories as well as notes that need to be structured and organized in a meaningful way for a prospective user to be productive with the codebase.

Goals of the Project:

1. Improve documentation readability by ensuring the language is accessible by individuals regardless of their language background.
2. Construct a guide and style format for new technical writers/contributors.
3. Teaching beginners how to document code optimally and effectively with best practices.

Repos to update:

- [Talawa](#)
- [Talawa-API](#)
- [Talawa-Admin](#)

Implementation:

Talawa uses [docusaurus](#) to build the [talawa-docs](#) website.

File structure of docs-repository that need to be updated:

```
PalisadoesFoundation / talawa-docs /
├── docs/
...
| └── talawa-admin/                # talawa-admin section
|   └── introduction.md           # markdown file for introduction page
|   └── talawa-api/              # talawa-api section
|     └── constraints/           # base-constraints folder
|       ├── admin.md/           # base-constraints / admin page
|       ├── organization.md/    # base-constraints/organization page
|       └── user.md/            # base-constraints/ admin page
|     └── api-overview.md/      # talawa-api overview page md file
```

```

| | └─ authentication.md/           # Authentication Page md file
| | └─ expectations.md/           # Expectations Page md File
| | └─ expectations.md/           # Expectations Page
| └─ talawa/                       # talawa section
| | └─ developer-environment.md   # developer page md file
| | └─ introduction.md           # introduction page md file
...
└─ src/
| └─ css/                         # css for docs site folder
| | └─ custom.css                 # css file for custom themes
| └─ pages/                       # folder for index.js
| └─ Talawa (1).drawio
└─ static/                         # folder for static files
...
# files for configuration

```

For each [markdown](#) file in docs folder, we need to give id and title(display text). I have good knowledge of markdown as I have done some previous contributions based on it.

Video tutorials can help developers to visualize and compare the results. For every moderate and advanced topic, I will record the video and will add it to the markdown page.

The language that I will be using is English and for every video tutorial, I will be using youtube for it.

For the technical writer's guide section we can include how to comment on a function or classes so that to successfully executed on CI/CD pipeline that we use for automating the documents.

The structure of the docs for this project scope should be as follows:

1. For Talawa

Talawa /

- Overview[Modified]: A brief description of Talawa App. Scope, downloads, core features, video, architecture, and mentors can be possible headings for this page.
- Setup[New]
 - a. Windows
 - b. Linux
 - c. macOS
- Contributors Guide[New]

- Technical Writers Guide[New]
- Usability (Video Tutorials)[New]

2. For Talawa-API

Talawa API /

- Overview[Modified]: This section can be found [here](#) but it needs a more detailed description or overview of the product usability.
- Setup[New]
 - a. Windows
 - b. Linux
 - c. macOS
- Contributors Guide[New]
- Technical Writers Guide[New]
- Base constraints [Modified]: This can be found [here](#) but every page can have video tutorials and a more detailed description of the respective endpoints of the API.
 1. User
 2. Organization
 3. Admin
- Authentication [Modified]
- Functionality[Modified]: Each endpoint should have a description and could have video tutorials or GIFs for the reference
- Expectation
- Advanced[New]: It can have an advanced configuration for the various endpoints to get relatable data.
- Usability(Video Tutorials)[New]

3. For Talawa-Admin

Talawa Admin /

- Overview[Modified]: A brief description of Talawa Admin Software. Scope, downloads, core features, video, architecture, and mentors can be possible headings for this page.
- Setup[New]
 - a. Windows.

- b. Linux
 - c. macOS
- Technical Writers Guide[New]
- Contributors Guide[New]
- Advanced[New]: This section can be helpful for the advanced configurations.
- Usability (Video Tutorials)[New]

Project Timeline

The project can be divided into three phases, one month for each phase.

- First month - Talawa Repository
- Second Month - Talawa-API repository
- Third Month - Talawa-Admin Repository

Detailed Timeline:

- **Phase 1: Documenting Talawa Repository**
 1. **Information gathering - 8 days (Week 1, 2):**
 - a. I will be learning about talawa and simultaneously keeping notes for every step so that new beginners can get an overview of the process from the documentation.
 - b. I will gather information about features, and usability under the guidance of mentors.
 - c. Setup
 2. **Doc writing Period - 14 days (Week 2, 3):**
 - a. I will start documenting in the order I mentioned in the proposed structure.
 - b. This period includes a recording of each tutorial's video.
 - c. I will make the Draft-PR for review.
 3. **Review - 5 days (Week 3):**
 - a. I will work on the requested changes.
 - b. Will commit the new changes and get it final reviewed.
 4. **First Month Evaluation - 3 days (Week 4):**
 - a. First Month Evaluation Period.
- **Phase 2: Documenting Talawa-API Repository**
 1. **Information gathering - 7 days (Week 1):**

- a. I will be learning about talawa-API and simultaneously testing its various endpoints to get a quick overview of its functioning.
 - b. I will gather information about features, and usability under the guidance of mentors.
 - c. Setup
 - 2. **Doc writing Period - 14 days (Week 2, 3):**
 - a. I will start writing the docs in the order I mentioned in the proposed structure for talawa-API.
 - b. This period includes a recording of each tutorial's video
 - c. I will make the PR for review.
 - 3. **Review - 5 days (Week 3, 4):**
 - a. I will work on the requested changes.
 - b. Will commit the new changes and get it final reviewed.
 - 4. **Second Month Evaluation - 3 days (Week 4):**
 - a. Second Month Evaluation Period.
- **Phase 3: Documenting Talawa-Admin Repository**
- 1. **Information gathering - 7 days (Week 1):**
 - a. I will be learning about talawa-Admin and simultaneously testing its various features to get a quick overview of its functioning.
 - b. I will gather information about features, and usability under the guidance of mentors.
 - c. Setup
 - 2. **Doc writing Period - 14 days (Week 2, 3):**
 - a. I will start writing the docs in the order I mentioned in the proposed structure for talawa-Admin.
 - b. This period includes a recording of each tutorial's video
 - c. I will make the PR for review.
 - 3. **Review - 5 days (Week 3, 4):**
 - a. I will work on the requested changes.
 - b. Will commit the new changes and get it final reviewed.
 - 4. **Third Month Evaluation - 5 days (Week 4):**
 - a. Third Month Evaluation Period.

Timeline weeks may vary according to the guidance of the mentors and OA.

Additional Information

OS: Dual Boot (Ubuntu and Windows)

Code editor: VS code

Working Hours

I can devote 25-30 hours per week time.

Why I am the best person to complete this Project?

- I have a good understanding of the technology that is required in the project which is javascript, and Node.js. As a result, it will not take much time for me to understand the code and functionality of the product.
- I have contributed to open source and used git and Github as a medium.
- I have built my personal projects on the top of the MERN stack.
- I have a good understanding of markdown files.
- I am passionate about technology, and learning and documenting it will help me to boost my skills as well.
- I can complete the doc under the given timeline.