

# DRISHTI: VOICE BASED EMAIL SYSTEM FOR VISUALLY IMPAIRED

Neha Gupta<sup>1</sup>, Nitin Chauhan<sup>2</sup>, Nirbhay Pal<sup>3</sup>, Mayank Bhatnagar<sup>4</sup>, Mohd Fardeen<sup>5</sup>  
Department of Computer Science and Engineering, Moradabad Institute of Technology,  
Moradabad, India  
nitinchn1998@gmail.com

## ABSTRACT

*A voice-based email writing and management system will facilitate visually impaired people to access and write emails in a problem-free manner. This system will enable visually impaired people to navigate through their emails and manage them using their voice commands eliminating the need for touch or mouse gestures which often hampers the usability of similar solutions for visually impaired people. This system greatly reduces reliance of visually impaired people on others for their activities regarding emails. Pre specific voice commands are used to perform actions in the system such as writing emails, checking unread emails, etc. In this application, we propose smartphone-based application working on either IOS or android operating system.*

**KEYWORDS:** *Drishti, Assistive email client, Email writing for visually impaired, Voice based email client, Social cause, Cross platform development.*

## 1. Introduction

Globally, the number of visually impaired people of all ages is estimated to be 285 million [1], of whom 39 million are completely blind. These people are usually dependent on their family members and close ones to aid them in their day to day tasks. Making these people self-reliant through technology is one of the greatest challenges in the technology world. The application proposed here is a little effort in the direction of making visually impaired people self-contained. We are proposing a smartphone-based system/application that allows the user to exchange emails without manually typing. The application is designed to enable users to do all email related work through voice commands such as writing emails, reading email, deleting emails etc. This overcomes the challenges faced by visually impaired people when they need to contact someone officially or want to write email for their clients as sometimes email is the only officially recognised way of communication to businesses or clients. The application's voice based user interface provides great user experience to the users and reduces their reliance on other peoples. The application contains a voice-based user interface to manage user's email related needs. This application is different from the standard system of email as generally we use touch gesture based systems which are inconvenient for blind people, these solutions are often not user friendly due to their dependence on touch gestures. Hence, we aspire to provide them a medium where they can do emailing with the help of their voice where there is no requirement of touch gestures. Touch gestures are often tricky to use and sometimes it becomes very difficult for a blind person to estimate their device's orientation. They face trouble in accessing the system as they cannot use it properly since it requires the visionary connection between the application and user, without which it is impossible to operate it. The native smartphone OS such as android does provide accessibility features, but they often become ineffective as the application which the user is trying to use is not optimised for their special needs. So, to overcome these challenges faced by people using accessibility features, the application is designed in such a way that makes it more accessible and easier to use compared to native accessibility features.

## **2. Module description**

### **2.1. Command module**

This is the primary module which handles the major functioning of the application. If the user wishes to write, read aloud, update, sort or listen to the email which has been written then he/she needs to command the system through this. This module will eventually transfer the control to whichever module has been called by the user. This basically provides the interaction of the user with the system and works as a bridge between the user and the application. There are predefined commands for specific tasks such as writing emails, checking unread emails, checking emails from specific people (specific email addresses) etc. The command module transfers control to module required as per user requested action and once the action is completed or user interaction is required by the module the control is again transferred to the command module

### **2.2. Event Listener**

Even though the application is completely voice based but for ease of access for people who are coming from other applications, the application provides an easy way to access the help menu to know about all available commands to navigate through the application and use it proficiently. And if the user receives any new mail then he will be informed (through audio) about that with the help of this module. So, basically any kind of notification which the user receives, it will be automatically reported to him. Then, he can decide whether to visit it or not

### **2.3. Writer**

This module is called when the user request for writing email is received by the command module it enables the user to write email address(es), subject to the email and email content. This recognizes the voice of the user and uses the speech to text conversion API to extract text from the user voice. This module only converts speech to text and redirects users to the command module in case any user interaction is required. For example when user is redirected to this module it sends request to the command module to ask user for email address(s) to whom email is to be sent and when user speaks after the command module ask for the email address(es) the voice is converted to the text from which email address(es) is(are) extracted and added to recipients list and again command module is summoned to ask user for subject of the email and so on. This module basically writes everything which the user wishes. If he/she wants to add a new paragraph or any type of punctuation the module can recognize user commands for them. The accuracy of the voice to text conversion depends upon the API provided by the OS on which the application is installed on.

### **2.4. Reader**

This module is summoned whenever the system requires to read any text to the user such as reading email etc, it uses the text to speech conversion API provided by the host OS. The user needs to speak the specified command to the command module which is predefined and then the application will start reading aloud the text as required by user request. This module also provides functionality to verify the email written by the writer module so that the user can send email which is free from any error and is exactly what he wanted. And if he finds any mistake or error then he can simply ask the application to stop there and perform changes (to perform changes writer module is called). This is also useful in listening to other emails (previously sent or received). The user can select any email which he wishes to look through and then command the system to read it for him. The voice of the system which it uses in reading is provided by the host OS which may or may not have fine controls to change synthesised speech speed or modulation and the accuracy of text to speech conversion depends upon the API.

### **2.5. Authenticator**

This module provides user authentication to securely provide access to registered users only, authentication is done through biometric data (fingerprint or face) using host OS provided APIs and device hardware capabilities. Providing security to visually impaired people is a necessity because they can be easily victimized by the malignant persons.

### **2.6. Help Guide**

This module, as the name suggests is used for providing help to users in understanding the commands to get familiarised with the system and use it efficiently. This module gives the description of all the commands which a user can use. The user can access this by double tapping anywhere and anytime  
Doi: [10.5281/zenodo.5139573](https://doi.org/10.5281/zenodo.5139573)

on the screen (as shown in fig 7) in the application whenever required. The user will get a proper understanding of how he can easily access the system and if he forgets any command then this will simply recall all the commands to the user.

### **3. Process Flow**

The proposed approach is described in the figure 1. When user opens app for the first time he/she is prompted to ask to their family member or closed one to help them is setup process(shown in fig-3) as host Oss requires physical intervention of user to grant permissions required by the application to work properly(as shown in fig-4, 5 & 6) on host OS, apart from that user id and password needs to be inputted manually and biometric authentication setup is also done here(if sensor is available). Once setup is completed user when opens application either through using voice assistant or any other assistive launcher user is prompted for biometric authentication once verified user is given access to the application. By default, on successful user login, the user is presented with summary of activities in their mailbox such as count of new mails, replies received for user sent emails etc. Command mode is also active by default waiting for user commands to process. When user command is received it is parsed; if command is correct user appropriate action is taken else user is prompted to repeat command or double tap to open help menu.

### **4. Available Commands**

#### **4.1. Create new E-mail**

This command opens new email dialog, user is asked for email address of recipient, which is verified for proper email address format, if email address is wrong then user is asked to dictate email address again else user is prompted for subject of the email and email body contents. Once email content is transcribed user is prompted to review email contents. Email contents are read aloud by the system to the user if user finds any error user can stop the cursor wherever the system is currently reading at and user can edit it by giving appropriate commands. Once email is verified by user it sent to the given email address.

#### **4.2. Read new emails**

This command reads aloud the email addresses and their respective subjects of all the new emails received since last user login. User can stop this at any time using appropriate commands.

#### **4.3. Delete**

As the name suggest this command is used for deleting the email which is currently opened (being read to user). User is prompted for confirmation for deletion once confirmed email is moved to bin.

#### **4.4. Change Folder**

This command is used to navigate throughout the user mailbox to go to different folders user mailbox have such as starred mails, spam, sent, outbox etc.

**4.5. Search** This command is used for searching throughout the user mailbox. When this command is given user is asked for search criteria such as search by date, search by email address etc.

### 5. Flow Chart

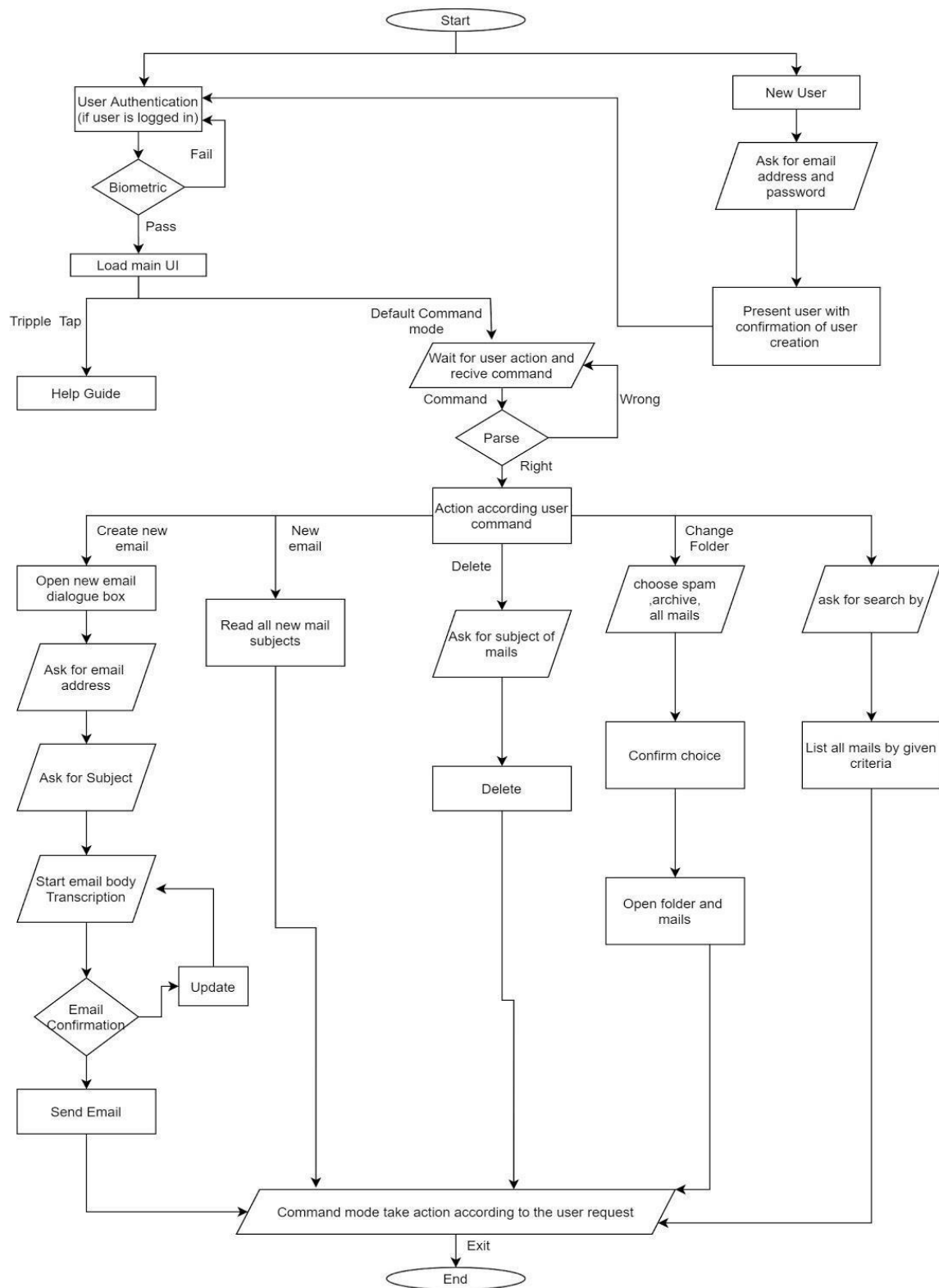


Figure 1. Process Flow

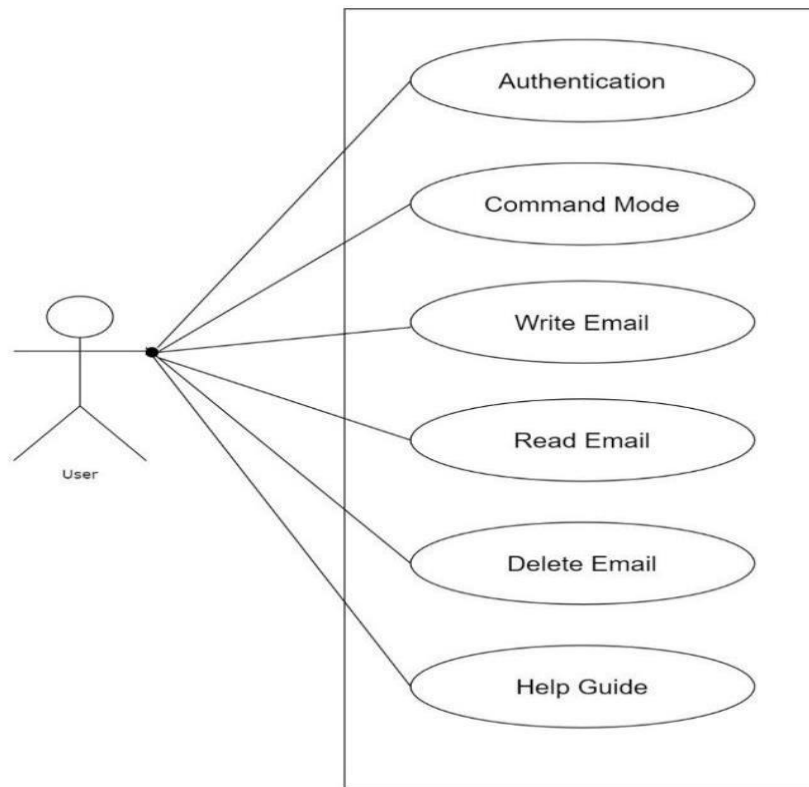


Figure 2. Use case

## 6. Software Details

The application is developed by using Flutter SDK [2] which enables to create native applications for multiple OSs from the same code base saving a lot of effort of porting apps to multiple platforms. The source code of the application is written in dart language [3] which supports both just-in-time (JIT) [4] and ahead-of-time (AOT) [5] compilation techniques which enables to visualize changes made to the application source code in real time and saves time in development of application [6]. The application is primarily developed for mobile devices, but it can be compiled for desktop platforms also as flutter SDK support both mobile and desktop platforms. For text to speech conversion flutter package flutter\_tts [7] is used, for speech to text conversion flutter package speech\_to\_text [8] is used and for authentication through biometric sensors, flutter package local\_auth [9] is used. These packages work as abstraction of native operating system libraries of host OS to the application and hence provides clutter free way to use services provided by the host OS instead of re-implementing them.

## 7. Advantages

1. This system helps in decreasing the social gap between the visually impaired and normal people.
2. It helps in the advancement of the lifestyle of unsighted people.
3. It helps them to gain employment.
4. This system is accessible by a person of any age group.
5. This application can be used by all types of people including regular persons, visually compromised as well as illiterate.

## 8. Mock-ups



Figure 3. Welcome to Drishti

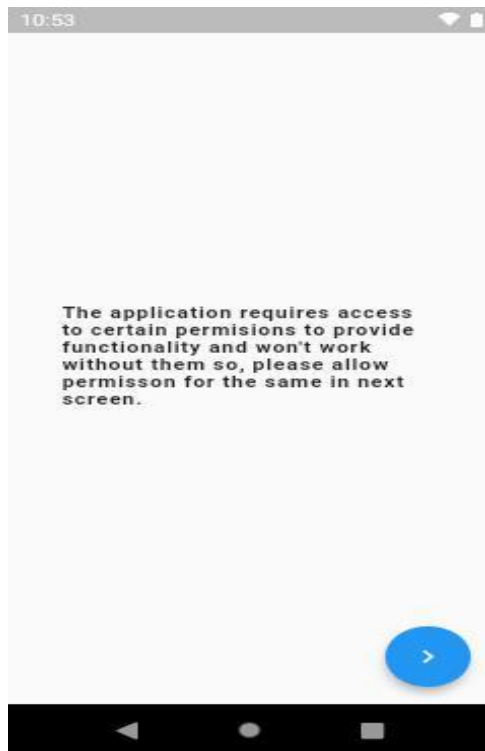


Figure 4. Access permissions

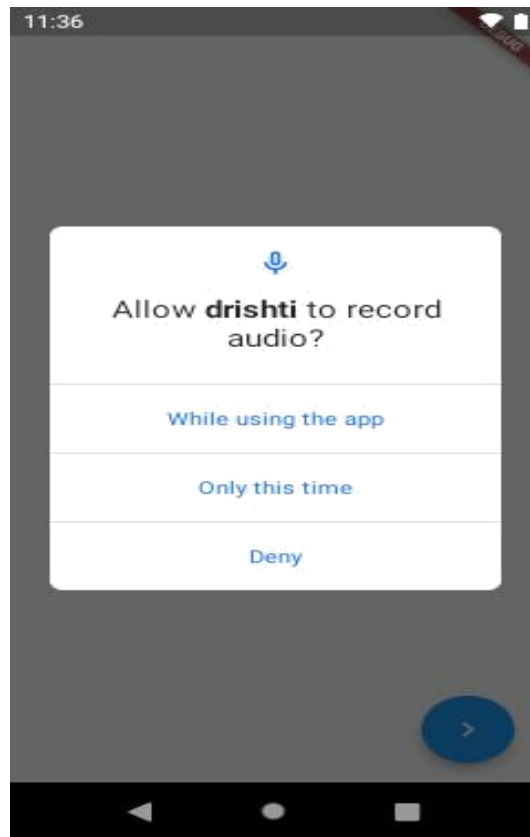


Figure 5. Allow the app to record audio

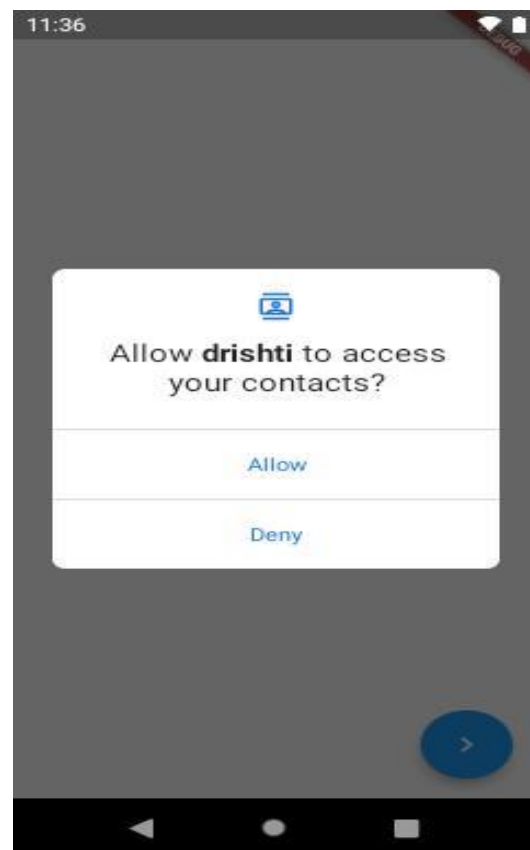


Figure 6. Allow app to access your contacts

Doi: [10.5281/zenodo.5139573](https://doi.org/10.5281/zenodo.5139573)

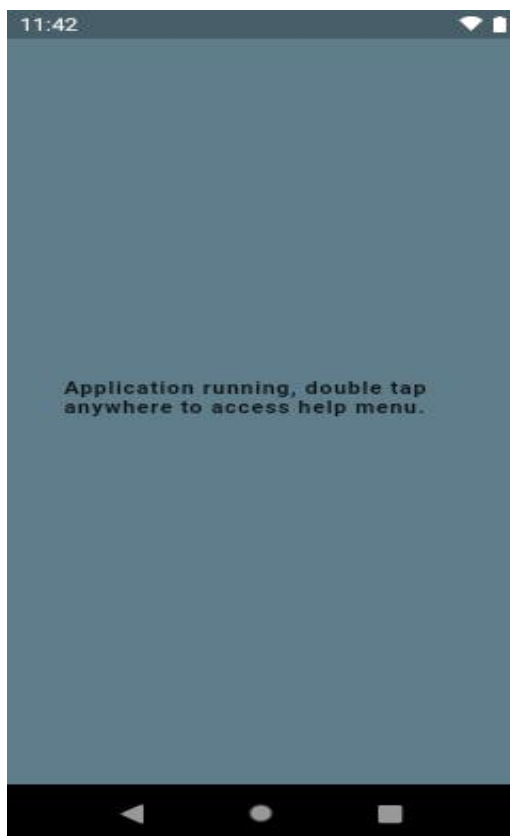


Figure 7. Double tap to access help menu

## 9. Conclusion

The unsighted people are a part of our society and it would be disgraceful for us and unfair to them if we cannot even make them able to write emails without the help of another person. Some reputed people who meet accidents and lose their sight get prone to cybercrimes. And hence, it is our responsibility to provide them a medium which provides the comfort of exchanging emails with ease and security.

This Email system will be beneficial for the visually impaired people to grow along with our society as it has a feature of speech to text, text to speech with speech reader which makes this system convenient to use by visually impaired person. This application provides the facility of writing, reading, updating, listening the emails and biometric security. The user is provided with a help guide so he/she will not require any help from another person in case he faces any trouble. This application helps them to be a part of growing digital India by allowing them to communicate through internet and making life of such people easier.

Also, if this project comes out to be a great success which we hope it to be then, we as developers have a motivation to make more systems like this to contribute to our society.

## 10.Future scope

As of now planned features are limited and primarily focuses on most basic and essential needs of user regarding email management but application has great potential to grow and add more features such as adding attachments, provide user with ability to open web links attached in the email in application itself to make their experience seamless by providing all feature of the application such as text dictation on the web page, filling forms on the web pages etc.



## REFERENCES

- [1] Pascolini D, Mariotti SPM. Global estimates of visual impairment: 2010. British Journal Ophthalmology Online First published December 1, 2011 as 10.1136/bjophthalmol-2011-300539.
- [2] Flutter SDK <https://flutter.dev/>
- [3] Dart Language <https://dart.dev/>
- [4] Just in time (JIT) [https://en.wikipedia.org/wiki/Just-in-time\\_compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation)
- [5] Ahead of time (AOT) [https://en.wikipedia.org/wiki/Ahead-of-time\\_compilatio](https://en.wikipedia.org/wiki/Ahead-of-time_compilatio)
- [6] Hot reload flutter <https://flutter.dev/docs/development/tools/hot-reload>
- [7] “Flutter tts” package documentation [https://pub.dev/packages/flutter\\_tts](https://pub.dev/packages/flutter_tts)
- [8] “Speech to text” package documentation [https://pub.dev/packages/speech\\_to\\_text](https://pub.dev/packages/speech_to_text)
- [9] “local auth” package documentation [https://pub.dev/packages/local\\_auth](https://pub.dev/packages/local_auth)

## Authors

**Neha Gupta**, an assistant professor at Moradabad Institute of Technology with an experience of 13 years. Pursuing Phd and an enthusiastic research scholar.



**Mohd Fardeen** linux enthusiast, currently pursuing B.Tech from Moradabad Institute of Technology. Love working on open-source projects and android development.



**Nitin Chauhan**, final year student of B.Tech from Moradabad Institute of Technology. Winner of Smart India Hackathon 2020. Interested in competitive programming and app development .



**Mayank Bhatnagar**, an engineering student of 4th year with specialization in computer science. Interested in front end and android app development.



**Nirbhay Pal**, currently a 4th year student of B.Tech pursuing from MIT, Moradabad. Keen enthusiast of web development and linux os administration.

