

Yojana: Advanced Collaborative Software Development Platform with Special Features for Visually Impaired Developers to Combat Disparity Amidst Software Developers

Tanuka Majumder¹, Sakshi Patel², Aastha Singh³ and Dr. Shobha Sharma⁴

¹⁻⁴Department of Electronics and Communication Engineering, Indira Gandhi Delhi Technical University for Women, New Delhi, India, 110006

Email: tanukamajumder6405@gmail.com, shobhaa_sharma16@yahoo.co.in, shobhasharma@igdtuw.ac.in

Abstract—Modern web applications have touched new horizons with the combination of unique ideas and technical knowledge. Yojana is a paragon of this kind of amalgamation. Yojana is a collaboration software equipped with a code editor that is used to develop software applications and also allows us to discuss and strategize all our ideas and keep track of our work. It ensures that all the developers can collaborate and coordinate for smooth functioning of a project. Drishti is the name that we have given to the code editor that provides a perfect platform to code and develop applications of your choice. It also has some distinctive features that also allows visually impaired people to code without any inconvenience. We are living in an era where only technical knowledge is not proved to be fruitful in the success of any project. Proper planning and discussion is essential for smooth execution of a project. Lack of coordination is one of the most important factors that contribute to inefficiency. Yojana is an ideal place to counter all these issues while working on a project. Yojana also gives you a medium to implement your ideas via Drishti. Yojana allows collaborators to give updates and also check the progress of the tasks performed by the other team members. It also provides detailed descriptions of the various actions performed and changes that are required can be suggested by the other team members. Yojana will surely simplify the entire process of project planning and make it more feasible for collaborators to ensure proper execution of the project. Yojana can be an innovator in a way to integrate the feature of a code editor equipped with web speech API which enables everyone including a visually impaired person to code without facing any difficulty. Both project planning and implementation are possible simultaneously in Yojana which sets it apart from the other collaboration tools.

Index Terms— Collaboration software, Coding platform, Web Speech API.

I. INTRODUCTION

In today's workplace environment, collaboration software plays a vital role in the implementation of the ideas and goals of a team working on a project. A good workflow can only be established if proper coordination and communication is there among the team members. The technology world demands more efficiency [1] in minimum time as it is evolving at a fast pace. A systematic and efficient route should be followed in project

planning otherwise the goals set by an organization may not be achieved on time. In a workplace or an organization it is important to see the contribution of all the employees who are working in a particular department and also to keep a track of how much work is done and the data of the tasks that are yet to be completed. Yojana provides you a medium to maintain a record of all the weekly goals that are achieved and the goals are not achieved yet. Updates of day-to-day tasks performed can be given by users on Yojana by following a very simple process which is less time-consuming and more effective. Drishti, a code editor which is added to Yojana is like an addition of wings to your idea facilitating you to code, develop & test your applications. Drishti has various features that make coding easier for visually impaired people. Visually impaired software developers can code and develop their working applications on Drishti without any inconvenience as it makes use of web speech API to make their experience better on the code editor. Team collaboration has several realms to emphasize as different teams work on different domains and all the teams have to be in sync to ensure a smooth workflow. Yojana can be used for all kinds of collaborations. Apart from business collaboration, personal collaboration and planning of informal events can also be done on Yojana. To combat the miscommunication and mismanagement in a team, Yojana can be a savior as you can achieve your weekly goals by planning and organizing all the tasks in a systematic way.

With the arrival of coronavirus, the world is facing a new working reality and various other challenges that were never experienced before. Team collaboration software can be extremely helpful [8] in maintaining the workflow and moving towards accomplishing the company goals. Not only workplaces but students in schools, universities and other institutions are also obliged to adopt online means to complete their assignments and projects. Yojana will provide them a suitable atmosphere to complete their work in the most systematic way possible. College students can collaborate and discuss and plan their projects without facing any difficulty to access the updates later. Yojana is very user-friendly and simple to use. It mainly works on two facets that are crucial for developing any application project planning and project implementation via code editor. You don't have to be a professional to work on Yojana as it caters to a wide audience from children to professionals by providing simple steps to collaborate and discuss with other team members the action plan. There is no scarcity of work collaboration tools but they are either very difficult to use or do not have any particular function needed by a user. Yojana is equipped with Drishti which allows you to implement and code your ideas and develop an application that helps you understand the feasibility of the plan and make changes accordingly.

II. DESIGN AND IMPLEMENTATION

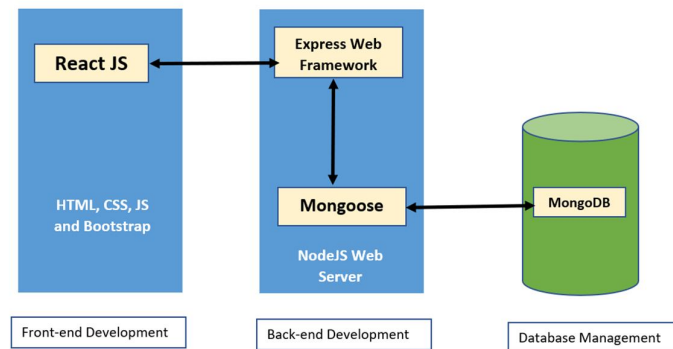


Figure 1: Design and implementation model

Yojana is created by building a Node.js server with Express. MongoDB database is also used to store the data used in the software. Yojana is developed by considering frontend and backend as two entities. MongoDB, Node.js and Express are used to develop the back-end of web applications [6]. This corresponds to HTML documents, scripts, database management, HTTP requests, etc. With the MERN stack, URL's are created & On these URLs, they then create, read and modify the data that is stored and retrieved in the MongoDB database [3]. These URLs represent functions, with HTTP calls as the originators. The data is sent by the requests and the server in return will be responsible for modifying the database. React's role in Yojana is to execute HTTP requests. With React, Ajax calls [7] are made which allows us to set up dynamic data downloads without the need for reloading the page.

Using the Component-Driven Development (CDD) methodology, combined with Behavior-driven development (BDD), helped in splitting the views into smaller components, making sure they are decoupled and reusable. Section reducer is created since we have only one section with many catalogs of handouts, we represent the section with an array of catalogs ids. Similarly, a catalogs reducer is created where we store the catalogs in an object where the keys are the ids of the catalogs, so that we can access the details of individual catalogs. Each catalog will have an `_id`, a title and an array of handout ids. We have actions to add a catalog, change its title, delete it, add a handout, move a handout (for the drag and drop) and delete a handout. Then we have a handouts reducer. As for the catalogs we have an object where the keys are the handout ids and we have action to add a card, change its title and delete it. We also capture the action `Delete_Catalog` to delete every handout of a catalog to be deleted.

```
const section = (state = { catalogs: [] }, action) => {
  switch (action.type) {
    case "Add_Catalog": {
      const { catalogId } = action.payload;
      return { catalogs: [...state.catalogs, catalogId] };
    }
    case "Move_Catalog": {
      const { oldCatalogIndex, newCatalogIndex } = action.payload;
      const newCatalogs = Array.from(state.catalogs);
      const [removedCatalog] = newCatalogs.splice(oldCatalogIndex, 1);
      newCatalogs.splice(newCatalogIndex, 0, removedCatalog);
      return { catalogs: newCatalogs };
    }
    case "Delete_Catalog": {
      const { catalogId } = action.payload;
      const filterDeleted = tmpCatalogId => tmpCatalogId !== catalogId;
      const newCatalogs = state.catalogs.filter(filterDeleted);
      return { catalogs: newCatalogs };
    }
    default:
      return state;
  }
};

const handoutsById = (state = {}, action) => {
  switch (action.type) {
    case "Add_Handout": {
      const { handoutText, handoutId } = action.payload;
      return { ...state, [handoutId]: { text: handoutText, _id: handoutId } };
    }
    case "Change_Handout_Text": {
      const { handoutText, handoutId } = action.payload;
      return { ...state, [handoutId]: { ...state[handoutId], text: handoutText } };
    }
    case "Delete_Handout": {
      const { handoutId } = action.payload;
      const { [handoutId]: deletedHandout, ...restOfHandouts } = state;
      return restOfHandouts;
    }
    case "Delete_Catalog": {
      const { handouts: handoutIds } = action.payload;
      return Object.keys(state)
        .filter(handoutId => !handoutIds.includes(handoutId))
        .reduce(
          (newState, handoutId) => ({ ...newState, [handoutId]: state[handoutId] }),
          {}
        );
    }
    default:
      return state;
  }
};
```

With our reducers in place, a store is created. Then, the store is imported and the component is wrapped with the Redux Provider. Then, we started to visualize the section. We have taken a top-down approach, first we imported the components required and then wrote them. We created a Section component, Catalog Component and Handout Component. Section component connects to Redux to retrieve the stored section and display all its catalogs with a Catalog component. The Catalog component receives the CatalogId from the section and retrieves the Catalog details from the Redux store. Several other features are implemented like editing and deleting handouts, moving catalogs and handouts. Various other components are included like AddCatalog component, CatalogEditor Component, EditButtons component, HandoutEditor component, etc. for the working of other features.

Drishti, our web application development environment allows every software developer to code and develop software applications. It is also useful to visually impaired software developers. They can also code and develop applications without any external help. Drishti makes use of web speech API to make a suitable coding environment for visually impaired developers. The Web Speech API enables all the developers to produce the speech input and text-to-speech output features [12] in the most efficient way. The Web Speech API consists of two functions, speech synthesis, also mentioned as text to speech, and speech recognition, or speech to text.

The Speech Synthesis (text to speech) interface of the Web Speech API is the main controller interface of the speech service and this will be used to retrieve information of the synthesis voices [4] which are available on the user's device like start and pause speech, and other commands besides. A computer system is employed for this purpose only and is named a speech computer or speech synthesizer, and it is mainly implemented in software, hardware products and other wide range of products. A text-to-speech (TTS) system [5] mainly converts simple/normal language text into speech.

III. YOJANA

Yojana is a software which helps students, software developers & experts collaborate, discuss and organize their ideas together in a section, keep track of their work and can also develop software applications all in one place. Yojana helps in accomplishing tasks faster and more efficiently. People can collaborate, collect all the project

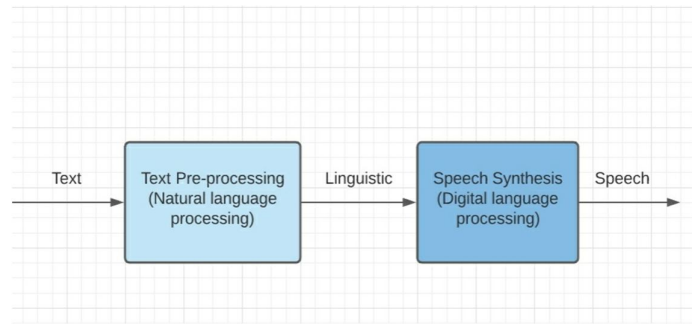


Figure 2: Speech Synthesis process

related things and organize them in a section, can update them and after discussing ideas among their team members, they can proceed to Drishti, a feature in Yojana which can be used by beginners, experts, software developers and even visually impaired people to code & develop software applications in more than 50 languages.

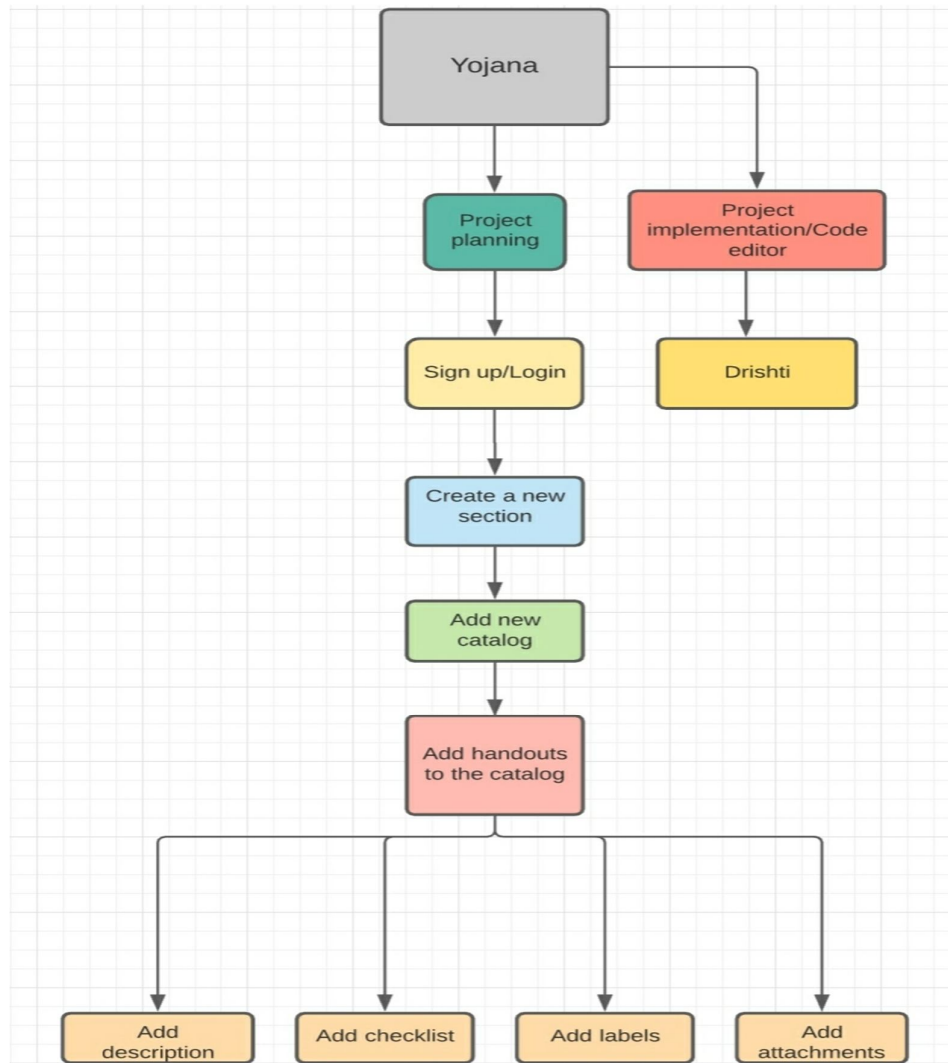


Figure 3: Proposed Model of Yojana

Users have to create their account on Yojana by signing up on it. After signing up, users have two options to go ahead - either they can directly move on to Drishti to code and develop software applications or if they need to discuss the project with their team members then they can create various sections in Yojana. Each section can represent a project & can be created by a team leader. Inside one section, any number of catalogs can be created and named. Catalogs present in a section represent all the steps to be followed to deliver the project. Catalogs can easily be moved by dragging and dropping. Inside a catalog, multiple Handouts can be created. Handouts represent different tasks needed to carry out inside that list. Name on the handout can be changed and a description can also be provided for that card. Labels can be added to the card for its unique identification and categorization. Leaders can grant access to their team members as per their requirement. Members who just need to follow the instructions are given view access and those who need to update the details can be given edit access by their team leaders. Checklist feature is also provided in a handout which represents the different steps required to complete the project. Various items or ToDo's can be added with checklists. Items can be added, updated or deleted by team leaders or members who have access to edit. Completed tasks can be checked so that handout will show that the no. of tasks completed out of total tasks. Various text files, audio & video files, documents, technical articles, presentations, links and even confidential data documents can be attached inside a handout and access can be granted to various members by the team leader. Handouts can be moved from one list to another list and their positions can also be changed within the same list. Catalogs and Handouts inside the section can be deleted and archived when not needed inside a project section. If the Handout or catalog is archived, it can be shown inside 'Archived Catalogs'/'Archived Handouts' section where users can find their archived cards and lists. Users can then decide if they want to delete that list/card or they want to restore them. Team leaders can also check their members' activity on a particular list and hence can track the project work easily. After discussing and organizing ideas, users can move on to Drishti to code the application.

A. Drishti

Drishti is a feature of Yojana which helps students, experts, and even visually impaired people to code & develop software applications efficiently without any external help. Drishti gives numerous options in terms of software development languages to visually impaired people. This helps the visually impaired people to code in the language of their choice & hence develop the applications easier & faster.

Problem

Drishti provides the complete development environment to software developers so that they can code their ideas into working applications. Apart from this, in today's world, many software developers include beginners: who want to start their career in software development & experts: who are experienced software developers; both of these categories include passionate visually impaired people who want to learn, grow and develop their career in the software development domain. They have a lot of opportunities today in the market which help them to type easily hence they are able to code easily with the help of existing hardware/software technologies in the market. But the problem arises when these existing hardware/software technology does not help these passionate developers know what code they have designed for their applications, how their designed software applications appear. If they want to check and look over their code that what they have designed they would need external help.

They would need the help of other people who would read the code for them which they designed. Similarly, If they want to know about how their designed application looks, in this case also they need to take external help. Apart from this, since they cannot see what they have built, how their application looks and if they want their application to appear in some specific way, then the question is: how would they know that the design of their application is the same as their decided design. For this too, they need external help. They will be in need of somebody who will help them know what they have designed. Then the developer will be able to decide whether the application built is correct or not. Everytime the developer needs to check the design, external help will be required which requires more and more time. This consumes a lot of time because every time somebody needs to be called for helping out the visually impaired person. Sometimes, the developer may not get external help due to many reasons which also badly impacts as the developer has to wait for somebody's help.

Features of Drishti

Drishti provides ease to all the software developers including visually impaired people to develop multi language supporting web applications. It also provides an excellent environment to visually impaired people to code & develop working software applications as well as test their applications on their own and at their own pace without any external help.

Drishti has 4 main features in the side navigation bar:

1. File Explorer
2. Upload
3. Speech synthesis features
4. Settings

All these features are exclusively designed for all the software developers while some unique features are designed specially for visually impaired software developers to help them code and develop software applications easily and provide them a healthy user friendly-environment. Non visually impaired software developers may/may not activate the features specially built for visually impaired developers according to their choice. The objective of this software is to create a single software development platform for all the software developers.

File Explorer: This section consists of a number of files created by the user while building a software application. This section is collapsible control section usually built for managing all the development files and the section includes:

1. **Adding/creating a new file:** This feature helps visually impaired people to add a new file into the existing folders or files. When the developers are developing a web application using multiple languages, they need to create different folders of the different languages. For example: If user builds a MERN Stack web application then, he/she needs to build a javascript(example: js) folder consisting of all the javascript files(example: index.js), a CSS folder consisting of all the design files, etc. necessary for building a software application. Visually impaired people can easily add files into an existing folder and start developing.
2. **Modifying the file name:** Developers usually modify their file names to some standard file names used in the development process. Some of the standard file names include: index.html, style.css, app.js, etc. Drishti provides this feature to help the developers change or modify their file names easily.
3. **Deleting a particular file:** Developers can use this feature of Drishti to delete any unnecessary files which they do not want to include in their project folders. Visually impaired software developers can easily delete any file from their list of their folders and test the changes in the browser.
4. **Downloading a particular file:** Drishti provides this feature to download any file or a set of files or a complete folder so that developers can save the files and their work locally on their machine for future use if they want to keep the work saved. Developers can easily make use of this feature to download the files and test the application on some other application development environments.

Upload: Developers can make use of this feature to upload any existing file which is on their local machine to Drishti if they want to reuse the code which they have designed earlier. Under this category, Drishti provides two features:

1. **Drop:** Developers can find this feature suitable if they want to drag and drop any file from any folder present on their local machine to Drishti and use the same file in their application folders.
2. **Choose:** Developers can easily choose any file using this feature which opens a dialog box on their local machine in which they can browse for the file they want to upload.

Speech synthesis:

1. **Speak Document:** This feature helps the visually impaired people to know what code they have designed to build a software application. When the developer hovers over this option in Drishti, Drishti speaks of this option as “Speak Document” and hence the developers will get to know where their cursor is. Then it allows developers to decide whether they want Drishti to read the whole document. If the developer wants to know the code he/she has written, the developer can click over that position(Speak document button) and hence Drishti will start to read the whole document line-by-line. This benefits the visually impaired to go through the design of their code on their own independently.
2. **Speak by Line:** This feature benefits the visually impaired people by letting them know what code they have typed on a particular line to build a software application. When the developer hovers over this option in Drishti, Drishti speaks of this option as “Speak By Line” [13] and hence the visually impaired software developer will get to know where his/her cursor is on their code. Then it allows the developer to decide whether he/she wants Drishti to read the code written in that line.
If developers want to know the code they have written in that line, they can click over that position(Speak by Line button) and hence Drishti first shows a dialog box which will ask the developer to input line number & then it will start to read the whole coding statement present in that line.
3. **Speak Console Result:** This feature will help visually impaired people to know what they have actually designed, that is the output or the developed application which appears on the browser. This will help them decide whether they built the exact application which they wanted to build without any external

help. This feature reads the whole built application word by word and helps the developer know what they have actually built and what were the requirements.

4. **Speak Console Log:** This feature will help visually impaired people to know what is appearing on the console log, that is used to test if the application developed is working fine. This will help them decide whether they have designed the exact code they wanted to, without any external help. This feature reads the whole console log word by word making the visually impaired developers independent of any external help.
5. **Stop Speaking:** Let's suppose the developer clicked over the "Speak by Line feature" and Drishti started speaking the code written in that line. Now if the user wants drishti to stop speaking at some moment, the user can click on this button & drishti will stop speaking the code.
6. **Search Cursor Position:** Visually impaired people usually find it difficult to search their cursor in the code. Drishti helps in this case. Using this feature, User will get to know about the current cursor position in the code.
7. **Cursor Jump to Line:** Whenever developers develop any application, they usually need to modify their code may be due to some bug, or due to incorrect output. This feature of Drishti allows the visually impaired developers to directly jump to a particular line wherever they want to start writing the code, or modify the code.

Settings:

1. **Voice:** Drishti allows the visually impaired software developers to control the rate and pitch while Drishti speaks. Users can control both the rate and pitch at which they want Drishti to speak. Drishti allows users to select a particular language in which Drishti will speak among multiple languages available [9] inside voice features.

Mode:

1. **Character Mode:** This feature will read the whole line of code character by character including each symbol for users. Users first have to activate this feature by clicking on the Character mode button and then users have to click on the run icon button displayed on the top right corner of Drishti. Drishti will start reading the line of code where currently the cursor is positioned.
2. **Code Mode:** This feature will read the whole line of code word by word for users. Users first have to activate this feature by clicking on the Code mode button [10] and then users have to click on the run icon button displayed on the top right corner of Drishti. Drishti will start reading the line of code where currently the cursor is positioned.
3. **Other features:** The features like overview mode and music mode are developed for providing a healthy peaceful environment to users. Music mode activates when the user clicks on the music mode button followed by clicking on the run icon button present in the top right corner of Drishti & there it plays music for a short duration of time for the users to make the coding environment peaceful and soothing.

Overview Mode helps users know that currently they are on which line in the code editor and it will also tell the number of characters present in that line.

Accessibility:

1. **Voice Feedback:** Drishti provides this feature so that developers can know where their cursor is on the code editor, which character they are on, etc, where they are writing the code [11].
2. **Voice Cue:** Drishti provides this feature so that developers can know when they hover over any feature on the navigation bar. Let's suppose the developer hovered over the file explorer icon, Drishti reads the icon button as "File Explorer" so that the visually impaired software developer gets to know that their cursor is currently over the file explorer button in the navigation bar. Now if they want to use the built Drishti features of the file explorer button in the navigation bar, they can click over that button. This helps the visually impaired people as they don't need any external support to make themselves know that where is the file explorer button, where are the voice controls and rest of the features. They can themselves navigate through the whole website and code the applications without any external support.
3. **Spotlight:** This feature of Drishti lets the software developers turn on or turn off the settings whenever they want to. This feature is also useful to visually impaired people because they can very easily change the settings without any external support. It mainly allows to activate or deactivate settings like "voice cue" & "character mode" which are the heart of Drishti web application.

Non visually impaired developers do not need to activate 'Voice Cue', 'Voice Feedback' and other features specially designed for visually impaired people. They can simply click over the feature and use it.

Theme: This feature is developed to make the development environment more user friendly. Two modes are supported in Drishti namely Light mode and Night Mode. Users can make use of this feature to change the coding environment modes as and when required.

Console:

1. **Console Bar:** Activation of this feature leads to the opening of the console bar which consists of :
 1. **Console:** Clicking on this button closes the console bar. While if users want to open the console bar again, they can re-click over the “console” button.
 2. **Run:** This feature enables users to test their code. After coding the whole application, The time comes to test and see the features of the application. Visually impaired software developers can simply click on the “Run” button to run the application and Drishti will show the output that is their designed web application.
 3. **Open in New Window:** If users want to see and test the features of their web application in a new window, then Drishti allows its users to click over the “Open in New Window” button and Drishti opens the web application in a new window where the user can navigate through the whole designed web application.
 4. **Open in New Tab:** If users want to see and test the features of their web application in a new Tab, then Drishti supports this feature as well. Users can click over the “Open in New Tab” button and Drishti opens the web application in a new window where users can navigate through the whole designed web application.
2. **Hide Console Log Area:** This feature allows the users to hide the console log area in the console bar and hence after activating this option, Drishti closes the console log area and displays only the resulting web application.
3. **Hide Rendering Result Area:** This feature allows the users to hide the Rendering Result Area in the console bar and hence after activating this option, Drishti closes the Rendering Result Area and displays only the Console Log.

Therefore, Drishti not only provides a complete software development environment to students & experts but also helps all the passionate visually impaired software developers to code, develop and test the software applications on their own without taking any external help.

Thus, using Yojana, developers can work first on their ideas, organize all their project ideas in one place, keep track of the team’s work as well as allow developers to code their extraordinary ideas into working applications. This software is thus saving their time and increasing their development efficiency. This software also allows visually impaired software developers to grow by themselves and develop numerous web applications on their own and hence serve their workplace or organizations independently.

IV. SCOPE OF YOJANA

Yojana is an all-in-one platform for discussion as well as execution of ideas by developing a software application in any coding language. Nowadays, various teams of any organization are working on developing world class softwares and projects. Before moving on with development of such projects which are going to impact billions of users, there is a need to define the project scope, set the project goals and objectives, determine the project deliverables [2], create project schedule and milestones, etc. With the help of Yojana, this can be easily done. Team leaders can create a section consisting of catalogs and handouts and plan a proper strategy with their team of how to create and analyze business cases [14], how to identify and meet stakeholders approval and ways to carry out risk assessment. All the project related details can be written in a handout inside a catalog and any number of people can be added to that handout for viewing/editing that handout. Organizations can grant view access to those employees who need to follow the updates and edit access can be given to the employees who are a part of project planning so that they can change any particular detail in a handout. Any update made to a section notifies the people added in that section immediately. Thus, with Yojana, there is a way to keep everyone in the loop on project updates.

Students, software developers and experts can easily discuss ideas with their team and then can move on to Drishti where they can easily code their ideas in a language of their choice. More than 50 development languages are supported in Drishti. Developers can plan, code and test their software applications easily. Drishti is also helpful to visually impaired software developers because they can code and develop easily without any external help. Numerous features are specially built for highly passionate visually impaired software developers who want to code and test their applications independently for their respective organizations. When visually impaired software developers hover over anything, Drishti reads that feature aloud and they get to know about that

particular feature and can hence decide whether to click over that feature or not. Drishti can also read the whole code line by line and can also tell where currently the cursor is positioned. Apart from this, Drishti can move the cursor to a position of the user's choice. Not only this, Drishti can also read the console log and can also describe the whole developed application/output in the user's chosen language. This saves a lot of time and increases the development efficiency of visually impaired software developers because they don't need any external help. Drishti supports a lot of features for such passionate software developers. Thus, Drishti is a single platform for all the software developers which helps them build world class projects easily.

Apart from project planning, Yojana is also beneficial to various technical/non-technical event organizers because they can easily discuss and write all the details in a handout and grant access to all other organizers of the event to follow for any updates. Yojana is also helpful in strategic planning/annual planning, staff meeting agendas, any personal work, communications calendars, presentation preparation, board & committee management, etc.

Yojana is even beneficial to school going students who work on multiple assignments and projects in a team. Yojana helps them split up their no. of tasks and plan out with their team to meet their goals. They can also add various ToDo's, completed work, etc. in a handout and hence their progress [15] can be tracked in real time. It is a great tool for synchronous or asynchronous communication, active learning, cooperative learning, and conducting research.

It is also beneficial to multiple Government agencies like the Ministry of home affairs, Ministry of labor and employment, etc, because they regularly need to plan numerous things and take big decisions. The feature of granting access to people at different levels is very helpful to government agencies because they maintain a lot of confidential data and documents. They can easily create multiple sections each with a different goal and write essential information in handouts inside a catalog and access can be granted to people as per the need in an easiest and fastest way. This also helps them maintain private information easily. Yojana can also help various disaster management organizations to discuss various disaster reduction steps and make a proper strategy to ensure an effective and coordinated response to disasters. Yojana is a great platform for a variety of work be it educational or personal uses, from school bulletin boards, lesson planning, collaborative student research projects and task management to planning and organizing various non-technical meetings. The best part of Yojana is its collaboration-ability – sections can be viewed, edited, and managed by multiple members so classrooms, teams, or organizations can keep track of everything.

V. CONCLUSION

Yojana benefits a large number of users in planning and prioritizing things and they can empower their whole team to take an interest in some significant conversations, both groups gathering and one to one meetings. They can also send any number of notes, share documents, presentations and even let them remark on their individual projects and tasks. Users can discuss their ideas as well as code their ideas to working applications with the help of Drishti. It is also helpful for the people who are not able to contribute in the field of technology due to the scarcity of opportunities like visually impaired software developers. Distinctively, Studies have shown that visually impaired people have a vision that is unique and have the potential to unfold new dimensions in any web development project. Due to their visual impairment they are not able to manifest their ideas but with Drishti they can code and show their ideas to the world and can also get employment in any organization where they can work independently. Yojana is an amazing tool for individuals or companies who enjoy using a visual project management tool with plenty of opportunities for integrations. For individuals or companies who need a tool that encourages collaboration, they can get more done in a very efficient manner. Yojana is a great platform for staying organized and getting work done. It is useful for multiple aspects of a business/organization. It can act as a tool for managing projects, allowing users to monitor and track their progress throughout the lifecycle of the project. Using yojana is a simple way to break up an intimidating big task or project into smaller pieces. Yojana can also be used to manage ongoing processes and tasks, such as monthly finance admin, human resource activities, training, sales and research.

REFERENCES

- [1] M. R. Solanki, A. Dongaonkar, A Journey of human comfort: web1.0 to web 4.0, International Journal of Research and Scientific Innovation (IJRSI), Volume III, Issue IX, pp. 75-78, 2016
- [2] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, Image analogies, in Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech., 2001, pp. 327-340.

- [3] Bozickovic, H., Stula, M. (2018). Web design Past, present and future. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).
- [4] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring Punctuation and Capitalization in Transcribed Speech," in Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4741–4744, 2009
- [5] P. Kabal, "TSP Speech Database," tech. rep., McGill University, 2002.
- [6] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," IEEE Internet Computing, vol. 14, no. 6, pp. 80–83, 2010. [Online]. Available: <http://doi.ieeecomputersociety.org/10.1109/MIC.2010.145>
- [7] Bray, "The JavaScript Object Notation (JSON) Data Interchange Format Interchange Format," 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7159>
- [8] Jackson, S. E., Chuang, C.-H., Harden, E. E., Jiang, Y., & Joseph, J. (2006). Toward developing human resource management systems for knowledge-intensive teamwork. Research in personnel and human resources management, 25(6), 27-70.
- [9] Y.-Y. Wang, A. Acero, and C. Chelba, "Is WordError Rate a Good Indicator for Spoken Language Understanding Accuracy," in IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 577–582, 2003.
- [10] Edward Loper, and Steven Bird "NLTK: The Natural Language Toolkit", Department of Computer and Information Science University of Pennsylvania, Philadelphia, 2002.
- [11] H. R. Silbiger and J. L. Sullivan, "IEEE Recommended Practice for Speech Quality Measurements," IEEE Transactions on Audio and Electroacoustics, vol. 17, no. 3, pp. 225{246, 1969}.
- [12] V. Mitra, W. Wang, and H. Franco, deep convolutional nets and robust features for reverberation-robust speech recognition, in Proc. of SLT, pp. 548553, 2014
- [13] T. Hain, A. el Hannani, S. Wrigley, and V. Wan, Automatic speech recognition for scientific purposes – webASR, in Interspeech 08, 2008, pp. 504507.
- [14] Leach, D. J., Wall, T. D., Rogelberg, S. G., & Jackson, P. R. (2005). Team autonomy, performance, and member job strain: Uncovering the teamwork KSA link. Applied Psychology, 54(1), 1-24.
- [15] Erchul, W. P. (1992). On dominance, cooperation, teamwork, and collaboration in school-based consultation. Journal of Educational and Psychological Consultation, 3(4), 363-366.