

Assignment 03

Documentation and Website Development for Open Source Software

Objective

Develop comprehensive project documentation and a professional website for your open-source software project. This assignment emphasizes creating online-hosted documentation and a fully functional website, including community interaction channels.

Part 1: Documentation Creation Using ReadTheDocs

Tasks

1. Set Up Documentation:

- Use the **ReadTheDocs** framework on your local machine.
- Choose the **Sphinx theme** for your documentation.

2. Create the Following Pages :

- **About the Project:** Provide a comprehensive overview of your project, including its purpose and objectives.
- **Getting Started:** Include installation instructions, prerequisites, and setup guidance.
- **How to Use:** Offer detailed usage instructions and examples.
- **Technical Overview:** Summarize the project's architecture, core components, and underlying technologies.
- **API Reference:** Detail available APIs, including endpoints, request/response formats, and usage examples.
- **Configuration Guide:** Explain configuration options, default settings, and customization methods.
- **Maintenance and Troubleshooting:** Provide procedures for routine maintenance, common issues, and their resolutions.
- **Contribution Guidelines:** Outline the process for contributing to the project, including coding standards and submission procedures.
- **FAQ:** Address common queries and concerns.

- **Release Notes:** Document the history of releases, highlighting new features, improvements, and bug fixes.

3. Upload Documentation:

- Commit the documentation into the "**docs**" folder in the main branch of your GitHub repository.

4. Connect Hosting Services:

- Link the GitHub "docs" folder to ReadTheDocs hosting.
 - Add the hosting link to the project's **README** file.
-

Part 2: Website Development Using GitHub Pages and Jekyll

Tasks

1. Theme Selection:

- Choose an appropriate theme from Jekyll's collection.

2. Website Structure:

- **Homepage:**

- Include a compelling mission statement.
 - Provide direct links to the live documentation and GitHub repository.
 - Use design inspiration from [Apache Hadoop](#).

- **Feature Showcase:**

- Highlight the project's key functionalities and capabilities.

- **Community Section:**

- Create a **Community Page** similar to [Django Community](#).

- Include the following:

- **Discussion Forums:** Create a section linking to forums for user and developer discussions (e.g., GitHub Discussions or external platforms like Discourse).
 - **Mailing Lists:** Establish mailing lists for announcements and support.
 - **Other Communication Channels:** Provide links to chat platforms like Slack, Discord, or Gitter.

- **Contact Information:**

- Provide straightforward methods for visitors to reach out (e.g., email, contact form, or social links).

3. Design and Style:

- Adhere to the provided structure while exercising creative freedom in design and styling.

4. Integration:

- Ensure the website includes live links to your documentation, GitHub repository, and discussion forums.
-

Deliverables

1. Documentation:

- A link to your ReadTheDocs-hosted documentation.

2. Website:

- A GitHub Pages link to your project's website.
-

Notes

1. **Extra Points Opportunity:** Students who create a new custom theme for Jekyll will receive extra points for their innovation and effort.
-

Submission Instructions

- Submit both the documentation and website links.
- Ensure all deliverables are finalized and uploaded to the repository before submission.