

Chapter-1 Introduction to project	1
1.1 Overview	1
1.2 Objectives	2
1.3 Advantages	2
1.4 Technology Used	3
Chapter 2 Feasibility Study	5
2.1 Technical Feasibility	5
2.2 Economical Feasibility	5
2.3 Operational Feasibility	6
2.4 Schedule Feasibility	6
Chapter 3 Requirements	7
3.1 Software Requirements	7
3.2 Hardware Requirements	7
Chapter 4 Bibliography	8

Chapter-1 Introduction to project

1.1 Overview

The world calls for efficient programmers who can write a seamless amount of clean and optimized code in the given time-frame. For writing an optimized and error free code, it is very necessary for a programmer to choose among the best code editors. We have quite a few impressive ones out there in the market for open-source peeps like atom, sublime, etc. All of them help in modifying and saving our content in a pretty simple yet efficient manner. Then comes the task of management. In the open source world, source code is meant to be a collaborative effort, where programmers improve upon the source code and share the changes within the community.

Github is one of the best version control repository and Internet hosting service. It offers all of the distributed version control and source code management functionalities such as wikis, task management, and bug tracking and feature requests for every project. Our modifications can be easily uploaded by committing or pushing the files to github. Anyone can review, inspect, modify and enhance your code. These two things are staple in almost every open-source programmer's life. Edithub further simplifies this two-step process. This high-performance code editor for the web aims at making a programmer's life lot easier but combining these two tasks. It is a fascinating, vivacious developer-friendly editor. Inspired by the way you work, you can fetch, edit and push back your repository to GitHub.

It is easy to use for beginners and highly powerful for advanced users. It comes with syntax highlighting for many languages including PHP, JavaScript, HTML, and CSS. It comes with a very intuitive user interface that makes it super easy to browse files and work on projects. Instead of first cloning or downloading the files, updating the code for bug fix issues, etc. and then pushing it back to your github account, it does all this work in one-go. Once authenticated, you can upload files either from your local machine or via your GitHub account.

After uploading the files, you can open a specific project and view all the files simultaneously in different tabs. The beginner friendly interface has a black screen that supports distraction free editing mode. The turbolinks make the navigation quick, smooth and easy.

This feature-rich editor provides lightweight fuzzy-search library, highlight matching parenthesis and live syntax checker. It provides GitHub omni Authentication, save and push support. It has a drag and drop text using the mouse. It makes it really tough to write a messy code due to its auto-indentation feature. The line-wrapping and code-folding enhance the readability of the code. It

enables you to see the preview update of your HTML documents live in a different browser window. This compact editor can handle large files upto the size of 4 million lines.

1.2 Objectives

- **High Performance editor:** This lightweight editor launches quickly and gives rich editing features like automatic indentation, highlighting matching parentheses, line wrapping, drag and drop text using the mouse, code folding, capability to handle huge documents etc.
- **Smart Syntax Highlighting:** This feature displays text, especially source code, in different colors according to the category of terms for many languages like javascript, ruby on rails, HTML, CSS, etc.. It facilitates writing in a structured language such that both structures and syntax errors are visually distinct. It provides a strategy to improve the readability and context of the text; especially for code that spans several pages. The reader can easily ignore large sections of comments or code, depending on what they are looking for.
- **Save and push support:** It allows the user to save the changes made in the files in his/her local machine easily. It also gives you backup by pushing it to GitHub in a simple two-step process the instant you're done modifying your code. This gives you the modified file at both, GitHub and your local machine, making it hard to loose the data.
- **Fuzzy lightweight search:** It provides the technique of finding strings that match a pattern approximately (rather than exactly). The problem of approximate string matching is typically divided into two subproblems: finding approximate substring matches inside a given string and finding dictionary strings that match the pattern approximately.
- **User Authentication:** Allowing users to login with multiple authentication providers brings great benefits but also results in some annoying edge cases. For example, when they login with one provider, logout and then login with another. This is managing session data since a logged in user is simply a person who has some session data confirming that they have been logged in. The OmniAuth callback which a provider will redirect to upon authenticating a user it is created to be powerful, flexible, and do as little as possible. We used omniAuth to authenticate users via disparate systems.

1.3 Advantages

The proposed system has various advantages:

1. Proposed system has fully Responsive UI. Moreover the Design is based on Atomic Theory which make it more concise and easy to understand

2. This system has two types of authentication. One is based on Github and another one is built in app authentication
3. User can either Upload a Repository or Can fetch a repository from GitHub.
4. Editor is fully developer Friendly with features like Syntax Highlighting, Auto Indentation and outdentation and many more.
5. User can search for file or folder using an Approximation Search which is also called as Fuzzy Search.
6. User can view their HTML file in browser by just pressing on single link.
7. Code Folding helps in wrapping the block.
8. Live Syntax checker augment da productiveness of developer.
9. Drag and Drop Text will ease the work for developer.
10. One can directly push content to github.

1.4 Technology Used

During Development of proposed system following technologies been used

1. **Ruby On Rails** - It is a web application development framework written in the Ruby language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. Experienced Rails developers also report that it makes web application development more fun. It makes the assumption that there is a best way to do things, and it's designed to encourage that way - and in some cases to discourage alternatives. It is an open source ruby framework for developing database-backed web applications, using the Model-View-Controller pattern.
2. **Javascript**- JavaScript is an easy-to-use object-based scripting language that can be embedded in the header of your web pages. This cross-platform and lightweight programming language can enhance the dynamics and interactive features of your page by allowing you to perform calculations, check forms, write interactive games, add special effects, customize graphics selections, create security passwords and more. It is not compiled but translated. The JavaScript Translator (embedded in browser) is responsible to translate the JavaScript code. JavaScript itself is fairly compact yet very flexible. Developers have written a large variety of tools on top of the core JavaScript language, unlocking a vast amount of extra functionality with minimum effort. These include Browser Application Programming Interfaces (APIs) built into web browsers.

3. **Sqlite-** SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application.
4. **Bundler-** Bundler provides a consistent environment for Ruby projects by tracking and installing the exact gems and versions that are needed. Bundler is an exit from dependency hell, and ensures that the gems you need are present in development, staging, and production. Starting work on a project is as simple as bundle install. It maintains a consistent environment for ruby applications. It tracks an application's code and the rubygems it needs to run, so that an application will always have the exact gems (and versions) that it needs to run.
5. **Sass(Syntactically Awesome Stylesheets)-** Sass is an extension of CSS that adds power and elegance to the basic language. It is the most mature, stable, and powerful professional grade CSS extension language in the world. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly, particularly with the help of the Compass style library.

Chapter 2 Feasibility Study

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the re-sources required to carry through. The feasibility study determines whether the proposed solution is feasible based on the priorities of the requirements of the organization. Feasibility study helps to evaluate cost effectiveness of proposed system. The feasibility study is carried out to test if the proposed system is worth being implemented. Not only cost but it also evaluates the time required , technical considerations and human resources required for the project.

Scope of Feasibility Analysis: In general terms, the elements of a feasibility analysis for a project should convert the following:

Need Analysis indicates recognition of a need for the project. The need may affect the organization itself, another organization, the public, or the government. A preliminary study is then conducted to confirm and evaluate the need. A proposal of how the need may be satisfied is then made.

Questions that should be asked include:

- Is the need significant enough to justify the proposed project?
- Will the need still exist by time the project is completed?
- What are the alternate means of satisfying the need?
- What are the economic, social, environmental, and political impacts of the need?

2.1 Technical Feasibility

The objective of the technical feasibility step is to confirm that the product will perform and to verify that there are no production barriers. Considering our proposed solution there is no such problem. The chosen technologies seem outstanding and perfect match for the requirements. Chosen technology can handle multiple users at the same time with fast response time. Also they helps to implement development for proposed system with ease. The proposed system can be developed using simple languages like Ruby ON rails JavaScript, Sass, ReactJS and HTML, CSS. The meteor framework and package

2.2 Economical Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. The concept of Economical Feasibility deals with the fact that a system that can be developed and will be used on installation must be

profitable for the Organization. Since the proposed system is a computer generated web-application, the cost requirements is basically negligible. Hence, this system is economically feasible.

2.3 Operational Feasibility

Operation feasibility is a measure of how people feel about the system. Operational Feasibility is dependent upon determining human resources for the project. The project is feasible here as well. The project can easily be operated on any operating system. It is mainly related to human organizational or social aspects. The points to be considered are - The system interface is standard, user friendly and provides extensive help. Hence no special training are required.

2.4 Schedule Feasibility

This involves questions such as how much time is available to build the new system, when it can be built. Our proposed solution can be built easily within the scheduled time. To determine the schedule feasibility. A project will fail if it takes too long to be completed before it is useful. It is necessary to determine whether the deadlines are mandatory or desirable.

Chapter 3 Requirements

3.1 Software Requirements

1. **Operating System:** Windows, Linux OS
2. **Backend Software:** Puma Server
3. **Frontend Software:** Bundler package manager, JavaScript, Ruby on Rails, SQLite
4. **Browser:** Any modern web browser e.g. Google Chrome, Mozilla Firefox, Opera
5. **Text Editor:** Any text editor for editing html, css, javascript and ruby on rails files. Atom text editor is recommended.

3.2 Hardware Requirements

1. **Processor:** Intel i3 processor or above for local machine
2. **RAM:** 2 GB RAM or above for local machine
3. **HDD:** Minimum 5 GB of free Hard drive space
4. **Supported Architecture:**
 - a. 32-bit(X86)
 - b. 64-bit(X64)

Chapter 4 Bibliography

1. <http://rubyonrails.org/>
2. <https://www.w3schools.com/js/>
3. <http://bundler.io/>
4. <https://www.sqlite.org/>
5. <http://sass-lang.com/guide>
6. <https://ace.c9.io/#nav=about>
7. <http://puma.io/>
8. <https://devcenter.heroku.com/articles/deploying-rails-applications-with-the-puma-web-server>
9. <https://github.com/>
10. <https://try.github.io/levels/1/challenges/1>