A WEB PLATFORM FOR ANALYZING AND SOLVING THE PROBLEMS IN THE SOCIETY

## A PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| **SHERIN .S** | **(963520104046)** |
| **POOJA .J.P** | **(963520104036)** |
| **VIJESH .G** | **(963520104312)** |
| **ASLIN .R** | **(963520104015)** |

***in partial fulfilment for the award of the degree***

**of**

# BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

**STELLA MARY’S COLLEGE OF ENGINEERING, ARUTHENGANVILAI**



# ANNA UNIVERSITY :: CHENNAI 600 025

## MAY 2023

ANNA UNIVERSITY : CHENNNAI 600 025

# BONAFIDE CERTIFICATE

Certified that this project report **“ A WEB PLATFORM FOR ANALYZING AND SOLVING PROBLEMS IN THE SOCIETY**” is the bonafide work of “**SHERIN. S (963520104046) , POOJA .J.P (963520104036) , VIJESH .G (963520104312) , ASLIN . R(963520104015)”** who carried out the project work under my supervision.

## SIGNATURE SIGNATURE

Dr. F.R. Shiny Malar M.Tech., Ph.D., Mr.C. Bastin Rogers M.E.,

## HEAD OF THE DEPARTMENT SUPERVISOR

Professor & Head Associate professor

Department of CSE Department of CSE

Stella Mary’s College of Engineering

Stella Mary’s College of Engineering

Aruthengenvilai-629 202 Aruthengenvilai-629 202

Submitted for the project viva-voce held on…………………………

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

First of all we thank the almighty, the supreme guide whose continuous grace and mercy see us through each day and given as courage in the undertaking of the project.

We sincere thanks goes to the chairman **Dr. Nazareth Charles, Ex. Indian Navy**, for his encouragement and for rendering a platform for completing the project in the successful manner.

We are indebted to **Dr. J . Jenix Rino, M.E.,M.B.A.,Ph.D**. Director of Stella Mary’s College of Engineering for providing all the necessary facilities for completing our project.

We are extremely grateful to **Dr. S. Suresh Premil Kumar, M.E., Ph.D.** Principal of Stella Mary’s College of Engineering for giving a source of inspiration throughout our study in the college.

It’s our solace to thank **Mrs. F.R. Shiny Malar, M.Tech., Ph.D.**, Head of the Department of Computer Science And Engineering for her continuous encourage and sustained interest in completion of our project.

We extended our thanks to **Mr. C . Bastin Rogers, M.E,** our Internal guide for her inspiring guidance and valuable advices in doing this project.

We also thank our institution and our faculty members without whom this project would have been a distant reality.

Next, we thank our parent, family members and friends for their moral support and continuous inspiration. Last but not least we thank all our well wishers who helped us during our project duration.

# ABSTRACT

This project involves the development of a web application that enables users in a nation to provide feedback and solutions to societal problems. The web application allows administrators to post new schemes and problems, which users can then provide feedback on and offer their own solutions. The feedback data is then analyzed using data analysis techniques, and machine learning algorithms are used to generate solutions that are in line with the expectations of the users.

The project focuses on improving the user experience by creating an easy-to-use web application that includes features such as login and registration, problem and solution posting, and viewing existing problems and solutions. Additionally, features such as notifications and a rating system are included to keep users informed about the progress of problems and solutions.

The project also addresses ethical concerns regarding the use of machine learning algorithms to generate solutions by ensuring transparency in the process. Overall, the project aims to facilitate communication between users and the government, and provide a platform for generating solutions to societal problems.

**TABLE OF CONTENTS**

**CHAPTER NO TITLE PAGE. NO**

|  |  |
| --- | --- |
| **ABSTRACT** | **iv** |
| **TABLE OF CONTENT** | **v** |
| **LIST OF FIGURES** | **vii** |
| **LIST OF ABBREVIATIONS** | **viii** |
| **1. INTRODUCTION** | **1** |
| 1.1 Overview | 1 |
| 1.2 Aim And Objectives | 1 |
| 1.3 Problem Definition | 2 |
| 1.4 Project Report Layout | 2 |
| **2. LITERATURE SURVEY** | **3** |
| **3. SYSTEM ANALYSIS** | **6** |
| 3.1 Existing System | 6 |
| 3.2 Proposed System | 6 |
| **4. REQUIREMENT ANALYSIS** | **7** |
| 4.1 System Requirements | 7 |
| 4.2 Feasibility Study | 9 |
| 4.3 Technical feasibility | 10 |
| 4.4 Non Functional Requirement | 11 |
| 4.5 Summary | 12 |
| **5. SOFTWARE DESIGN** | **14** |
| 5.1 Physical Design | 14 |
| **6. SYSTEM IMPLEMENTATION** | **18** |
| 6.1 System Implementation Maintenance and Review | 18 |
| 6.2Hardware Evaluation Factors | 18 |
| 6.3Software Evaluation Factors | 19 |

1. TEST PROCEDURE AND TEST CASES 20
   1. [Test Procedure 20](#_TOC_250001)
   2. [Test Cases 21](#_TOC_250000)
2. CONCLUSION AND FUTURE SCOPE 22
3. SCREENSHOTS 24
4. REFERENCES 31

# LIST OF FIGURES

## FIGURE NO FIGURE NAME PAGE NO

* 1. User Case Diagram 15
  2. Activity Diagram 16
  3. Communication Diagram 17

9 Screenshots 24

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| PYC | Python Compiled File |
| API | Application Programming Interfaces |
| UML | Unified Modelling Language |
| UI | User Interface |
| VSCODE | Visual Studio Code |
|  |  |
|  |  |

**Chapter – 1**

**Introduction**

**1.1 OVERVIEW**

The purpose of this project is to develop a web application that allows administrators to post various schemes and problems, and citizens to provide feedback and solutions to these problems. The application is designed to facilitate communication between citizens and administrators, and enable the generation of solutions based on user feedback.

The web application has been developed using various technologies, including HTML, CSS, Bootstrap 5, Django, and dbSQLite3. These technologies have been used to create a user-friendly interface and ensure seamless functionality across different devices and platforms.

The application features a home page that provides users with access to different pages, including a problem statement page that lists various problems and allows users to post their own problems and solutions. The application also allows users to view other people's problems and solutions, and administrators can easily access and analyze user feedback using the data analysis and machine learning techniques implemented in the application.

The project report describes the design and implementation of the web application, including the various features and functionality of the application. The report also discusses the various techniques used for data analysis and machine learning, and how they have been implemented in the application.

Overall, this web application has the potential to facilitate communication between citizens and administrators, and help to solve real-world problems in society. The application's user-friendly interface, coupled with its data analysis and machine learning capabilities, makes it an effective tool for generating solutions based on user feedback. The report provides an in-depth analysis of the web application's design and functionality, and its potential impact on society.

**1.2 Objective:**

The objective of this project is to develop a web application that enables citizens to provide feedback and solutions to problems faced by society, facilitating communication between citizens and administrators. The application's data analysis and machine learning capabilities will be leveraged to generate solutions based on user feedback, providing administrators with valuable insights and enhancing decision-making processes.

To achieve this objective, the project will focus on the following:

1. Designing and implementing a user-friendly web application: The web application will be designed to be easy to use and accessible to citizens with different levels of technical expertise. The application's user interface will be optimized for different devices and platforms, ensuring that it is accessible to as many people as possible.
2. Developing a problem statement page: The application will feature a problem statement page that lists various problems faced by society. Citizens will be able to post their own problems and solutions, enabling administrators to gather valuable feedback and insights from a diverse range of users.
3. Implementing data analysis techniques: The web application will leverage data analysis techniques to provide administrators with insights into user feedback. The data analysis techniques will be used to filter out unwanted or repeated feedback, enabling administrators to focus on the most critical issues.
4. Incorporating machine learning capabilities: Machine learning techniques will be used to generate solutions based on user feedback. The machine learning algorithms will analyze the data collected from user feedback and provide recommendations for potential solutions. This will help administrators to make informed decisions and develop effective strategies to address society's problems.
5. Testing and evaluating the application: The application will be thoroughly tested and evaluated to ensure that it meets its objectives and functions as intended. Feedback from users will be solicited and analyzed to identify areas for improvement and enhance the application's functionality.

The project's success will be evaluated based on the following criteria:

1. The application's ability to facilitate communication between citizens and administrators.
2. The effectiveness of the data analysis and machine learning techniques in generating solutions based on user feedback.
3. The application's user-friendliness and accessibility.
4. The quality of the application's design and functionality.
5. The application's potential impact on society and its ability to provide effective solutions to real-world problems.

The development of a web application that facilitates communication between citizens and administrators and leverages data analysis and machine learning techniques to generate solutions based on user feedback is a critical step in addressing the problems faced by society. The objective of this project is to develop a web application that meets these criteria and provides a valuable tool for administrators to make informed decisions and develop effective strategies to address society's problems.

The project will focus on designing and implementing a user-friendly web application, developing a problem statement page, incorporating data analysis and machine learning capabilities, and testing and evaluating the application. The success of the project will be evaluated based on the application's ability to facilitate communication, the effectiveness of the data analysis and machine learning techniques, the application's user-friendliness, the quality of the design and functionality, and the potential impact of the application on society.

The successful completion of this project will provide administrators with a powerful tool to address society's problems and enable citizens to provide valuable feedback and solutions. The project's outcomes will contribute to the development of effective strategies to address real-world problems and enhance the quality of life for people in society.

**Problem Definition:**

The problem we aim to address is the lack of efficient communication between citizens and administrators regarding the various schemes and problems in society. Often, citizens are unaware of the schemes implemented by the government, and are unable to provide feedback or suggest solutions to existing problems. This results in a gap between citizens and administrators, leading to ineffective governance.

**CHAPTER-2**

**LITERATURE SURVEY**

Citizen participation is a broad term that encompasses a variety of ways in which citizens can get involved in the governance of their communities. It can include activities such as voting, volunteering, contacting elected officials, and attending public meetings. Citizen participation is important because it helps to ensure that governments are accountable to the people they serve. It also helps to build trust between citizens and their government, and it can lead to better decision-making.

Online platforms that allow citizens to provide feedback and solutions to societal problems have become increasingly popular in recent years. These platforms aim to bridge the gap between citizens and administrators, allowing for better communication and collaboration in problem-solving.

One such platform is FixMyStreet, which was launched in the United Kingdom in 2007. This platform allows citizens to report issues such as potholes, broken streetlights, and graffiti to their local authorities. The reports are then passed on to the relevant authorities, who are responsible for addressing the issues. FixMyStreet has been successful in engaging citizens in the problem-solving process and has helped to improve local communities.

Similarly, in India, the government launched the MyGov platform in 2014, which allows citizens to provide feedback on government policies and initiatives. The platform also allows citizens to participate in contests and surveys, and the feedback received is used to inform policymaking.

Another popular platform is OpenIDEO, which is a global online community that allows users to collaborate on solving various challenges. The platform has been used to address issues such as climate change, healthcare, and education, and has been successful in generating innovative solutions.

There have been several studies on the effectiveness of online platforms in citizen engagement and problem-solving. For example, a study by Kim and Lee (2014) found that online platforms can facilitate citizen participation in local governance and improve the quality of decision-making. Similarly, a study by Lee and Kwak (2017) found that online platforms can help to address social problems by leveraging the collective intelligence of citizens.

Machine learning has also been used in various online platforms to improve the accuracy and efficiency of data analysis. For example, in the context of citizen science, machine learning algorithms have been used to classify images and identify species (Sukumar et al., 2017). In the context of social media, machine learning algorithms have been used to analyze user sentiment and identify trending topics (Huang et al., 2017).

The literature suggests that online platforms can be effective in engaging citizens in problem-solving and improving the quality of decision-making. Machine learning can also be used to improve the accuracy and efficiency of data analysis in these platforms. However, further research is needed to understand the effectiveness of these platforms in different contexts and to identify best practices for their design and implementation.

In the context of this project, the literature survey highlights the potential of the proposed web application in facilitating communication between citizens and administrators and generating solutions based on user feedback. The survey also provides insights into the design and implementation of similar platforms and the use of machine learning in data analysis.

The proposed web application will be designed to be user-friendly and accessible to citizens with different levels of technical expertise. The application will also be designed to be scalable, so that it can accommodate a large number of users.

The web application will use a variety of machine learning techniques to analyze user feedback and generate solutions. These techniques will include natural language processing, sentiment analysis, and machine learning algorithms.

The web application will be evaluated in terms of its ability to facilitate communication between citizens and administrators, generate solutions based on user feedback, and improve the quality of decision-making.

**CHAPTER-3**

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

Currently, there is no dedicated web application available that effectively addresses the communication gap between administrators and the general public regarding schemes, problems, and their potential solutions. The lack of a centralized platform hinders efficient collaboration and feedback collection. Administrators face challenges in disseminating information about schemes, while individuals struggle to provide their valuable input and suggestions. Moreover, there is no automated mechanism for data analysis and machine learning to derive meaningful insights from the feedback received. Consequently, decision-makers and leaders find it difficult to identify popular schemes or effective solutions proposed by the public. The manual system where the admins and users have to communicate through emails, phone calls, or in person. This system is inefficient and time-consuming. It is also difficult to track the progress of problems and solutions. Some of the websites that exist for this purpose are not user-friendly and are not efficient. Most people do not know about these websites.

**Proposed Solution:**

To address the existing limitations and enhance the communication and collaboration between administrators and the public, a proposed solution is to develop a web application that serves as a centralized platform for sharing schemes, addressing societal problems, and gathering feedback and solutions from users. The application will offer the following key features:

1. User-Friendly Interface: The web application will have an intuitive and user-friendly interface, ensuring easy navigation for both administrators and users. Clear and concise instructions will guide users in posting feedback, solutions, and accessing scheme details.

2. Scheme and Problem Management: Administrators will have the ability to post new schemes and problem statements on the platform. They can provide comprehensive descriptions, relevant documents, and important dates associated with each scheme or problem to ensure clarity and transparency.

3. Feedback and Solution Submission: Users will be able to provide feedback on schemes and propose their own solutions to societal problems. They can submit their opinions, suggestions, and ideas in a structured format, enabling easy analysis and evaluation of the feedback received.

4. Data Analysis and Filtering: The application will incorporate data analysis techniques to analyze and filter the user feedback data. Advanced algorithms will be used to identify and remove empty, repeated, or irrelevant feedback, ensuring clean and valuable data for further analysis.

5. Machine Learning-based Solution Generation: By leveraging machine learning algorithms, the application will train on the filtered dataset to understand the preferences and expectations of the users. This will enable the generation of potential solutions based on the collective feedback received, helping administrators and leaders identify viable approaches to address societal problems effectively.

6. Highlighting Prominent Solutions: The application will prominently display the most popular or highly-rated solutions to the identified problems. This will aid higher officials and leaders in quickly accessing the solutions that align with the expectations and needs of the public, facilitating efficient decision-making and problem-solving processes.

7. Secure User Authentication: The web application will employ a robust user authentication system to ensure secure access and prevent unauthorized usage. Users will be required to register and login to participate in providing feedback and solutions.

The proposed solution aims to bridge the gap between administrators and the public, empowering citizens to actively contribute their perspectives and solutions to societal problems. By leveraging data analysis and machine learning, the application will facilitate the identification of popular schemes, effective solutions, and insights that can shape policy-making and decision-making processes.

**CHAPTER-4**

**REQUIREMENT ANALYSIS**

The web application should provide user registration and authentication features, allowing users to create accounts and securely log in. Administrators should have the capability to manage schemes by adding, editing, and deleting them. Similarly, problems should be managed, enabling administrators to post and update problem statements. Users should be able to view the schemes and problems posted on the platform, allowing them to gain insights into the issues at hand. The application should facilitate the submission of feedback and solutions from users, ensuring that they can contribute their ideas and suggestions to address the identified problems. Additionally, a data analysis component should be implemented to filter and analyze the feedback data, removing any redundant or irrelevant information. This will provide administrators with clean and meaningful data for decision-making purposes. Machine learning algorithms should be integrated to generate potential solutions based on the collected feedback, enhancing the application's ability to offer valuable insights and recommendations. Finally, the system should highlight prominent solutions to help higher officials and leaders easily access and consider the most favored or highly-rated proposals.

**FIGMA**

Figma is a powerful cloud-based design and prototyping tool that enables collaborative interface design and development. It provides a comprehensive platform for designing, prototyping, and sharing user interfaces and visual assets. With its intuitive and user-friendly interface, Figma allows designers and teams to work seamlessly together, regardless of their geographical locations.

One of the key advantages of Figma is its real-time collaboration feature. Multiple designers can work simultaneously on a project, making edits, adding components, and providing feedback in real-time. This fosters efficient teamwork and eliminates the need for version control or file syncing, as everything is stored and updated in the cloud.

Figma offers a wide range of design features and tools to create visually stunning interfaces. Designers can easily create and customize vector-based shapes, icons, and illustrations, or import existing design assets. It supports various design elements like frames, layers, and styles, allowing for consistent and scalable designs across the entire project.

Another notable feature of Figma is its prototyping capabilities. Designers can create interactive prototypes by defining links, transitions, and animations between different screens and components. This allows stakeholders and team members to experience the user flow and interactions, providing a more realistic representation of the final product.

Collaboration extends beyond the design phase in Figma. Design files can be easily shared with stakeholders, clients, or developers, who can leave comments, annotations, and even make design suggestions directly on the interface. This streamlines the feedback and review process, ensuring effective communication and alignment among project stakeholders.

Figma also integrates with other design and development tools, allowing for seamless workflows and integrations. It supports plugins and integrations with popular design systems, version control systems, and project management tools, further enhancing its versatility and adaptability to different project requirements.

In summary, Figma is a feature-rich and collaborative design tool that empowers designers and teams to create, iterate, and collaborate on user interfaces and visual assets. Its real-time collaboration, powerful design features, prototyping capabilities, and seamless integrations make it a preferred choice for designers working on web and mobile applications. \]

**HTML**

HTML (Hypertext Markup Language) is the standard markup language used for creating web pages and applications. It provides a structured framework for organizing and presenting content on the internet. With its simplicity and wide adoption, HTML plays a crucial role in web development.

One of the key strengths of HTML is its versatility and compatibility. It is supported by all modern web browsers, making it accessible across different platforms and devices. HTML allows developers to structure content using elements such as headings, paragraphs, lists, tables, and forms. These elements provide a semantic structure that improves accessibility, search engine optimization, and overall user experience.

HTML also supports the integration of multimedia elements, including images, videos, and audio. By using appropriate tags and attributes, developers can embed media files directly into web pages, enhancing their visual appeal and interactivity. Furthermore, HTML provides the foundation for responsive design techniques, allowing websites to adapt and display optimally on various screen sizes and resolutions.

Another key feature of HTML is its ability to create hyperlinks, enabling seamless navigation between different web pages and resources. Hyperlinks can be used to connect internal pages within a website or link to external websites, facilitating information retrieval and enhancing user engagement.

HTML is a markup language that can be combined with CSS (Cascading Style Sheets) to control the visual presentation of web pages. CSS provides extensive styling options, allowing developers to customize the appearance of HTML elements, such as colors, fonts, layouts, and animations. This separation of content (HTML) and presentation (CSS) ensures maintainability and scalability in web development projects.

Furthermore, HTML5, the latest version of HTML, introduced new features and APIs that enhance the capabilities of web applications. It introduced semantic elements like `<header>`, `<nav>`, `<section>`, and `<footer>`, which aid in structuring web page content more meaningfully. HTML5 also brought multimedia enhancements, including the `<video>` and `<audio>` elements, as well as the `<canvas>` element for dynamic graphics and animations.

In summary, HTML is a fundamental building block of the web. Its simplicity, compatibility, and versatility make it an essential language for structuring and presenting content on the internet. With its continuous evolution and support for modern features, HTML remains a cornerstone of web development, enabling developers to create engaging and interactive experiences for users.

**CSS**

CSS (Cascading Style Sheets) is a style sheet language used for describing the visual appearance and layout of HTML documents. It complements HTML by providing the means to define colors, fonts, sizes, spacing, and other stylistic elements of web pages. With CSS, developers have precise control over the presentation and aesthetics of their websites.

One of the primary advantages of CSS is its ability to separate the style from the structure and content of a web page. This separation allows developers to create reusable style rules and apply them consistently across multiple web pages. By centralizing the styling instructions in a separate CSS file, the overall maintenance and scalability of a website are significantly improved.

CSS provides a wide range of selectors and properties that enable developers to target specific HTML elements and apply desired styles. Selectors can be based on element types, class names, IDs, or hierarchical relationships, allowing for granular control over the styling. Properties such as color, font-family, margin, padding, and background enable customization of various visual aspects of web pages.

With CSS, developers can create responsive designs that adapt to different screen sizes and devices. Media queries in CSS allow for the application of specific styles based on the characteristics of the user's device, such as screen width, resolution, or orientation. This enables the creation of mobile-friendly and flexible layouts, enhancing the user experience across various devices.

CSS also supports advanced styling techniques like animations and transitions. Developers can apply animations to elements, defining keyframes, duration, and timing functions. This adds dynamic and interactive elements to web pages, creating engaging user experiences. Transitions, on the other hand, allow smooth and gradual changes in CSS property values, providing visual feedback during user interactions.

Additionally, CSS can be used to control the layout and positioning of elements on a web page. Flexbox and CSS Grid are two powerful layout models in CSS that enable developers to create complex and responsive page structures. They offer flexibility in arranging and aligning elements, adjusting their size and position based on available space.

CSS preprocessors, such as Sass and Less, extend the capabilities of CSS by introducing variables, nesting, mixins, and other features. These preprocessors enhance code organization and maintainability, allowing for more efficient and modular CSS development.

In summary, CSS is a crucial component in web development, enabling developers to customize the visual appearance, layout, and responsiveness of web pages. Its separation from HTML promotes code reusability and maintainability, while its extensive range of selectors, properties, and techniques offer great flexibility in styling. With CSS, developers can create visually appealing, interactive, and responsive websites that provide engaging user experiences.

**BOOTSRAP**

Bootstrap is a popular front-end framework that provides a comprehensive set of tools, components, and styles for building responsive and visually appealing websites and web applications. It simplifies the process of web development by offering pre-designed elements and a responsive grid system that adapts to different screen sizes.

One of the key advantages of Bootstrap is its extensive collection of CSS classes and components. These include buttons, forms, navigation bars, cards, modals, carousels, and much more. By utilizing these ready-to-use components, developers can save time and effort in designing and coding common UI elements, ensuring consistency and a professional look and feel across their projects.

Bootstrap incorporates a responsive grid system that allows developers to create fluid and flexible layouts. The grid system is based on a 12-column structure, which can be easily customized and adjusted to accommodate different screen sizes. This enables the creation of responsive designs that automatically adapt to various devices, such as desktops, tablets, and mobile phones.

In addition to its CSS components, Bootstrap also includes a JavaScript component library. This library offers interactive features, such as dropdowns, tooltips, modals, and scrollspy, that enhance the functionality and user experience of web applications. The JavaScript plugins can be easily integrated into projects, providing dynamic and interactive elements without the need for writing complex JavaScript code from scratch.

Bootstrap is highly customizable, allowing developers to modify the default styles and components to match their project requirements. It provides a wide range of customizable variables and mixins, making it easy to tailor the design and appearance to specific branding or design guidelines. This flexibility ensures that Bootstrap can be adapted to suit a variety of design aesthetics and project needs.

Another advantage of Bootstrap is its active and supportive community. It has a large user base and an extensive collection of online resources, including documentation, tutorials, and forums. This makes it easier for developers to find assistance, troubleshoot issues, and stay up to date with the latest Bootstrap developments and best practices.

In summary, Bootstrap is a powerful and widely-used front-end framework that simplifies the development of responsive and visually appealing websites and web applications. With its vast array of pre-designed components, responsive grid system, JavaScript plugins, and customization options, Bootstrap provides developers with a solid foundation to create modern and engaging user interfaces. Its active community ensures ongoing support and resources for developers, making it an excellent choice for rapid and efficient web development.

**JAVASCRIPT**

JavaScript (JS) is a versatile programming language primarily used for adding interactivity and dynamic behavior to websites. It is a client-side scripting language that runs in the user's web browser, allowing for enhanced user experiences and interactive web applications.

One of the key strengths of JavaScript is its ability to manipulate and modify the Document Object Model (DOM) of a web page. With JS, developers can dynamically update and change the content, structure, and styling of web elements, providing real-time feedback and interactivity to users. This capability enables the creation of responsive and interactive web interfaces.

JavaScript offers a wide range of built-in functions and methods that empower developers to perform complex operations and computations. It supports variables, data types, operators, loops, conditionals, and functions, providing the essential building blocks for writing logic and algorithms. Additionally, JavaScript provides support for object-oriented programming concepts, allowing for code organization and reuse.

The language is extensively used for handling user events, such as button clicks, form submissions, mouse movements, and keyboard interactions. Developers can attach event listeners to specific elements and define custom actions or responses when those events occur. This facilitates user interactions and enables the creation of interactive features like form validation, animations, and interactive components.

JavaScript also provides support for asynchronous programming through the use of callbacks, promises, and the more recent async/await syntax. This allows developers to handle time-consuming tasks, such as fetching data from servers or performing network requests, without blocking the user interface. Asynchronous programming enhances the performance and responsiveness of web applications.

Moreover, JavaScript has a vast ecosystem of libraries and frameworks that extend its capabilities. Libraries like jQuery simplify DOM manipulation and provide additional utilities, while frameworks like React, Angular, and Vue.js offer powerful tools for building complex and scalable web applications. These tools enhance productivity, code organization, and maintainability, making JavaScript a popular choice for modern web development.

Furthermore, JavaScript is not limited to the client-side environment. With Node.js, developers can use JavaScript on the server-side as well. This enables the development of full-stack web applications where JavaScript is used for both front-end and back-end development, allowing for code reuse and seamless integration between client and server components.

In summary, JavaScript is a versatile and essential programming language for web development. Its ability to manipulate the DOM, handle events, perform computations, and support asynchronous programming makes it a powerful tool for creating interactive and dynamic web experiences. With its extensive ecosystem and widespread adoption, JavaScript continues to be a fundamental language for building modern web applications.

**DJANGO**

Django is a high-level Python web framework that facilitates rapid and efficient development of secure, scalable, and maintainable web applications. It follows the model-view-controller (MVC) architectural pattern, emphasizing clean design, code reusability, and the principle of "Don't Repeat Yourself" (DRY).

One of the key advantages of Django is its comprehensive set of built-in features and tools. It includes an object-relational mapper (ORM) that simplifies database interactions, enabling developers to work with databases using Python objects instead of writing SQL queries. This promotes efficient data modeling and abstraction, reducing the complexity of database operations.

Django provides a robust and flexible authentication and authorization system, allowing developers to handle user registration, login, password resets, and user permissions with ease. This built-in authentication system ensures secure user management and protects against common security vulnerabilities.

Another strength of Django is its automatic admin interface. With just a few lines of code, developers can create a fully functional and customizable administrative interface for managing data models and performing CRUD (Create, Read, Update, Delete) operations. This saves significant development time and effort, especially for back-end administration tasks.

Django follows the "batteries included" philosophy, which means that it comes with a wide range of reusable components and libraries. These include form handling, file uploads, caching mechanisms, URL routing, template engine, and internationalization support. These built-in functionalities streamline the development process and allow developers to focus on implementing business logic rather than reinventing the wheel.

Django's templating system provides a powerful way to separate HTML markup from Python code, enabling clean and maintainable code organization. It supports template inheritance, template tags, and filters, allowing for flexible and dynamic content rendering. Additionally, Django supports front-end frameworks like Bootstrap for enhanced user interface design.

Django's scalability and performance are also notable. It incorporates caching mechanisms, database query optimization techniques, and asynchronous task processing through libraries like Celery and Django Channels. These features ensure efficient handling of high traffic and complex web applications.

Django's community is vibrant and active, offering extensive documentation, tutorials, and third-party packages. This makes it easier for developers to get started, find solutions to common challenges, and stay updated with the latest best practices in Django development.

In summary, Django is a powerful Python web framework that empowers developers to build robust, secure, and scalable web applications. Its built-in features, authentication system, automatic admin interface, and extensive ecosystem make it a preferred choice for developers seeking productivity, maintainability, and code efficiency. With Django, developers can focus on delivering high-quality applications while leveraging the framework's best practices and community support.

**DBSQLITE3**

SQLite3 is a lightweight, open-source, embedded relational database management system (RDBMS) that is widely used for small to medium-sized applications. It provides a self-contained, serverless architecture, making it easy to integrate and deploy within various software systems.

One of the key advantages of SQLite3 is its simplicity and ease of use. It is a single-file database that does not require a separate server process, which simplifies setup and administration. Developers can easily incorporate SQLite3 into their projects by linking the appropriate libraries and utilizing the provided APIs.

SQLite3 supports a subset of the SQL language, allowing developers to interact with the database using standard SQL statements. It offers essential SQL features, such as creating and modifying database tables, inserting and retrieving data, and executing queries. The SQL syntax is intuitive and familiar to developers experienced with relational databases, making it straightforward to work with SQLite3.

Despite its lightweight nature, SQLite3 is robust and reliable. It provides transactional capabilities, ensuring data integrity and consistency. Developers can use transactions to group multiple database operations into atomic units, allowing for rollbacks in case of errors or failures. This feature is especially valuable in multi-user scenarios where data concurrency and consistency are critical.

SQLite3 is designed to be highly efficient in terms of performance and resource usage. It is optimized for read-heavy workloads, making it ideal for applications that require fast data retrieval. Additionally, SQLite3 has a small memory footprint and requires minimal disk space, making it suitable for resource-constrained environments.

Another notable feature of SQLite3 is its portability. It is cross-platform and can run on various operating systems, including Windows, macOS, Linux, and mobile platforms like Android and iOS. This flexibility allows developers to build applications that can be deployed on different platforms without significant modifications.

SQLite3 offers extensive documentation and a supportive community, providing resources and assistance to developers. The official SQLite website includes detailed documentation, tutorials, and examples, making it easy to get started and troubleshoot issues. Additionally, there are numerous third-party libraries and frameworks that provide additional functionality and abstraction layers for working with SQLite3.

In summary, SQLite3 is a lightweight and versatile relational database management system that offers simplicity, reliability, and efficiency. Its self-contained architecture, SQL support, transactional capabilities, and cross-platform compatibility make it a popular choice for embedded databases in a wide range of applications. With its small footprint and ease of integration, SQLite3 enables developers to incorporate database functionality into their projects with minimal overhead.

**GIT**

Git is a widely used distributed version control system that allows developers to efficiently manage and track changes in source code during the software development process. It provides a reliable and scalable solution for collaboration, code sharing, and maintaining a history of project modifications.

One of the key advantages of Git is its distributed nature. Each developer working on a project has a complete local copy of the repository, including the entire history of changes. This enables developers to work offline and independently, committing changes to their local repositories. They can later synchronize their changes with a central repository or share them with other team members, facilitating seamless collaboration.

Git offers a streamlined workflow for managing changes in code. Developers can create branches to isolate their work on specific features or bug fixes. This allows multiple team members to work on different aspects of a project simultaneously, without interfering with each other's changes. Once the changes are complete, branches can be merged back into the main codebase, ensuring a controlled and organized integration of new code.

The ability to track changes in Git is a powerful feature. Every commit represents a specific modification to the codebase, including details such as the author, timestamp, and a unique identifier. This enables developers to review the history of changes, understand the evolution of the project, and easily revert to previous versions if needed. Git also provides tools for comparing different versions of files, identifying modifications, and resolving conflicts during merging.

Git simplifies collaboration by offering seamless remote repository integration. Developers can push their local changes to a shared repository, such as GitHub or Bitbucket, making it accessible to other team members. This central repository acts as a single source of truth, where everyone can contribute, review, and discuss code changes. It also provides a platform for code reviews, issue tracking, and project management, enhancing team collaboration and productivity.

Another notable feature of Git is its support for branching and merging strategies. Developers can adopt various branching models, such as GitFlow, to manage the release cycles, hotfixes, and feature development effectively. Git's merging capabilities allow for smooth integration of changes across branches, ensuring a stable and up-to-date codebase.

Git is highly flexible and supports integration with various development tools and services. Continuous Integration (CI) and Continuous Deployment (CD) systems can be seamlessly integrated with Git repositories, automating build processes, and ensuring code quality. Additionally, Git provides APIs and hooks that allow developers to customize and extend its functionality according to project-specific requirements.

In summary, Git is a powerful distributed version control system that offers efficient and flexible management of source code during software development. Its distributed nature, streamlined workflow, change tracking capabilities, collaboration features, and integration options make it an essential tool for teams and individual developers. With Git, developers can work together seamlessly, track changes effectively, and maintain a reliable and organized codebase throughout the development lifecycle.

**GITHUB**

GitHub is a widely used web-based platform that provides hosting for software development projects utilizing Git version control system. It offers a range of features and tools that facilitate collaboration, code sharing, and project management for developers and teams.

One of the key advantages of GitHub is its distributed version control system, Git. Developers can easily track changes to their code, create branches for different features or bug fixes, and merge changes seamlessly. This promotes a collaborative workflow and ensures that the codebase remains organized, versioned, and easily accessible.

GitHub provides a centralized repository for storing and managing code. Developers can create repositories for their projects, upload code files, and maintain a complete history of changes. The repository acts as a central hub where multiple contributors can collaborate, review code, and provide feedback.

Collaboration is a fundamental aspect of GitHub. Developers can clone repositories to their local machines, make changes, and push them back to the remote repository. GitHub allows for easy code review through pull requests, where developers can propose changes, discuss them, and ensure quality control before merging them into the main codebase.

GitHub offers powerful issue tracking and project management tools. Developers can create issues to track bugs, feature requests, or other tasks. These issues can be assigned to team members, labeled, and organized into milestones, providing a comprehensive view of the project's progress and priorities.

Additionally, GitHub provides a platform for open-source development and community collaboration. Developers can contribute to existing projects by forking repositories, making changes, and submitting pull requests. This fosters an environment of knowledge sharing, collaboration, and improvement within the developer community.

GitHub Pages is another noteworthy feature that allows developers to host static websites directly from their GitHub repositories. This makes it easy to showcase projects, documentations, or personal portfolios without the need for a separate hosting service.

Furthermore, GitHub integrates with a wide range of development tools and services. It supports continuous integration and deployment (CI/CD) pipelines through integrations with popular CI/CD tools. This enables automated testing, building, and deploying processes directly from the repository.

GitHub's extensive ecosystem includes a marketplace with various third-party integrations, extensions, and workflows. Developers can enhance their workflows by integrating tools for code analysis, project management, code review, and more.

In summary, GitHub is a powerful platform that provides version control, collaboration, and project management capabilities for software development. Its integration with Git, centralized repository, code review features, issue tracking, and project management tools make it a go-to choice for individual developers, teams, and open-source communities. With its vast ecosystem and continuous improvements, GitHub empowers developers to work efficiently, collaborate seamlessly, and contribute to the success of their projects.

**VSCODE**

Visual Studio Code (VSCode) is a versatile and widely-used source code editor developed by Microsoft. It provides developers with a robust and feature-rich environment for writing, editing, and debugging code across various programming languages. With its extensive set of extensions and customization options, VSCode has become a popular choice for developers of all levels of expertise.

One of the key strengths of VSCode is its lightweight and fast performance. It offers a streamlined and efficient user interface, ensuring a smooth coding experience even when working with large codebases. Its efficient resource utilization allows for quick startup times and minimal impact on system resources.

VSCode supports a wide range of programming languages and offers built-in syntax highlighting, autocompletion, and code formatting. It provides a consistent and intuitive editing experience, making it easy to write clean and error-free code. The editor's intelligent code suggestions and linting capabilities help catch errors and improve code quality, increasing productivity and reducing debugging time.

A notable feature of VSCode is its powerful extension ecosystem. It offers a vast marketplace of extensions developed by the community, enabling developers to enhance the functionality and tailor the editor to their specific needs. These extensions cover various aspects of development, including language support, debugging tools, Git integration, code snippets, themes, and more. The ability to customize and extend VSCode makes it a versatile tool for different programming workflows and environments.

VSCode also includes integrated version control through Git. This allows developers to manage their code repositories directly within the editor, with features like staging changes, committing, branching, and merging seamlessly integrated. This simplifies the collaboration process and facilitates efficient code management.

Another notable feature of VSCode is its powerful debugging capabilities. It provides an integrated debugging environment with support for breakpoints, step-by-step code execution, variable inspection, and more. This enables developers to troubleshoot and diagnose issues in their code effectively, leading to faster bug resolution and improved code quality.

VSCode offers a range of productivity features, such as split views, multiple cursors, and keyboard shortcuts. These features enhance code navigation and editing efficiency, empowering developers to work more quickly and effectively. Additionally, VSCode has extensive documentation and a supportive community, providing resources and assistance to users.

In summary, Visual Studio Code is a versatile and highly customizable source code editor that provides developers with a powerful and efficient environment for writing, editing, and debugging code. Its lightweight nature, extensive extension marketplace, Git integration, debugging capabilities, and productivity features make it a go-to choice for developers across various programming languages and workflows.

**Feasibility Study**

1. Introduction

The proposed project aims to develop a web application that facilitates scheme feedback and problem-solving for the general public. The system will allow administrators to post new schemes or problems, while users can provide solutions and feedback. The collected data will undergo data analysis to filter and present clean data for better understanding and decision-making. Additionally, machine learning techniques will be employed to generate solutions based on user expectations. This feasibility study will assess the viability and potential success of the project.

2. Technical Feasibility

The technical feasibility of the project focuses on assessing the technical requirements and resources needed for successful development and implementation. The project requires expertise in web development technologies, including HTML, CSS, JavaScript, and backend programming languages such as Python or PHP. The team should have experience in designing user-friendly interfaces, implementing data analysis techniques, and integrating machine learning algorithms. Adequate hardware and software resources, including servers, databases, and development tools, must be available or easily accessible to support the project requirements.

3. Economic Feasibility

The economic feasibility analysis involves evaluating the costs and benefits associated with the project. This includes expenses related to hardware, software licenses, infrastructure, hosting, and ongoing support. A cost-benefit analysis should be conducted to determine if the potential benefits, such as improved problem-solving and efficient scheme feedback, outweigh the investment required. It is important to consider factors such as return on investment, long-term sustainability, and potential revenue streams.

4. Legal and Ethical Feasibility

The legal and ethical feasibility of the project requires a thorough assessment of applicable laws, regulations, and ethical guidelines. The web application must comply with relevant data privacy regulations to ensure the protection of user data. Intellectual property rights should be respected, and appropriate measures should be taken to secure user information. User data should be handled securely and anonymized during data analysis to protect individual privacy. The team must conduct a thorough assessment of legal requirements and establish mechanisms to ensure ethical usage of user data.

5. Operational Feasibility

The operational feasibility analysis assesses whether the proposed web application can be seamlessly integrated into existing systems and workflows. It should be compatible with various devices and browsers to ensure accessibility for a wide range of users. Adequate user training and support should be provided to administrators and users to ensure smooth adoption and utilization of the system. Additionally, considerations should be made for scalability and future enhancements to accommodate increasing user demands and evolving requirements.

6. Schedule Feasibility

The schedule feasibility examines whether the proposed timeline is realistic and achievable. It requires a thorough understanding of the project scope, resource availability, and potential risks. A detailed project plan, including milestones and deliverables, should be developed to ensure timely completion. It is important to factor in potential challenges, dependencies, and contingencies to minimize schedule risks and ensure successful project execution.

7. Conclusion

Based on the assessment of technical, economic, legal, ethical, operational, and schedule feasibility, the proposed web application for scheme feedback and problem-solving appears to be feasible. With the right expertise, resources, and careful planning, the project has the potential to provide a valuable platform for administrators and users to collaborate, analyze data, and generate effective solutions for societal problems. However, it is essential to continuously evaluate and adapt the project plan to address any potential challenges or changes in requirements during development and implementation. By considering the feasibility aspects outlined in this study, the project can be successfully executed to achieve its objectives and deliver a valuable solution for scheme feedback and problem-solving.

**CHAPTER-5**

**SOFTWARE DESIGN**

The purpose of the design phase is to plan a solution for the problem specified by the requirements. System design aims to identify the modules that should be included in the system, their specifications, and how they interact with each other to produce the desired results. The goal of the design process is to create a model that can be used later to build the system.

One important aspect of the design phase is the physical design, which involves creating a graphical representation of the system. This representation shows the system's internal and external entities, as well as the flow of data into and out of these entities. Internal entities refer to components within the system that transform the data.

To represent the physical design of the system, various diagrams such as data flow diagrams and use case diagrams can be used. Data flow diagrams illustrate the flow of data between different components of the system, including inputs, outputs, processes, and data stores. Use case diagrams depict the interactions between users and the system, showcasing different use cases and the relationships between them.

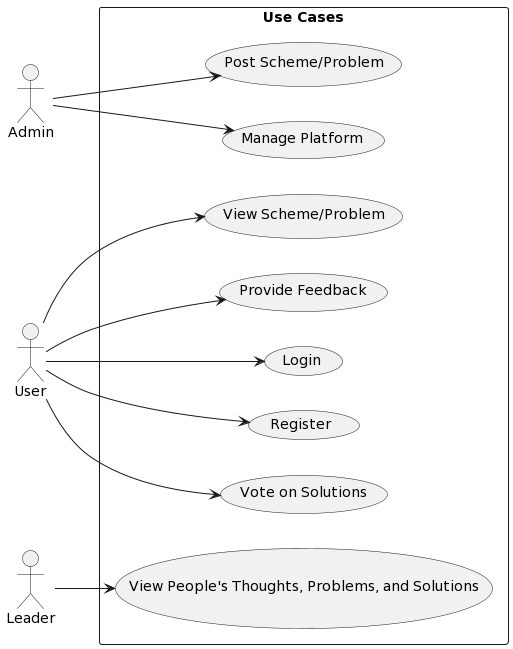
The physical design phase is crucial in visualizing the system's structure and ensuring that it aligns with the specified requirements. It helps in identifying the key components, their functions, and how they collaborate to achieve the desired functionalities. This phase provides a clear blueprint for developers to follow during the implementation stage.

In addition to the graphical representations, the physical design phase also involves making decisions regarding the hardware and software architecture that will support the system. This includes selecting the appropriate technologies, programming languages, frameworks, and databases based on the project's requirements and constraints.

Overall, the physical design phase plays a vital role in translating the abstract requirements into a concrete representation. It provides a visual guide for developers, stakeholders, and users to understand the system's structure and functionality. Through effective physical design, the development team can ensure that the system meets the specified requirements and is capable of delivering the desired outcomes.

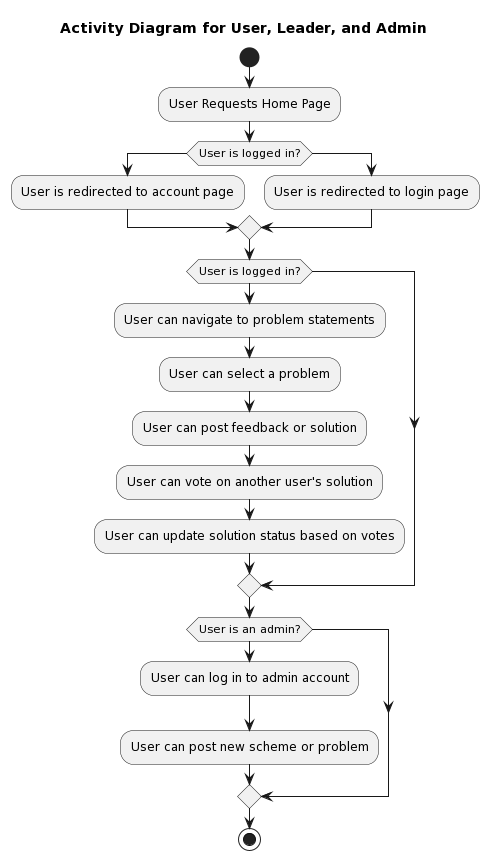
**USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. User in a use case diagram is any entity that performs a role in one given system. A use case represents a function or an action within the system. User may use to login as a member and access all the features provided in the Web application.



**ACTIVITY DIAGRAM**

The activity diagram for the project provides a visual representation of the flow of activities and actions within the web application. It depicts the sequence of steps and decisions involved in various processes, such as user registration, feedback submission, solution entry, and data analysis. The activity diagram helps to illustrate the interactions between different entities and the order in which activities occur. It serves as a valuable tool for understanding the overall functionality of the system and identifying potential areas for improvement or optimization. By analyzing the activity diagram, developers and stakeholders can gain insights into the user journey and ensure that the application operates smoothly and efficiently.



**CLASS DIAGRAM**

The class diagram for the project provides a concise representation of the system's structure and the relationships between its key components. It visually depicts the classes, their attributes, and the associations between them.

In the class diagram, classes are depicted as boxes, with their names written inside. The attributes of each class are listed below the class name, representing the data that the class holds. Relationships between classes are shown through arrows, indicating associations, inheritances, or dependencies.

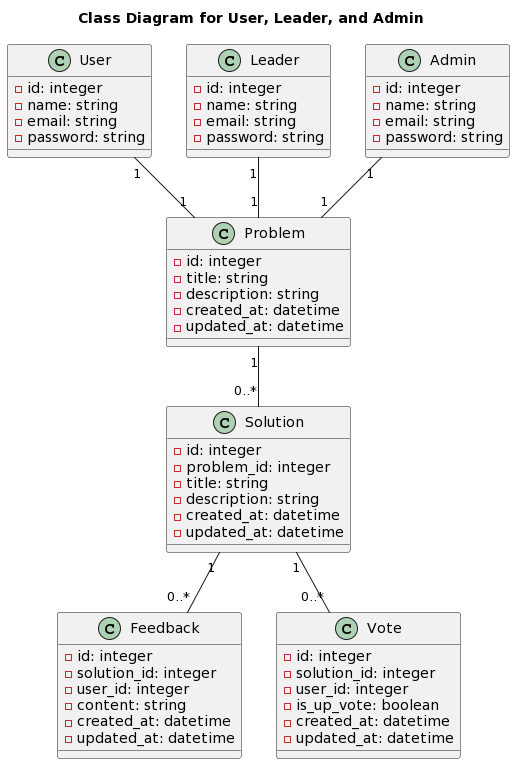
For our project, the class diagram will include important classes such as User, Administrator, Scheme, Problem, Solution, and Feedback. The User class will have attributes such as username, email, and password, representing the user's login information. The Administrator class will extend the User class and include additional attributes related to administrative privileges.

The Scheme class will contain details about each scheme, including a title, description, and deadline. The Problem class will store information about the identified problems, such as a description and the associated scheme. The Solution class will hold the proposed solutions, including details like the description, associated problem, and the user who submitted the solution.

The Feedback class will capture user feedback, including their opinions, suggestions, and ratings for schemes or problems. It will have attributes like feedback description, rating, and the user who provided the feedback.

The class diagram will depict the associations between these classes, indicating how they are connected. For example, there will be an association between the User class and the Feedback class to represent the fact that a user can provide feedback.

The class diagram serves as a visual guide for developers, helping them understand the structure and relationships within the system. It assists in the implementation phase by providing a clear blueprint of the classes and their interactions, aiding in the development of a robust and efficient web application for scheme feedbackand problem solving.



**SEQUENCE DIAGRAM**

A sequence diagram is a type of behavioral diagram that illustrates the interactions between different objects or components in a system over time. In the context of the project for scheme feedback and problem solving, the sequence diagram showcases the flow of messages and actions between the system's entities.

The sequence diagram depicts the chronological order of events and the communication between various components, such as users, administrators, and the application itself. It captures the dynamic behavior of the system, showcasing how different entities interact and collaborate to accomplish specific tasks.

