DEVELOPING REPOSITORY OF USER-CENTRIC TOOLS USING FOSS

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

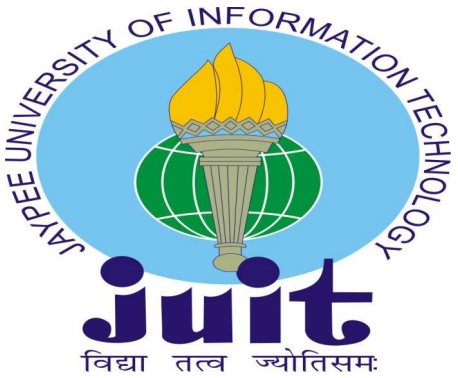
By

Prishita Singh (191223)

Under the supervision of

Dr. Pradeep Kumar Gupta

to



Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Developing repository of user-centric tools using FOSS."** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my work carried out over a period from January 2023 to May 2023 under the supervision of **Dr. Pradeep Kumar Gupta** (Associate Professor, CSE & IT). I also authenticate that I have carried out the above-mentioned project work under the proficiency stream **Cloud Computing.** The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)
Prishita Singh (191223)

This is to certify that the above statement made by the candidate is accurate to the best of my knowledge.

(Supervisor Signature)
Dr. Pradeep Kumar Gupta
Associate Professor
Computer Science & Engineering and Information Technology (CSE&IT)
Dated:

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: _____ __Department: _____ Enrolment No _____

Contact No. _____E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

Acknowledgment

Firstly, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing that made it possible for us to complete the project work successfully.

I am really grateful and wish my profound indebtedness to **Dr. Pradeep Kumar Gupta,** Assistant Professor, Department of CSE Jaypee University of Information Technology, Wakhnaghat. His deep knowledge & keen interest in the field of Cloud Computing helped us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would also generously thank each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. I would also like to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Prishita Singh(191223)

Table of Content

List of Figure

List of Tables

Abstract

The project “Developing repository of user-centric tools using FOSS” is an open-source project which provides the freedom to the developers to modify the code according to their needs. Any type of program whose creator elects to make the source code available without charge is considered open-source software. Anytime software has an open-source license, anybody can download, change, and distribute it without having to pay the software's original inventor any money.

Our perspective towards the project is to create a repository of tools, developed using Free Open Source Software, that is user-friendly in nature. Such user-friendly tools can be used by anyone for their personal use without having to spend extra money on buying similar proprietary tools available in the market. Open Source will be crucial to modern computing, especially in the creation of applications and IT infrastructure.

This project is a contribution towards the FOSS community and intends to promote the FOSS culture, inviting people to use the tools enclosed at zero cost and improvise them at the same time. It will enable users to download the tools multiple times in different locations without restrictions or monitoring.

CHAPTER 1: INTRODUCTION

1.1 Introduction

Open-source software increasingly impacts a vast variety of users directly or indirectly. Many technical users, such as software developers prefer open-source solutions over proprietary ones, they install open-source software on their PCs and use them directly.

Many issues related to the purchase and use of information technology can be resolved by using Free and Open Source Software (FOSS), which has been identified as a key technique.

FOSS is a method for developing, deploying, and maintaining large software systems globally that varies in fascinating ways from the concepts and techniques that have already previously been promoted for software engineering.

1.2 Problem Statement

Source code is the raw information that is transformed into an executable program. In the case of proprietary software (like Microsoft Windows, Adobe Photoshop, etc.), source code is closely guarded by software companies.

Open-source software shares the source code such that anyone may reproduce the program or alter it to meet their needs. Therefore, open source does not usually cost any money to purchase and is easy to customize for a specialized task

The specific demands of the product's users are the primary focus of the user-centered design. The goal is to modify the product's design to meet the needs of the user and application.

Existing techniques for assessing perceived usability are frequently restricted to a single component, and expanding them to cover additional elements is labor-intensive. A generic process for assessing perceived usability is unavailable.

FOSS can be used -

- To reduce the costs associated with proprietary software.
- In promoting technological development and bridging digital gaps.
- To promote the culture of open-source software.
- By educational institutes of the software for academic and administrative purposes.

1.3 Objectives

The use of Free and Open Source Software (FOSS) has been identified as an important strategy for solving many of the issues associated with the use of Information Technology.[1]

The primary objective of developing a repository of user-centric tools is to increase access by different users with ease.

In contrast to proprietary software, where the software is under restrictive copyright licensing and the source code is usually hidden from the user, in FOSS the source code is openly shared so that people are encouraged to voluntarily improve the design of the software.

Open-source software and free open-source software are often treated as the same. However, there are differences between them concerning the licenses assigned to the respective software. OSS may use the General Public License(GPL) or some other license that allows integration of software which may or may not be free, whereas FOSS on the other hand is usually licensed with the GNU GPL.

1.4 Methodology

Open source code is typically published and kept in a public repository. Anyone with access to the repository is welcome to utilize the code on their own or add enhancements to the project's overall functionality and design. For this project, we chose GitHub to create a repository of the different user-centric tools that will be developed throughout the project. At first, the user-centric tools will be developed, following which, a repository will be created using GitHub.

1.5 Organization

The rest of this report is divided into the following sections. The work related to Free Open Source softwares and the various techniques used to situate the work in this project report in reference to the literature currently in use are discussed in the following chapter.

The system development from a backend perspective is thoroughly covered in Chapter 3 along with the technology used. Chapter 4 is devoted to analyzing and contrasting the performance of the suggested approach. Chapter 5 of the project report provides a review of the key findings and upbeat predictions for the future.

CHAPTER 2: LITERATURE SURVEY

2.1 Free/Open Source Software Development: Recent Research Results and Methods.

As per promoters of free programming like Richard Stallman and the Free Programming Establishment, Open Source Software Development (OSSD) is a product improvement technique. Open Source Software (OSS) is likewise seen by certain eyewitnesses as a social development that is independent of yet associated with free programming development.

The capacity to remotely get to source code, leave it open to examination and adjustment, and make it accessible for reallocation to others with few limitations — besides the option to guarantee these opportunities — is the distinctive component of free programming, and most of the open source programming.

Contingent upon whether OSS copyright and end-client permitting understanding is associated with a specific OSS code base, OSS infrequently adds or disposes of comparable opportunities or copyright honors.

Basically, OSS is dependably a type of free programming yet not generally the situation. Along these lines, it is much of the time suitable to utilize the terms FOSS or FLOSS (L for Libre, as the elective name "libre programming" is popular in a few districts of the world) to envelop two similar or regularly indistinguishable strategies for programming creation.

Thusly, in spite of the fact that product authorization may affect them, the focal point of this article is on FOSSD methods, cycles, and elements as opposed to programming licenses. In any case, where essential, certain examinations canvassed in this audit might be expressed in wording elite to

either free programming or open source programming (OSS), contingent upon the circumstance. [2]

Linux ends up in an excellent situation to dislodge Microsoft Windows as legislatures from everywhere in the world search for options in contrast to Microsoft programming. Linux is utilized by the state-run administrations of France, Germany, the Netherlands, Italy, Spain, and the Unified Realm to help open guidelines, animate decentralized programming advancement, improve security, and lower the expense of programming.

Since Chinese specialists have total admittance to the source code and can test it for security issues, the Chinese government forcefully empowers Linux as their working arrangement of decision. Radicals assembled in Brazil to support the use of open-source programming.

Also, confidential programming firms' standard working techniques presently remember open-source programming rehearses for the type of advancement approaches like Outrageous Programming (XP). In these circumstances, the improvement strategies put an emphasis on fast prototyping procedures and successive delivery cycles that impersonate the errand of consistently testing new programming emphasis.

Methods like XP frequently still rely upon the endeavors of few experts working in a shut climate, a long way from being a really democratized programming improvement philosophy that looks to effectively include a sizable client base in the improvement cycle.

There are a rising number of studies that offer some comprehension or experiences about FOSSD rehearses, every one of which thus ponders a few cycles that are presently ineffectively perceived. The accentuation is on experimental examinations of FOSSD drives utilizing little and enormous scope research tests and insightful strategies from different scholarly fields.

Various Web areas for research articles that tentatively or reasonably explore FOSSD projects have various different investigations of FOSS. They incorporate those at the MIT FOSS research local area entrance, which has gotten around 200 papers up until this point, and those at Plug School in Ireland, which connects to various unique issue diaries and the procedures of worldwide studios on FOSS research.

2.2 Free/Open Source Software Development: Recent Research Results and Emerging Opportunities

Rather than endeavoring to review the whole collection of studies contained in these assortments, the choice is to test a more modest gathering of review that bring up fascinating issues or troublesome issues for understanding what impacts how FOSSD endeavors are done, as well as what sorts of socio-specialized connections arise en route to work with these endeavors.

Nonetheless, the people who participate in, add to, and join FOSS projects regularly act in manners that are worth laying out validity and notoriety, accomplishing "nerd distinction", being imaginative, as well as giving and being caring with one's time, information, and source code.

The Tigris.org open source computer programming local area entry, participating in such a task is a down-to-earth methodology to hold or propel programming improvement abilities on account of FOSS for programming plan frameworks. One more technique to fabricate social capital and gain regard from peers is to turn into the fundamental player (or hub) in an informal organization

Do FOSS engineers work "for nothing" or to serve their expert and professional advancement? Members will commit their time and work to the sporadically difficult and requesting obligations of programming improvement for an assortment of individual and expert profession-related reasons.

Self-assurance, peer acknowledgment, project association or personality, self-advancement, and conviction in the intrinsic worth of free programming are a couple of these.

No one has the managerial ability to teach a venture part what to do when to do it, how to make it happen, or why in the self-assurance process. FOSS designers are allowed to chip away at projects that provoke their singular advantages.

As per one examination, somewhere around 60% of FOSS designers work on at least two ventures, while just around 5% work on at least ten. In any case, a venture's center designers frequently produce the extraordinary majority of the source code that is remembered for the FOSS that is made openly accessible.

They additionally commonly have command over the task's plan and advancement technique. Most of the members typically just add to one module, however, a little level of modules might contain patches or changes made by many supporters.

End clients habitually take on new jobs as engineers or partners in FOSS projects. Most FOSS engineers are likewise end clients of the product frameworks they make, consequently, there might be an expert thought process and stake in the progress of such frameworks.

By far most of the members, in any case, presumably really like to utilize FOSS frameworks, until or until their utilization motivates them to make a move by making a commitment. A committed client with fundamental specialized abilities may truly go through every job and lastly become a center engineer (or "senior").

To create, use, and develop FOSS frameworks, members inside FOSS projects as often as a possible interpretation of different jobs inside both specialized and informal organizations.

There are a rising number of studies that offer some knowledge or discoveries on FOSSD rehearses. These investigations are to test a bunch of reviews that bring up fascinating issues or troublesome issues for understanding what impacts how FOSSD endeavors are achieved as well as what sorts of socio-specialized connections arise en route to work with these endeavors.

Subsequently, as opposed to surveying the examination discoveries given beneath as main traits of most FOSSD tasks or cycles, accept them as take-off points for additional exploration.

To get to or take part in a FOSS improvement project, FOSS designers, end clients, and different workers habitually give their very own registering assets. They also offer their own Web association, and they could try and host individual Pages or instructive information bases.

FOSS software engineers likewise bring their own arrangement of instruments and improvement strategies to a venture. The proceeding with the project, its common (public) information relics, and the last open source code are totally upheld by supported individual asset speculation. It disseminates the expense of building and keeping up with the virtual association's data foundation, which makes up a FOSSD project.

The data sources and curios that members use to depict, forbid, or recommend what's going on in a FOSSD project are known as programming informalisms[5]. They are informal account materials that are very easy to utilize and accessible to every individual who needs to partake in the undertaking or simply glance around.

Then others show how programming informalisms could supplant formalisms like "necessity determinations" or programming plan documentations, which are believed to be significant by the SE people group to fabricate great programming. Nonetheless, these product informalisms often catch the inside and out legitimations and conversations behind adjustments made to specific improvement cycles, antiques, or source code documents.

The most well-known informalisms utilized in FOSSD projects are

(I) email list correspondences and messages,

(ii) strung message conversation discussions, release sheets, or gathering online journals,

(iii) news postings,

(iv) project digests, and

(v) texting or Web hand-off talk. Alongside

(vi) traditional framework documentation and

(vii) outside distributions, they additionally contain

(viii) situations of use as connected Sites,

(ix) how-to manuals,

(x) daily agendas,

(xi) FAQs, and other organized records, and

(xii) project Wikis.

(xiii) FOSS project property licenses are lawful arrangements that indicate what sorts of programming and related project materials are safeguarded assets that can then be shared, seen, refreshed, and reallocated.

Complex FOSS source code and application advancement call for responsibility and trust between project members. Despite the fact that trust and obligation might be elusive assets in a FOSSD project, proceeding with programming and undertaking improvement work possibly happens when these immaterial assets and frameworks of social control are set up.

Programming improvement action seems to be hypothetically concentrated in a FOSSD project meritocracy, yet is genuinely scattered in an independent and decentralized manner.

Nonetheless, in spite of the way that these marks have been utilized to portray different techniques for organizing FOSSD projects, it is neither just a "basilica" nor a "marketplace." All things considered, virtual venture the board (VPM) is utilized by meritocracy when it runs as a virtual endeavor to prepare, coordinate, control, create, and guarantee the nature of FOSS improvement endeavors.

There is a large number of OSS advancement projects in the OSS development, each with its own site. The vulnerability exists on whether the OSS development ought to be viewed as a counter-development to the restrictive or shut source universe of business programming improvement, or basically another computerization pattern.

For example, leaders from organizations that just foster restrictive programming have said that open source programming (all the more definitively, programming delivered under the GNU Public Permit, or "GPL") is a disease that spreads to protected innovation [2]. Other business sources,

then again, appear to completely contradict such orders and think about OSS as a possible region for key speculation.

In general, obviously, FOSS is progressively affecting society at the modern, legislative, and global levels, and it is doing as such in a manner that no other programming innovation or advancement strategy has even verged on accomplishing. This is on the grounds that free programming and OSS have worked with the rise of enormous-scope social (or computerization) developments.

Further examination is expected to decide why this has occurred, what influence it will have on FOSS going ahead, and whether corporate or public (government) strategy drives will progressively zero in on the creation, reception, arrangement, use, and backing of FOSS undertakings and applications. [3]

Yet, it is likewise clear that it is getting less and more uncertain that any association, state, or nation will actually want to stop the spread of FOSS or the FOSS developments in the public eye in the short-and medium-term.

The little example of FOSSD research referenced above is taken from a more extensive gathering of studies that have been inspected somewhere else. Although elective exploration philosophies utilized in different observational investigations of FOSSD are not analyzed in this review, it is essential to take note that these examinations utilize various experimental systems.

These incorporate multi-modular demonstrating and examination of FOSSD socio-specialized association organizations, methodical studies, ethnographic

exploration, mining of FOSSD antiquity or informalism files, and intelligent practice and industry surveys.

Initially, a singular engineer's advantage, inspiration, and commitment to a task and its givers are dynamic and not endless [cf. 49] with regards to taking part in, joining, and adding to FOSS projects.

A feeling of double-dealing by others can rapidly break up a member's obligation to additionally contribute, or more regrettably, to convince different members to leave an open source project that has gotten sidetracked.

Second, with regards to participation, coordination, and control, FOSS projects are not excluded from conflicts about who will deal with what or who will refresh and change it.It is likewise a cost they pay for convincing and wrangling with others to acknowledge their convictions and goals, regularly over web communications.

An arranged socio-specialized organization of reliance must be fabricated and kept up with the time, exertion, and center put into exchange and refereeing.

Third, the improvement of associations and local areas through contribution, curios, and instruments recommends a rising dependence on other FOSS drives. There is a need to help and defend non-benefit establishments that have arisen because of colossal, multi-part FOSS projects requiring their scholarly privileges to be safeguarded.

The FOSS project local area and its collusions will in this way begin to crumble on the off chance that such an organization begins to go to pieces or on the other hand in the event that the new guidelines can't be acknowledged and fulfilled.

Fourthly, as has previously been referenced, the socio-specialized trap of FOSS incorporates both private and public assets like individuals' time, exertion, consideration, ability, feeling (convictions and values), and registering assets.

Intelligent practice, industry reviews, overview research, ethnographic examinations, moderately ongoing techniques for mining programming vaults[2], multi-modular demonstrating and examination of the socio-specialized cycles and organizations found in supported FOSSD projects, and other ordinary strategies for examination show that the experimental investigation of FOSS is growing.

At long last, experimental exploration of FOSSD is widening the meaning of what we can see, find, assess, or find out about how enormous Programming frameworks can be made or as of now exist.

Source code, relics, and online vaults for FOSS projects address and give new, unreservedly open information sources that are, lavishness and intricacy in a way that has until recently never been accessible for examination overall premise Programming process demonstrating, for example, and conventional reproduction exploration and application has relied upon examinations and approvals are upheld exactly by processes from this present reality.

In any case, such data has much of the time been difficult to find, costly to get, and regularly not shared or dependent upon free reanalysis because of elements consolidating non-revelation or secrecy arrangements. FOSS process information and venture antiquities are put away in project ancient rarity archives, product curios that can be accumulated, analyzed, shared, and reevaluated uninhibitedly and transparently.

2.3 User-centered design of power tools

Since numerous provider markets, for example, the market for power devices, have changed to a purchaser market, client focused plan (UCD) is turning out to be increasingly more essential for item originators and makers [3]. Accordingly, the nature of the products created by the different makers slowly adjusts.

The emotional assessment of the items assumes a greater part in the buying decision when the specialized usefulness and objective nature of the items are equivalent high. UCD will probably make client cooperations with innovative frameworks as powerful and proficient as conceivable while likewise expanding client fulfillment. UCD enjoys many benefits for human-machine frameworks where there is a great deal of direct actual contact between the client and the item.

While involving such items as power devices, the client is in the progression of information and power with the specialized item. The objective of the originator of such items is to work on an item's apparent convenience. Different ease of use includes that rely upon various elements, including human variables, item credits, and natural variables, which affect how convenience is seen. Finding and carrying out convenience factors during the advancement cycle is oftentimes troublesome, costly, and tedious.

The development of client-focused frameworks that unequivocally connect with the client, similar to drive instruments, is a troublesome undertaking. Because of the incredibly adaptable and mind-boggling utilization of force instruments, various convenience factors should be considered during framework advancement to amplify apparent ease of use.

The nonexclusive USE-Cycle, which means eliminating time-and cash-concentrated process steps in the client-focused improvement of force devices, is given in this work. By iteratively adjusting and consolidating a few power devices, the introduced strategy constructs a general and coordinated approach by using previously existing and profiled approaches. In the main period of the Utilization Cycle, a field study is directed to distinguish convenience factors that mean quite a bit to the apparent ease of use.

Steps are introduced for setting up a testing climate that empowers a reproducible and objective assessment of seen convenience, giving the valuable chance to improve the advancement of force devices in light of the applications saw in the field review. The results of the emotional evaluation are at long last evaluated utilizing a picked estimation technique.

The field study's organized procedure ensures that all fundamental ease-of-use factors are found and considered while fostering a power device's UCD.

The efficient strategy for assessing apparent convenience emotionally and the evaluation of the discoveries using estimation methods make the assessment of seen ease of use more goal and empower a lessening in the necessary subject populace, which further decreases work.

Due to its conventional structure, assessing apparent ease of use is less difficult and more compelling in light of the fact that it requires less work to adjust to a specific item.

The item originator can more readily get a handle on the connections between the results of individual ease-of-use components and the item credits by measuring the emotional assessment utilizing estimating innovation.[4]

Past the area of innovative advancement methods, open-source programming has made progress. As a general rule, "open source" has been utilized to depict a wide assortment of things, including yoga and shirt plans.

At the point when these non-mechanical instances of open source are introduced, it is suggested that many individuals can take part in the development of these particular imaginative endeavors. Despite the fact that it is obvious that the term's subsidiary implications have their foundations in the product business, it is imperative to remember that product improvement is a particular type of improvement and may not give the most regular correlation with different kinds of creation.

2.4 Role Of Truecaller Application In Preventing Phone Call And Text Message Scams

New specialized improvements continue to show up. There are other latest renditions accessible when we are as yet utilizing something just distributed in a particular variant, such as cellphones, TV, and the applications we utilize consistently on our gadgets. Nonetheless, it can't be rejected that human interest in innovation, which they expect to meet their day-to-day needs, drives innovation's always fast turn of events.

It appears to be that individuals and their gadgets are presently indivisible because of how profoundly imbued they are in friendly collaborations. As is widely known, an ever-increasing number of individuals are utilizing cell phones. Most gatherings or classes as of now have cell phones in their control.

These days, individuals, everything being equal, from the extremely youthful to the exceptionally old, are talented at dealing with cell phones and other

touch-delicate rectangular gadgets. This incorporates representatives, self-employed entities, road sellers, understudies, and housewives.[5]

Today, various enormous companies make a scope of electronic gadgets that can show an assortment of information media, virtual entertainment, and diversion that definitely stand out. These gadgets likewise incorporate applications with a scope of cutting-edge capacities.

CHAPTER 3: SYSTEM DESIGN & DEVELOPMENT

3.1 Technologies Used

An Internet hosting service for software development and version control using Git, known as Github. It provides the distributed version control of Git plus access control, bug tracking, task management, software feature requests, continuous integration, and wikis for every project.[6]

Tool I -

- **Technologies used -**

  Robocopy: Windows has had a command-line tool called robocopy. It can be seen as an alternative for the less functional XCopy tool. At the command prompt, you can specify a drive path or server path to copy/move files.

- **Platform used -**

  VSCode: Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

  Command line interface(cmd): It is the default command-line interpreter for the OS/2, eComStation, ArcaOS, Microsoft Windows, and ReactOS operating systems.

- **Language used -**

  Python: Python is an object-oriented, interpreted, and interactive programming language. It is also used for parallel computing systems

and has a comparatively simple and easy syntax for coding and still, it is a powerful programming language.[7]

- **Libraries used -**

  Tkinter: Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions.

  Subprocessor: The subprocess module lets you spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

Tool II -

- **Technologies used -**

  Gradle: Gradle is a build automation tool for multilingual software development. The tasks of compilation and packaging, as well as testing, deployment, and publication, are all under its control. The languages that are supported are Java, C/C++, and JavaScript.

- **Platforms used -**

  Android Studio: The official integrated development environment (IDE) for Google's Android platform, Android Studio, is based on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is the primary IDE for building native Android applications, and can be downloaded for use on Windows, macOS, and Linux-based operating systems.

- **Languages used -**

  Java: Java is a high-level, class-based, object-oriented programming language. Because it is a general-purpose programming language,

compiled Java code can run on any platform that accepts Java without the need to recompile.

XML: Extensible Markup Language (XML) is a markup language and file format for storing, sending, and recreating arbitrary data. It outlines a set of guidelines for document encoding in a way that is both machine- and human-readable.

- **Database used -**

  The database provides phone number info using a third-party crowdsourced phone number database from some other proprietary app.

Tool III -

- **Technologies used -**

  Pandoc : Pandoc is a command line tool that is a free-software document converter, widely used as a writing tool and as a basis for publishing workflows. Pandoc can convert between a variety of markup and word processing formats, such as Word docx, HTML, LaTeX, and many varieties of Markdown.

  LaTeX : LaTeX is a top-notch typesetting programme that has tools for producing technical and scientific documentation.LaTeX is available as free software.

- **Platform used -**

  VSCode: Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

- **Language used -**

  Python: Python is an object-oriented, interpreted, and interactive programming language. It is also used for parallel computing systems and has a comparatively simple and easy syntax for coding and still, it is a powerful programming language.[7]

- **Libraries used - tkinter, python-docx, PyPDF2**

  Tkinter: Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions.

  Python-docx: The python-docx module allows you to create new documents as well as make changes to existing Microsoft Word(.docx) files.

  PyPDF2: The PyPDF2 module is a Python library used to carry out operations on PDF files, such as extracting document-specific data,, dividing PDF files into pages, merging PDF files etc. Since it is a pure Python library, it has no platform-specific dependencies on any external libraries, hence can be used on any platform.

Tool IV -

- **Technologies used -**

  Xampp: Xampp is a free and open-source stack of cross-platform web server components created by Apache Friends. It primarily includes the Apache HTTP Server, database, and interpreters for PHP and Perl scripts. The components used by Xampp are the same as those used by most real-world web server developments, hence it makes it easier to transition from a local server to a live server.

- **Platforms used -**

  VSCode: Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

- **Languages used -**

  PHP: PHP is a general-purpose programming language designed specifically for web development. PHP code is usually processed on a web server by an interpreter implemented as a module or as a CGI executable.

  HTML: HTML (HyperText Markup Language) is the standard language for documents designed to display in a web browser. HTML gives the structure of a web page semantically. It is often used with technologies such as CSS and scripting languages like JavaScript. A web browser receives HTML documents from local storage or a web browser. HTML describes the structure of a web page semantically.

  CSS: CSS (Cascading Style Sheets) is a style sheet language that is used for describing the look of a document written in a markup language such as XML or HTML. It is designed to allow the separation of content and presentation, including the layout, colors, and fonts of the web pages.

  MySQL: MySQL is a relational database management system (RDBMS) that is open-source. An RDBMS, such as MySQL, works with an operating system to implement a relational database in a computer's storage system, manages users, permits network access, and makes it easier to evaluate database integrity and create backups.

3.2 Use case diagram

Tool I

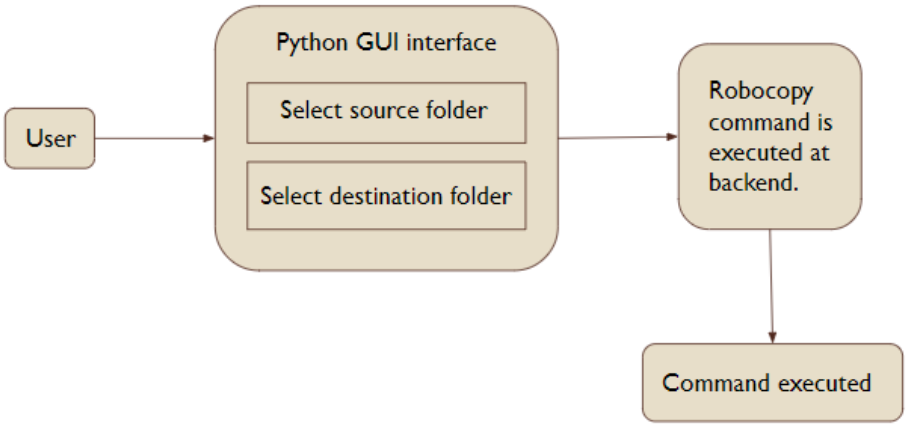


Figure 1: Represents the flow chart for Tool I.

Tool II

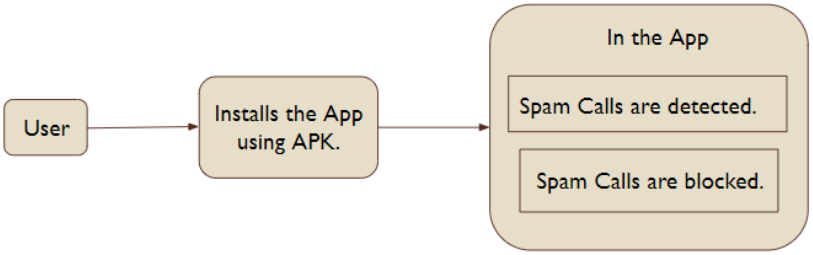


Figure 2: Represents the use case flow chart for Tool II.

Tool III

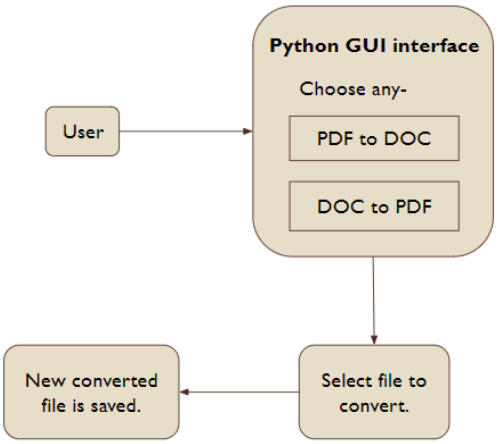


Figure 3: Represents the use case flow chart for Tool III.

Tool IV

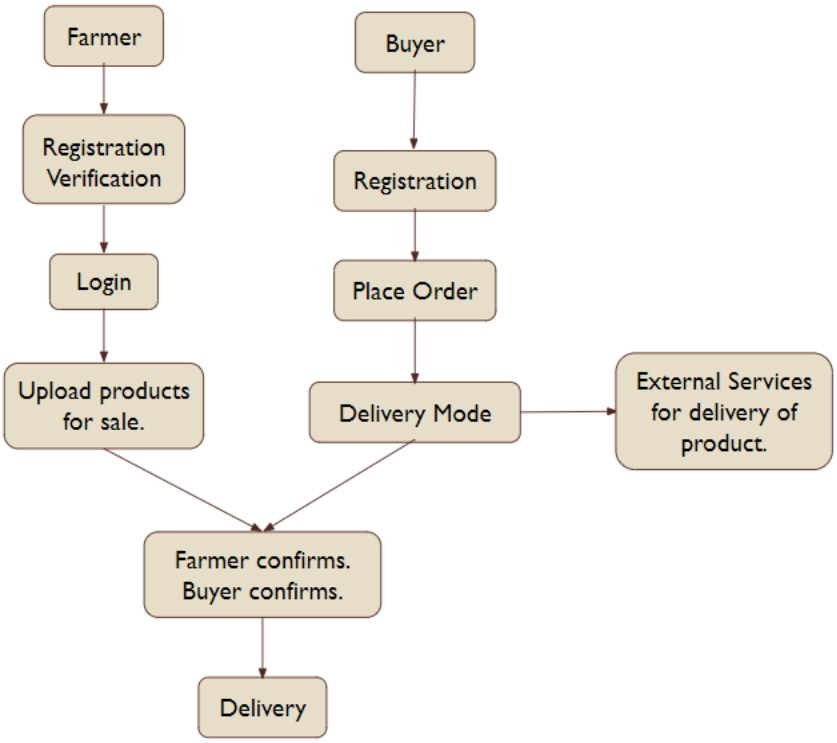


Figure 4: Represents the use case flow chart for Tool IV.

3.3 Implementation

A repository containing user-centric tools using FOSS(Free Open Source Software). Software packages are kept in a repository, also known as a repo. Along with metadata, a table of contents is frequently also kept.

Usually, source control or repository managers are in charge of running a software repository. Repositories can be installed and updated automatically using package managers.

For now, the repository will be hosted on GitHub. GitHub is an open source platform used to create repositories for source code of tools and technologies which can be used/altered by other developers.

Tool I

Robocopy stands for “Robust File Copy”, it is a command-line directory and/or file replication command for Microsoft Windows.

Robocopy is one of the most-used command-line utilities to copy large volumes of data. It is very powerful too, which can get complex at times. It is useful for performing large jobs or daily backups that do not require live interaction of the user but take time to execute completely.

Table 1. Comparison between XCopy and Robocopy.

XCopy	Robocopy
Directors can be copied using	Robocopy can specify beforehand what to do if there are conflicting

XCopy.	file names.
XCopy can mimic the exact structure of the source directory and copy all files, including subdirectories, recursively.	With Robocopy the user can exclude some files or folders as per their requirement.
Files can be excluded from Xcopy depending on their name or extension.	Robocopy can schedule copy-paste operations beforehand.
XCopy does not have the option of mirroring, like Robocopy.	Robocopy can be used to mirror or synchronize directories
XCopy does not support an option where commands can be scheduled to take place at a certain time.	By using the /RH ("Run-hours") option in Robocopy, commands can be pre-scheduled.
Monitoring of files is not supported in XCopy.	The /MON or /MOT commands in Robocopy provide file and directory monitoring.

Instead of copying all files from one directory to another, Robocopy will be able to examine the destination directory and remove files that are no longer part of the main tree.

Additionally, it will compare the files in the target directory with the ones that need to be copied, saving time by avoiding duplicating files that have not changed.

Table 2. Few Robocopy commands can be implemented.

S.R. No.	Command	Function
1.	/S	Copy all subfolders from the source folder to the destination folder
2.	/E	Copy all subfolders, including empty subfolders, from the source folder to the destination folder.
3.	/COPY:[DATSOU]	Provides various Copy options. Default is /COPY:DAT. (D=Data, A=Attributes, T=Timestamps, S=Security, O=Owner info, U=aUditing info.)
4.	/SEC	Copy files with security.
5.	/COPYALL	Copy all file information from source to destination.
6.	/NOCOPY	Copy no file information.
7.	/NODCOPY	Copy no directory information.
8.	/A	Copy files only with the Archive attribute from the source folder to the destination folder.
9.	/M	Similar to /A, but remove the Archive attribute from the source files.
10.	/FFT	Assuming FAT File Times.
11.	/CREATE	Create a directory tree and zero-length files only

12.	/L	List only files. The file timestamp is not copied, and no file is deleted.
13.	/MOV	Move files(delete files from source after copying).
14.	/MOVE	Move files and directories(delete files from source after copying).
15.	/B	Copy files in backup mode.
16.	/Z	Files are copied in restartable mode. These files can survive a network glitch.
17.	/J	Copy using unbuffered I/O (Can survive a network glitch.)
18.	/MIR	Ensuring all files in the destination folder are in the source folder, and deleting the destination folder.
19.	/XO	Exclude older files if the destination file exists and is the same date or newer date than the source(No overwriting of files).
20.	/PURGE	Delete the destination file/folder that no longer exists in the source.

The Python Tkinter library is used to create a simple user-friendly GUI. The user selects the source directory, the destination library, and the Robocopy command to be executed. Figure.1 shows the GUI developed using Python Tkinter and the subprocessor library.

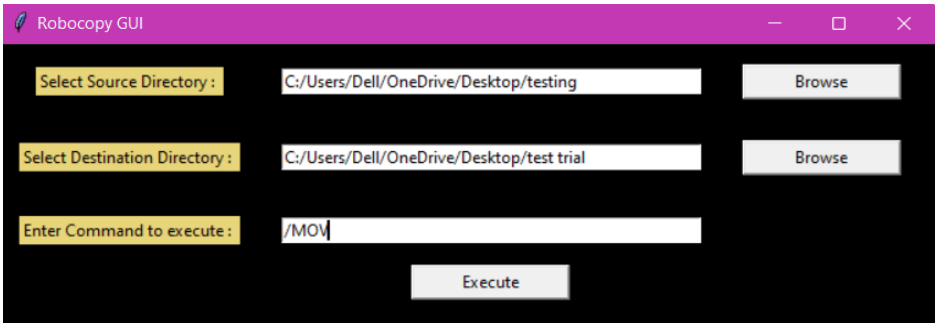


Figure 5: GUI interface to perform Robocopy. /MOV command is used to move the files from the source folder to the destination folder.

At the backend of the user interface, Robocopy is executed in the terminal via the command-line interface to perform the required operation. Figure.2 represents the command-line execution of the Robocopy command in the python terminal.

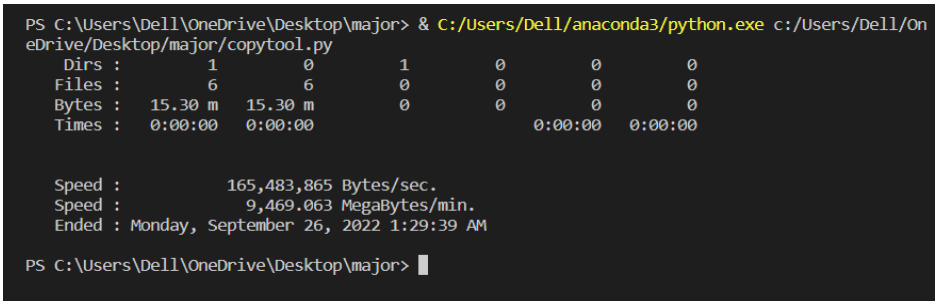


Figure 6: Robocopy command is executed on the backend on the command line interface.

A confirmation prompt appears after the successful execution of the entered command. Figure.4 shows the prompt which appears.

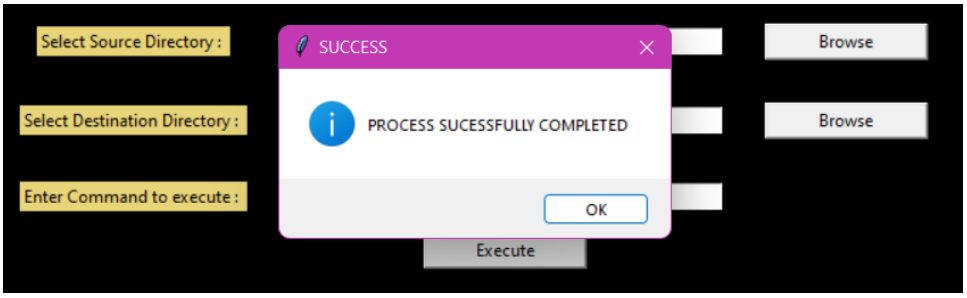


Figure 7: Confirmation of execution of the process.

The screenshot shows the comparison of the source folder and the destination folder.

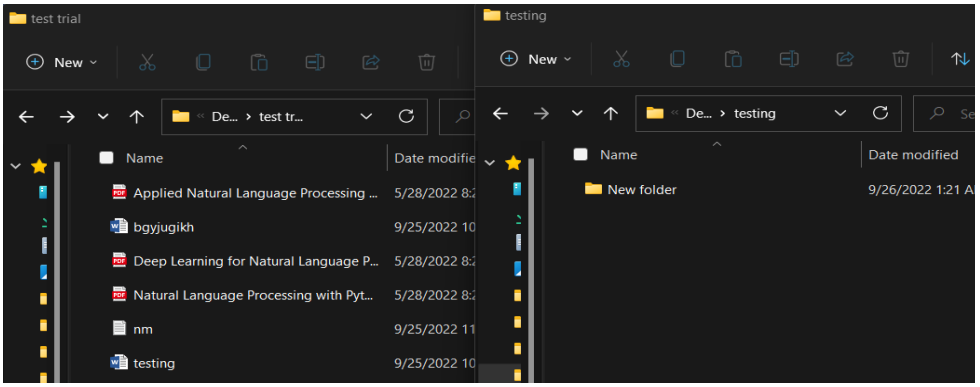


Figure 8: Files moved successfully from the source to the destination folder.

Tool II

The offline database provides the following information -

1. Different Phone numbers.
2. Category of the number. For example, if the number is from telemarketers, debt collectors, or some other scam call.

After Installing the APK file, when the application is opened the app asks for permission to access phone logs, and to manage phone calls, as shown in Figure 7 and Figure 8.

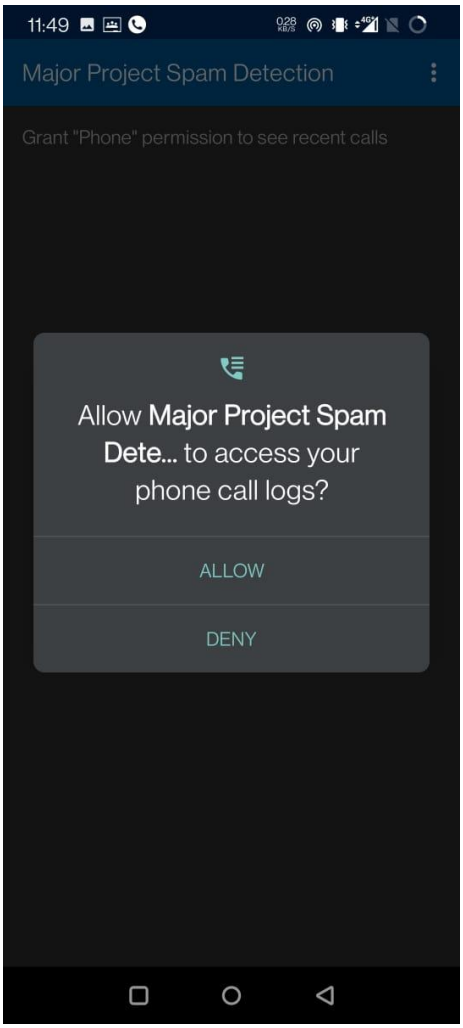


Figure 9: Prompt asking for access to Phone call logs.

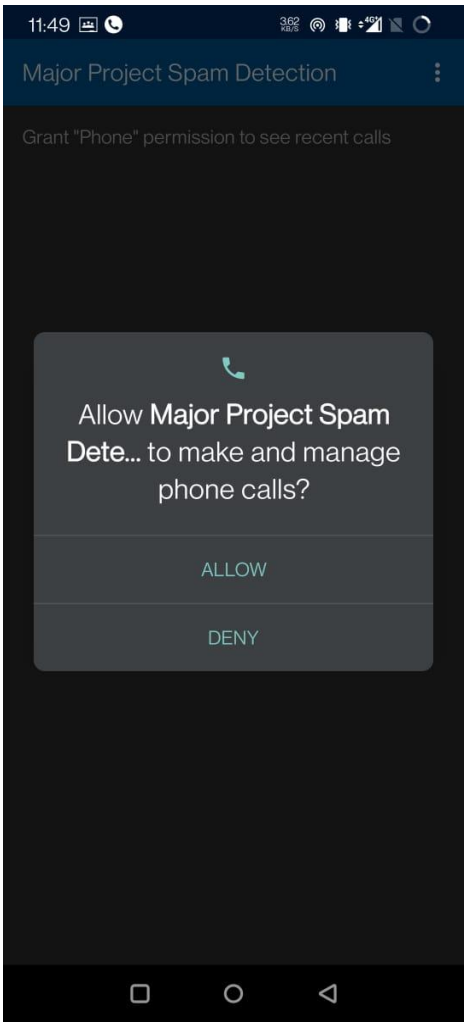


Figure 10: Prompt asking to manage and make phone calls.

In Figure 9, the snippet of the about page of the app shows the details about the application and the developers. After giving the required permissions to

the application the app will then detect phone numbers that are spam and let the user know as shown in Figure 10.

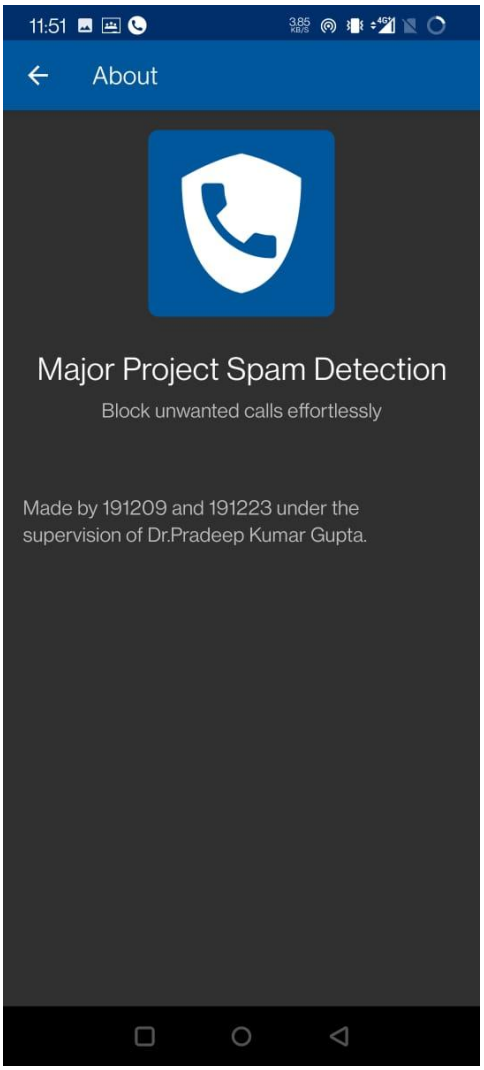


Figure 11: The about page of the Application.

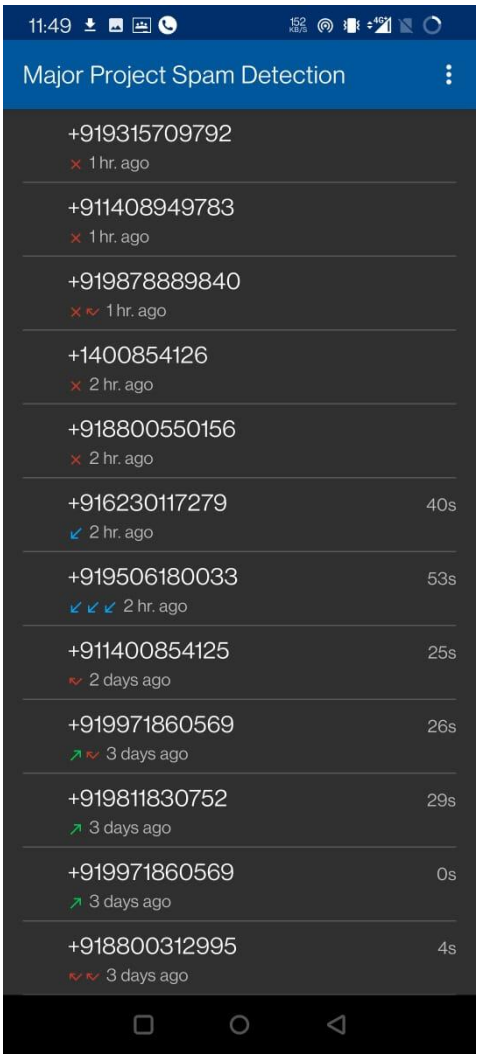


Figure 12: Interface of the Application after the permissions are provided.

The Application is now running and will alert the user about any spam calls, without the user having to pick up the calls. The user will get knowledge of the spam calls beforehand.

Furthermore, the user can use a BlackList, where they can add the phone numbers of recurrent spammers and block the list. This way they will not receive any more calls from the scam number.

Tool III

Each character, line, and image in a PDF document is placed using absolute coordinates on the page, utilising a set document structure. Each page stands alone entirely.

In contrast, Doc(x) is a flowable document format that has no idea of pages or coordinates at all. Automatic content flow fills the available space or follows various shapes in paragraphs and tables. Positioning is relative to the margins or the preceding paragraph rather than absolute.

In other words, only text, curves, and images are known to PDF. Paragraphs, tables, lists, relative spacing, and shapes are all covered by Docx.

Pandoc converts a source file to a PDF file by using the pdflatex, xelatex, or another TeX command. By default, Pandoc creates PDF documents using LaTeX. Therefore, you must first install a LaTex processor if you want to create PDF documents.

Upon running the python script, the GUI interface opens up, as shown in Fig.13 .

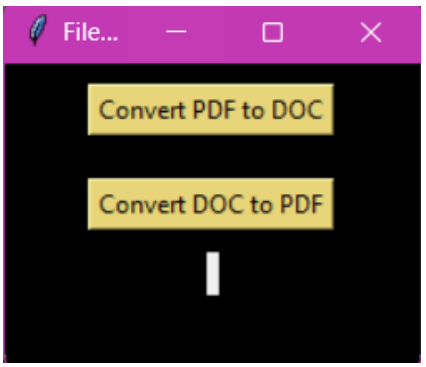


Figure 13: Interface of File convertor.

The user can then select if they want to convert to PDF or DOC. The Select file Window will open for the user to select the file to be converted.

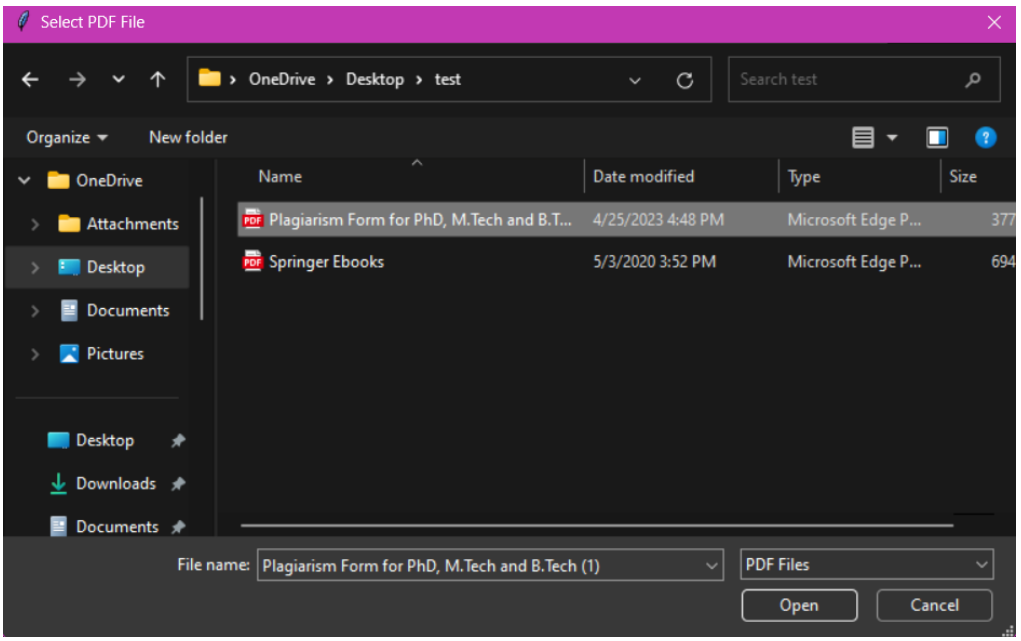


Figure 14: Selecting PDF file to convert to DOC.

Once the file is selected, the converted .docx file will be saved in the same folder with the same name as the original .pdf file. A completion message will be displayed in the GUI window, as shown below in Fig.15 .

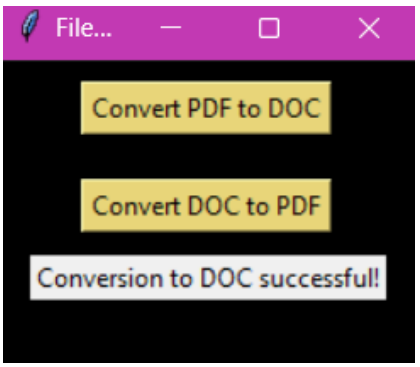


Figure 15: Confirmation message after successful conversion to DOC.

When converting a DOC file to a PDF file, the Select file window will open. The user then selects the file to be converted.

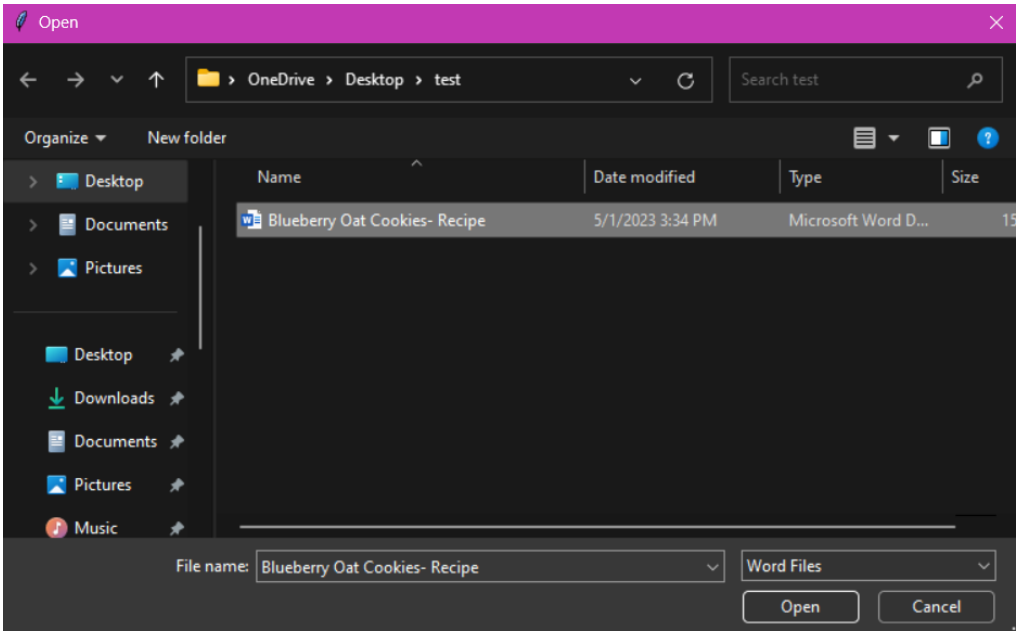


Figure 16: Selecting DOC file to convert to PDF.

After selecting the file to convert, the user then has to decide the name of the converted file and its location, as shown in Fig 17.

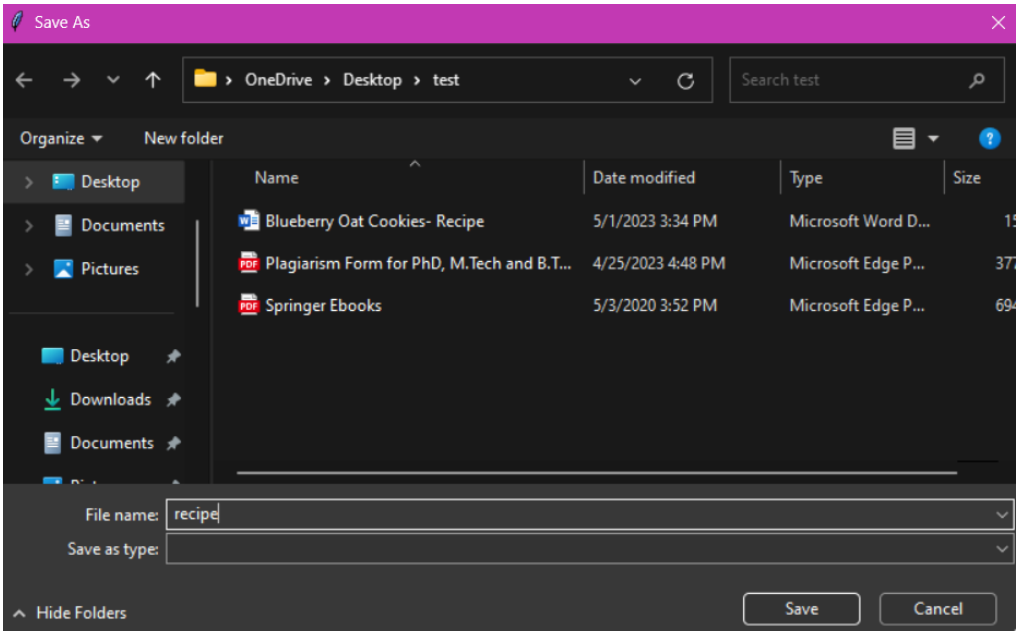


Figure 17: Setting name for the DOC file to be saved.

Followed by this, a completion message will be displayed in the GUI window, as shown below in Fig.18.

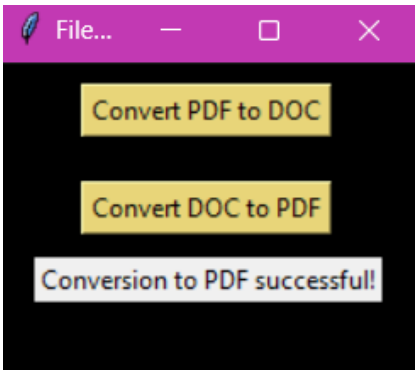


Figure 18: Confirmation message after successful conversion to PDF.

Tool IV

A website for farmers to sell their products directly online, without the involvement of a third party/middleman. The buyers can directly search by area to find the nearest farmer selling the products they need.

The front end of the website uses HTML and CSS. At the backend, there is PHP, and MySQL to store the data of the farmers and customers.

The Homepage of the website is shown in Fig 19.

Figure 19: Homepage of website.

A farmer can go to the option of Farmer to login as a farmer. The login page for a farmer is shown in Fig 20.

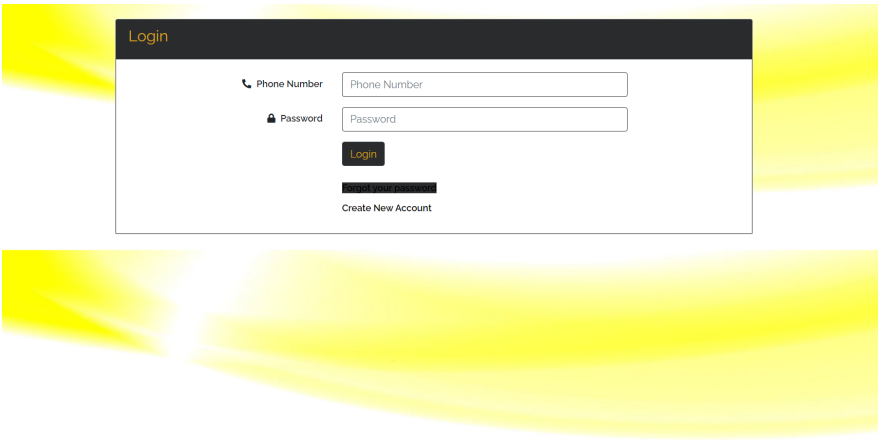


Figure 20: Farmer Login Page.

After Logging in, the Farmer Homepage opens up, shown in Fig 21.

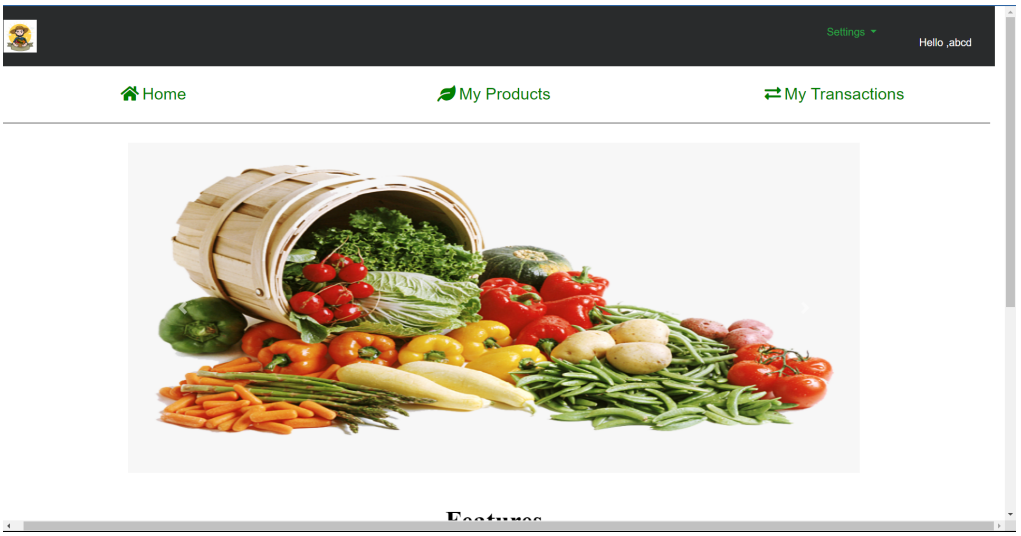


Figure 21: Farmer Homepage.

To upload their products, the farmer can visit the My Products section in the center of the page.

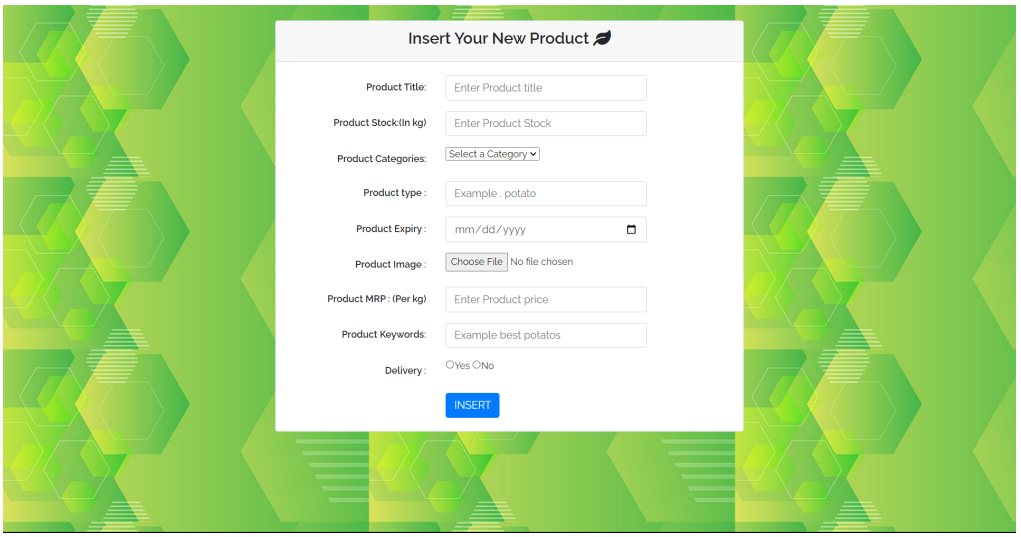


Figure 22: Adding product page.

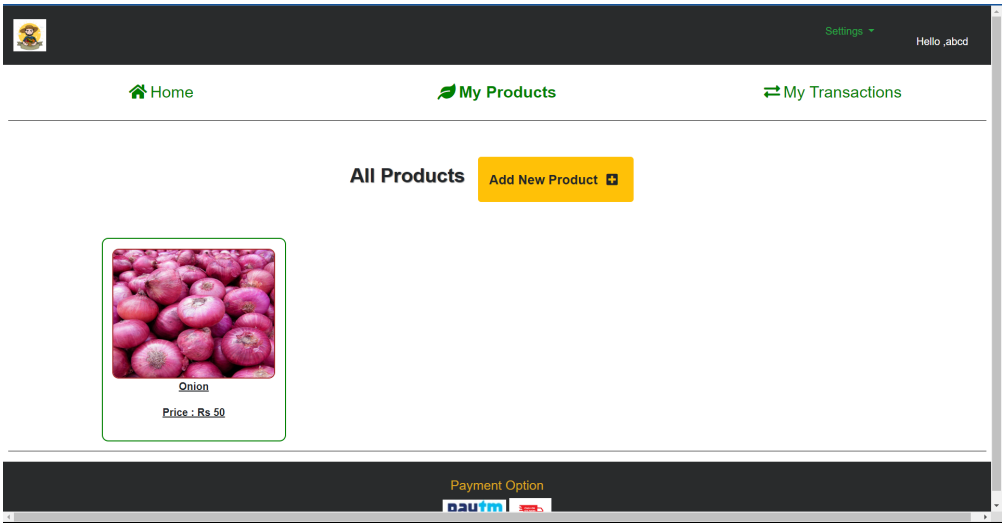


Figure 23: Farmer products section showing added products.

Buyer Login

Similarly, the buyer login page is shown in Fig 24. The buyer can login and access their homepage.

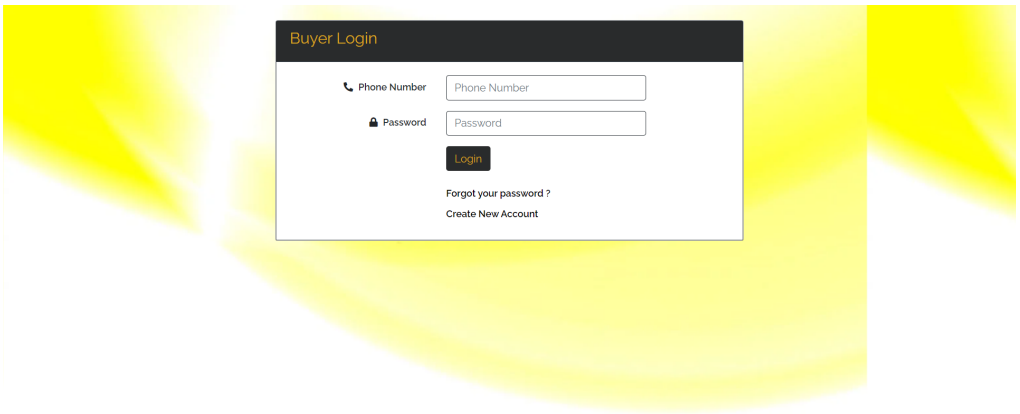


Figure 24: Buyer Login Page.

The buyer homepage is shown in Fig 25. The buyer can select the products they want to buy and add them to their cart.

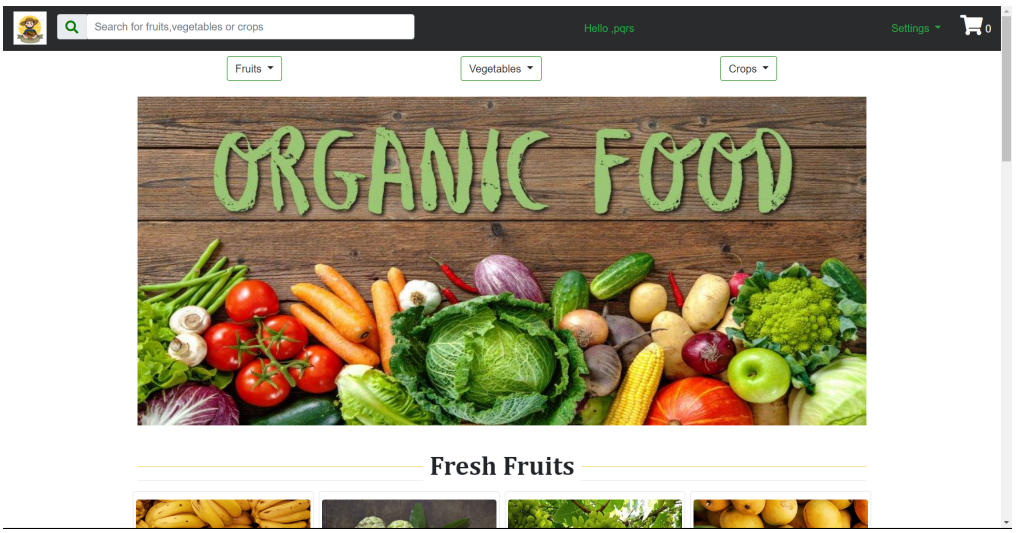


Figure 25: Buyer Homepage.

Creating a Repository

Various platforms are available on the internet to create repositories for the open-source use of those files. I decided to use GitHub to create a repository of user-friendly tools.

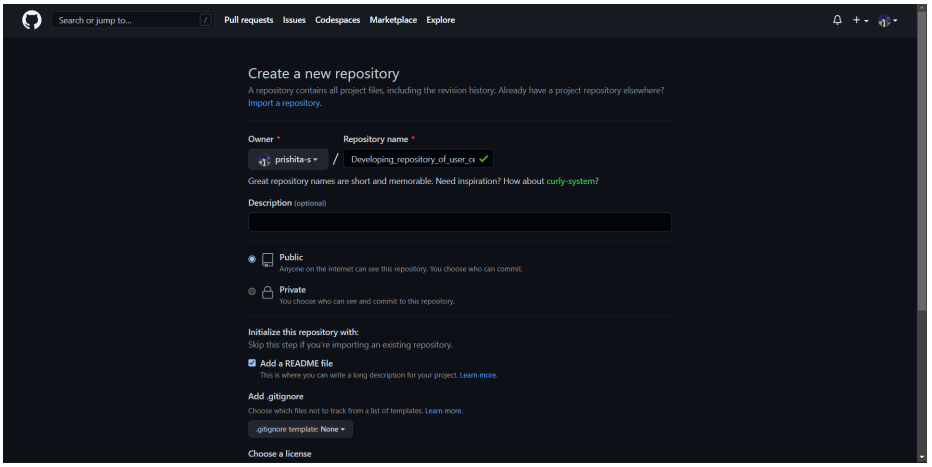


Figure 26: Create a repository on GitHub.

In the repository, a folder for each of the tools is created, and the tools are added to their respective folders.

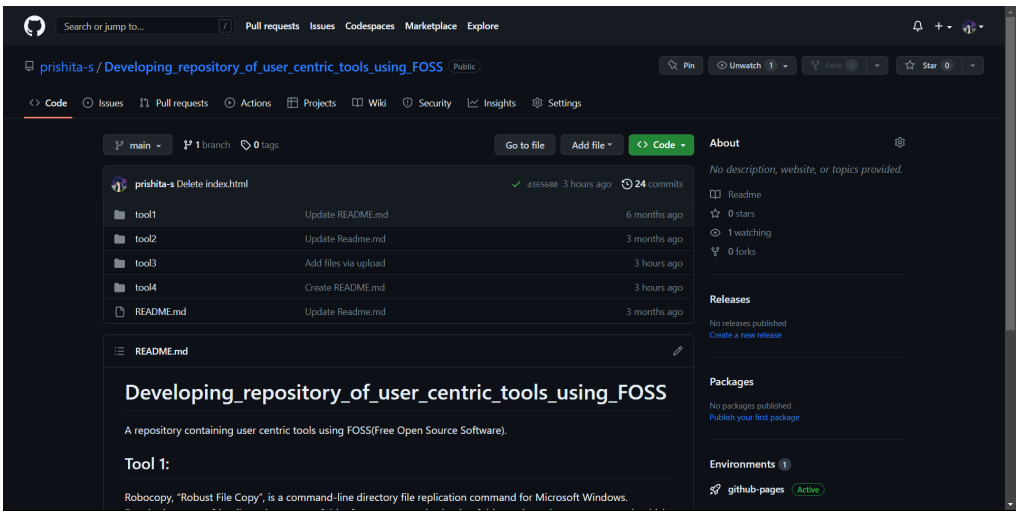


Figure 27: Snippet of the repository created.

Users across the internet can download a copy of the tools from the repository to use on their personal or work computers.

CHAPTER 4: EXPERIMENTS & RESULT ANALYSIS

It is inevitable for mistakes or things that go wrong to occur when creating new applications or even making improvements to ones that already exist. Performance may be adversely affected right away or gradually.

The performance of the tools can be analyzed by observing them and implementing different functions according to the requirements.

Tool I

Before copying the source and destination folders are shown in Figure 11.

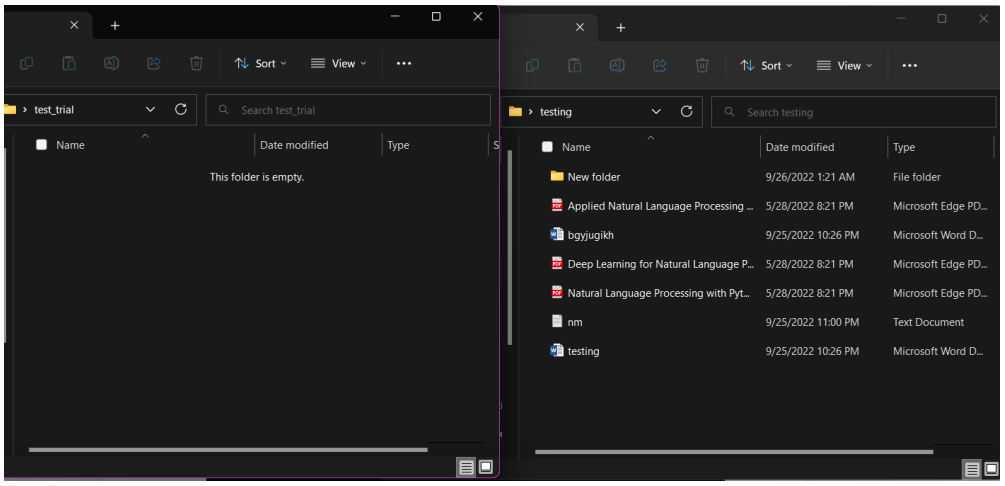


Figure 28: Source and destination folders before executing copy commands.

First, we copy the files using XCopy to see the time it takes to copy the files from the source folder to the destination folder.

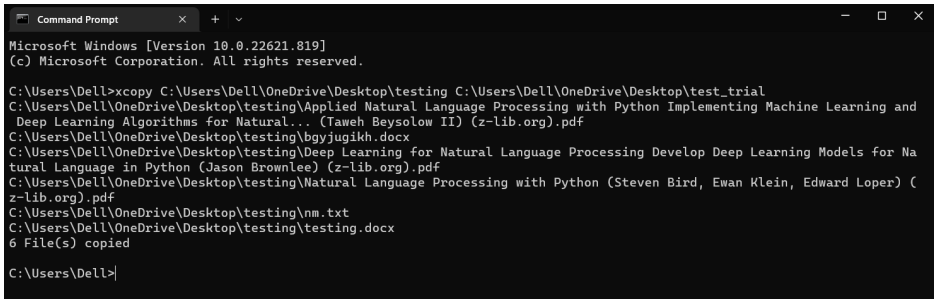


Figure 29: Copying using XCopy.

We then copy the same files using the Robocopy GUI created.

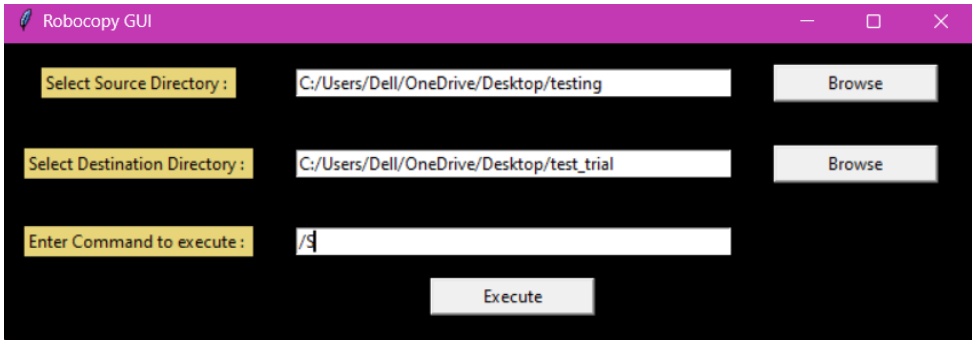


Figure 30: Copying using Robocopy.

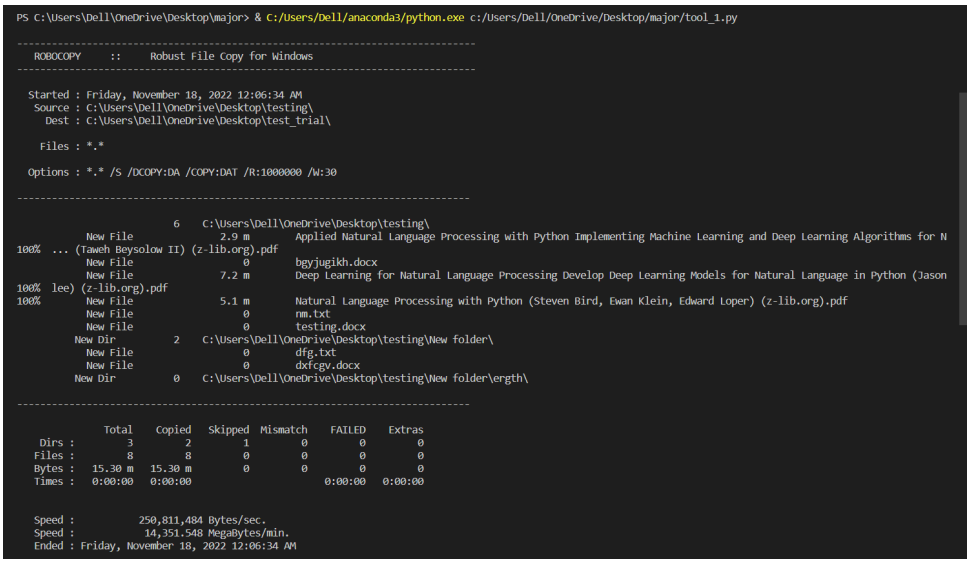


Figure 31: Backend of Robocopy command executed in GUI.

After executing the copy commands of both Xcopy and Robocopy GUI, the source and destination folders are shown in Figure17 .

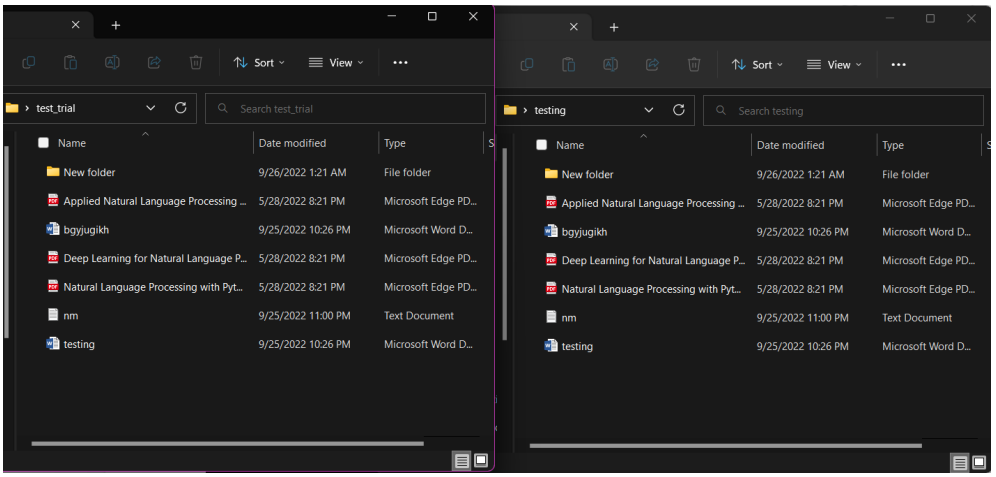


Figure 32: After executing copy commands.

The figures below show another command being executed using the Robocopy GUI tool.

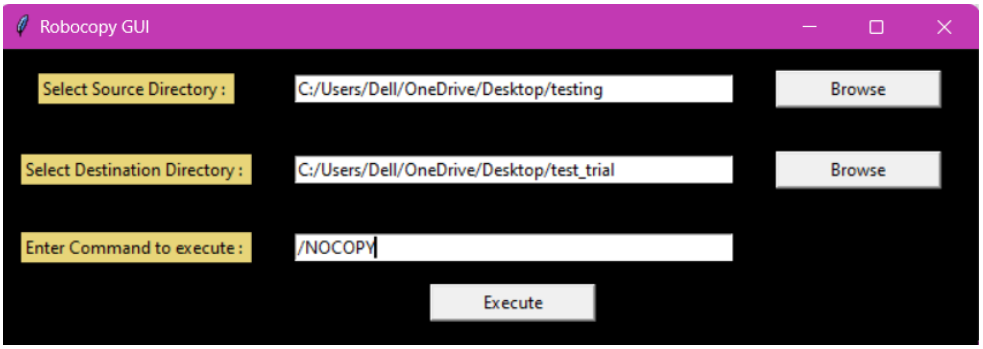


Figure 33: Another command executed in GUI.

```
------------------------------------------------------------------------------
   ROBOCOPY     ::     Robust File Copy for Windows
------------------------------------------------------------------------------

  Started : Friday, November 18, 2022 12:09:20 AM
   Source : C:\Users\Dell\OneDrive\Desktop\testing\
     Dest : C:\Users\Dell\OneDrive\Desktop\test_trial\

    Files : *.*

  Options : *.* /NOCOPY /R:1000000 /W:30

------------------------------------------------------------------------------

                           6    C:\Users\Dell\OneDrive\Desktop\testing\

               Total    Copied   Skipped  Mismatch    FAILED    Extras
    Dirs :         1         0         1         0         0         0
   Files :         6         0         6         0         0         0
   Bytes :   15.30 m         0   15.30 m         0         0         0
   Times :   0:00:00   0:00:00                       0:00:00   0:00:00
   Ended : Friday, November 18, 2022 12:09:37 AM
```

Figure 34: Backend of the executed command.

Tool II

Local Blocking of calls using a Blacklist can be done to prevent spam calls from particular phone numbers.

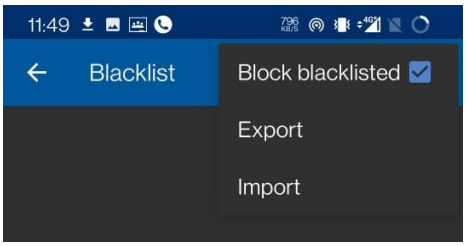


Figure 35: Blocking calls from the Blacklist.

Phone numbers can be added to the Blacklist with a name so that the user knows which number they have blacklisted. These Blacklisted numbers can now be prevented from calling the user. Figures show how a Blacklist is created and saved in the application.

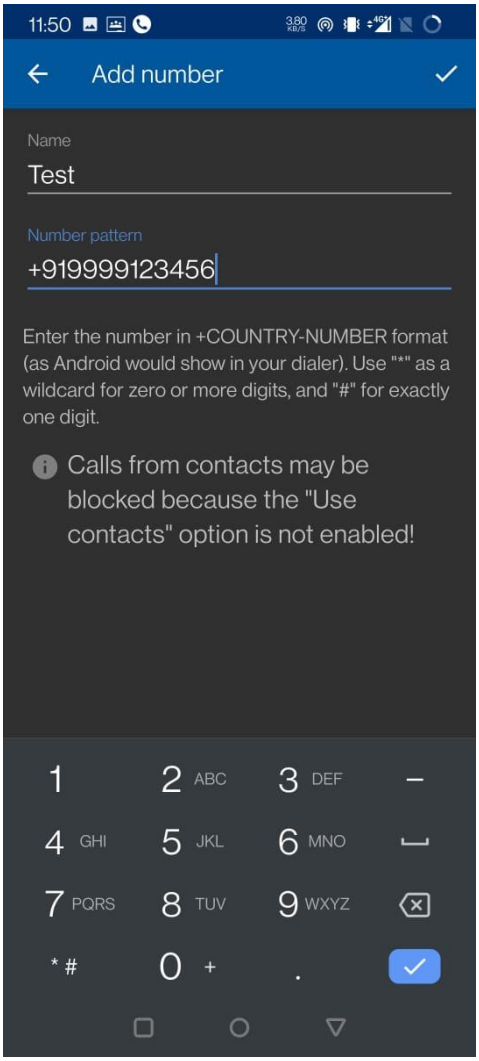


Figure 36: Adding a number to the Blacklist

Figure 37: Shows the Blacklist.

When a number in the blacklist calls the user, the call by the caller will automatically be cut. In this way, the receiver will get a notification about the call from the number, but won't have to pick up the call. The call will already be cut beforehand.

Tool III

After Converting the doc file to PDF format, it is saved in the destination folder with the name recipe.pdf.

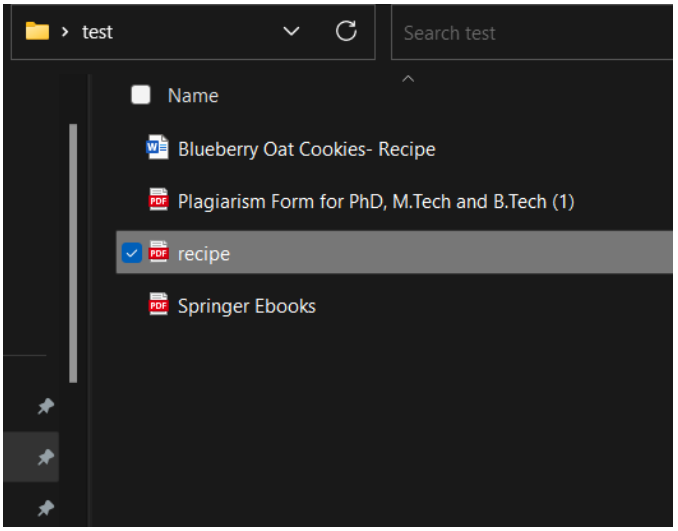


Figure 38: The converted PDF file named recipe.pdf is saved in the same folder as the original .docx file.

The converted doc file is saved in the same folder as the original .pdf file, shown in Fig 39.

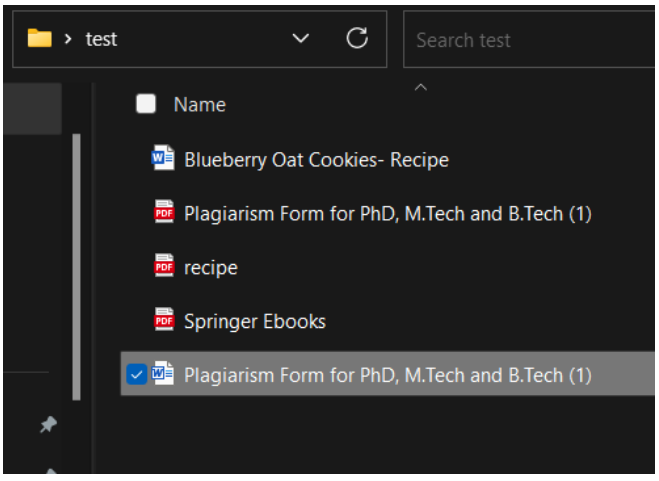


Figure 39: The converted DOC file is saved in the same folder as the original PDF file.

Tool IV

Websites for farmers selling products directly to buyers are very few over the internet. Farmers can directly log in with their mobile number and add products to sell directly to their customers.

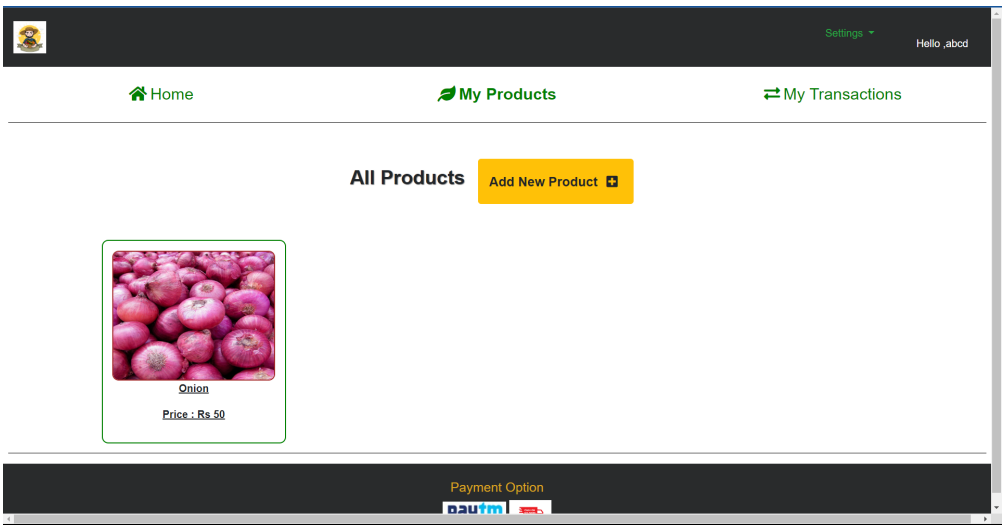


Figure 40: My products page where farmers can add their products to sell.

Customers can directly search by category fruit/vegetable/crop, select their city and buy the product from the nearest available farmer.

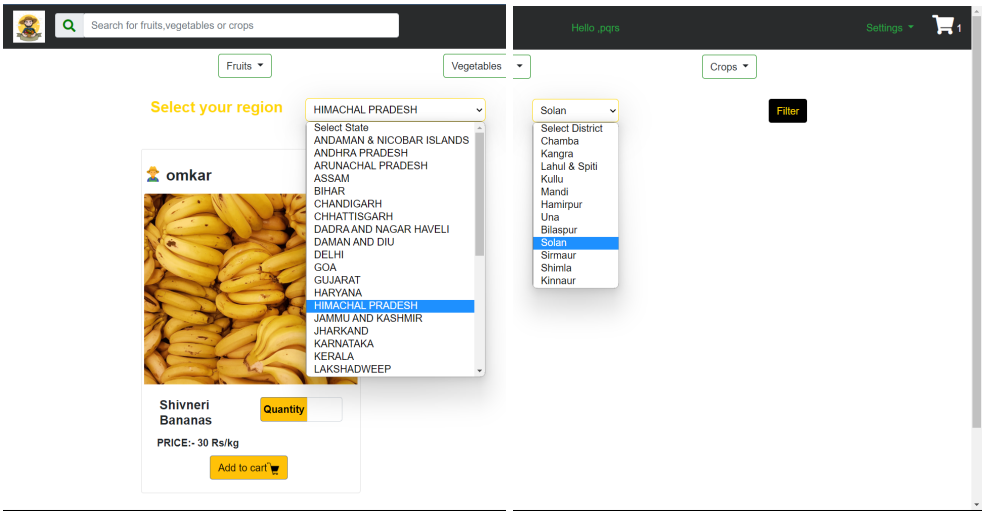


Figure 41: Buyer can filter by region.

After adding products to their cart, the buyer can then proceed to the checkout and select their delivery option. There are two delivery options, one is where the farmer delivers the product to the buyer, and the other is where the buyer pickups the product from the farmer.

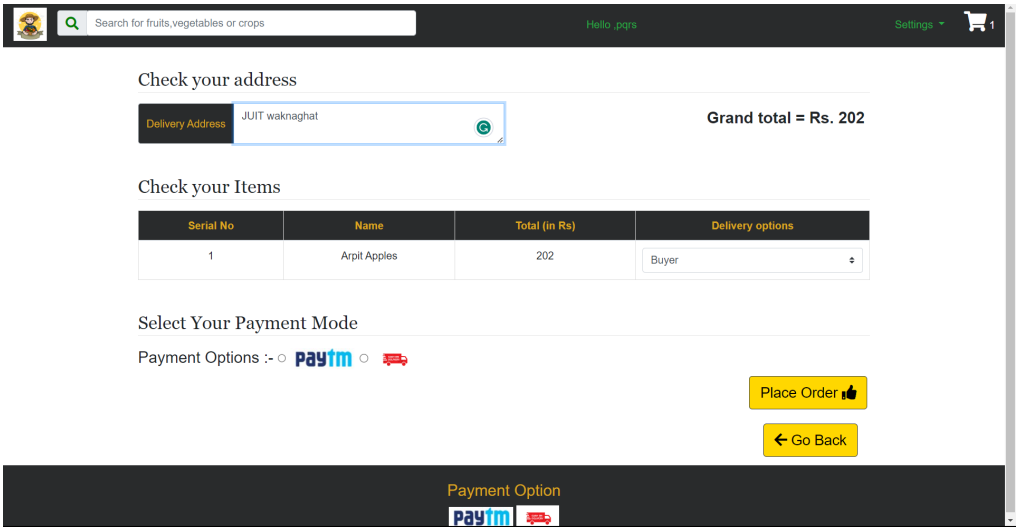


Figure 42: Checkout page.

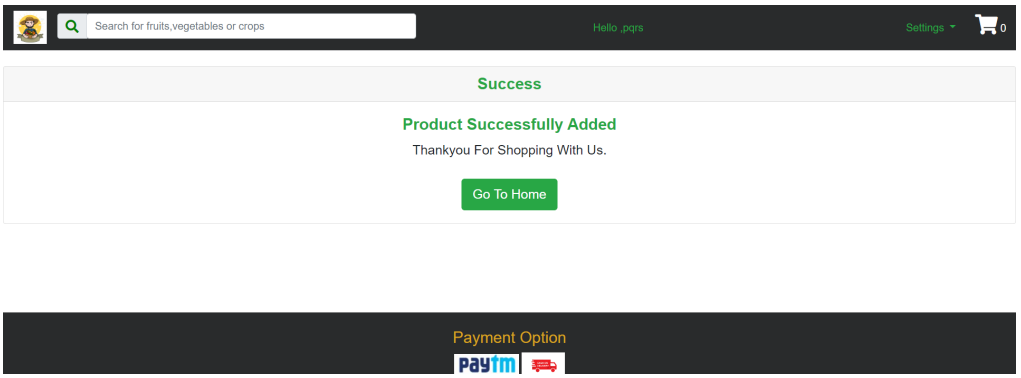


Figure 43: Transaction complete.

CHAPTER 5: CONCLUSION

5.1 Conclusions

Open source software is widely used and continues to grow. Every aspect of our digital environment, from PCs to mission-critical programmes, is supported by open source.

Due to the rapid pace of development and data processing, open source will need to be the reactive method of development. The problem of data overload will make it even more necessary to have quick access to open source software.

5.2 Future Scope

The created repository can be further expanded by developing more user-centric tools and adding them to the repository.

A similar repository can be created on the University server for easier access to university individuals that want to perform various tasks possible by the developed tools of the repository.

The scope of the individual Tools can be seen as

- **Tool I**

  When the files/folders are copied using the Robocopy command in the command-line interface, one can use multiple commands for a particular set of files/folders, as shown in Figure.8.

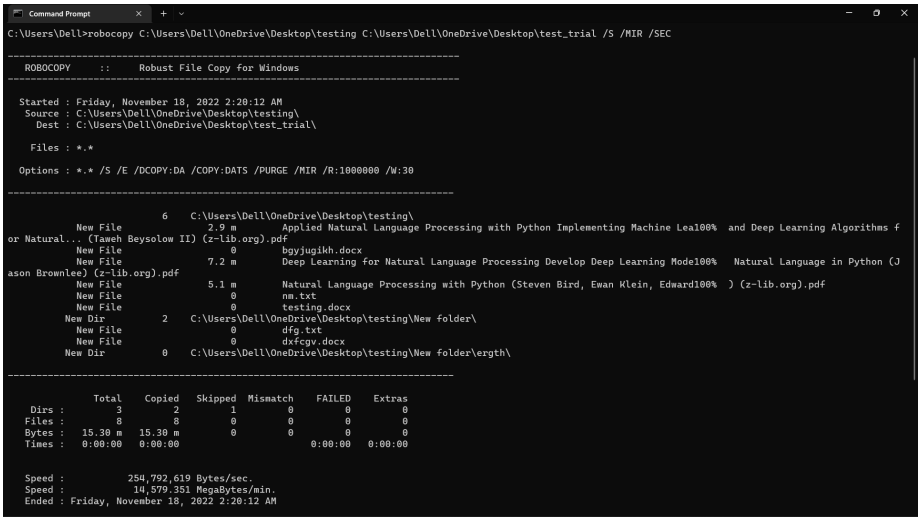


Figure 44: Multiple commands of Robocopy in the command-line interface.

But, when using multiple commands, as used above in Figure.8, in the Robocopy GUI an error is faced.

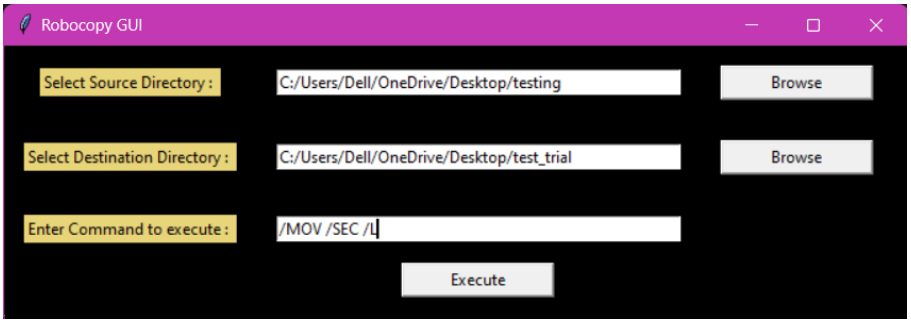


Figure 45: Executing multiple commands in the Robocopy GUI.

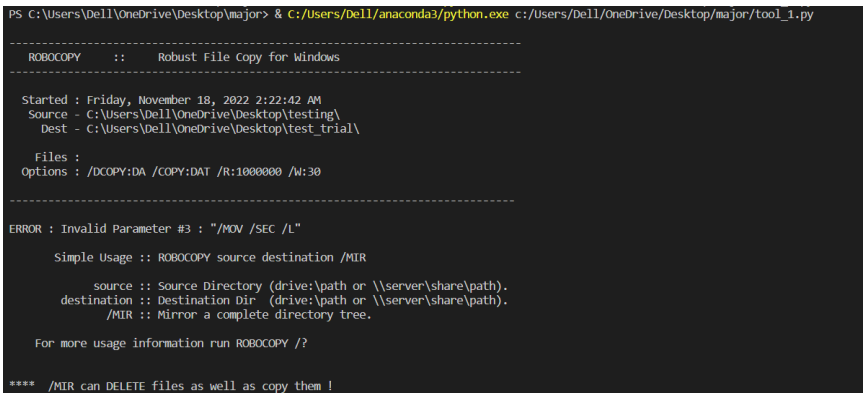


Figure 46: Error faced when executing multiple commands in the Robocopy GUI.

The above mentioned error can be overcome by modifying the GUI python code to accommodate multiple commands. Doing this will make the GUI interface more usable.

Another scope of improvement in the GUI can be the mentioning of the various commands that Robocopy can execute. For users who have never used the Robocopy command in their PC's, this will be a boon. A list of commands can be provided with their functions, making the Tool more user friendly.

- **Tool II**

Currently the database we used is offline,the users have the option to download the database on their device. We can switch to an online database, then users won't have to download the database and the database can be updated from time to time to add new numbers for better blocking of spam phone numbers.

In the drop down menu, as shown in Figure 26, additional features can be added to get better results of spam filtering.

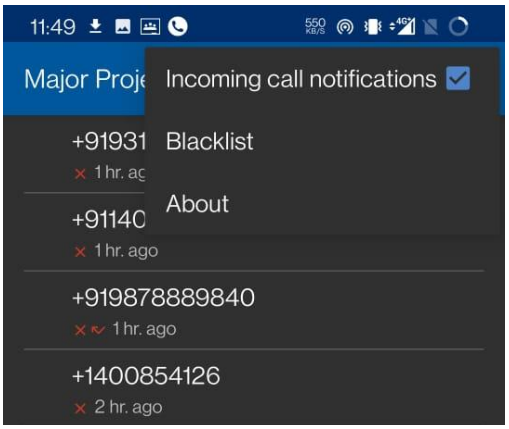


Figure 47: Drop down where more features can be added in the future.

Ratings/Reviews can be added for a caller's number. These ratings can be provided by the user. Users can provide a positive, negative or neutral rating based on their experience of the caller. A feature can be added such that numbers with negative ratings are automatically blocked.

- **Tool III**

  At this moment the tool can only be used to convert from DOC to PDF and visa versa. More different formats of documents can be added for the conversion of various file formats.

By default, Pandoc creates PDF files using pdflatex, which struggles to handle Unicode characters. When we convert Markdown pages with Unicode characters to PDF files, we run into issues.

```
Error producing PDF.
! LaTeX Error: Unicode character ╔» (U+05AF)
               not set up for use with LaTeX.

See the LaTeX manual or LaTeX Companion for explanation.
Type  H <return>  for immediate help.
 ...
```

Figure 48: Error when converting doc with Unicode characters.

- **Tool IV**

In some remote areas, people are not comfortable with new technologies, an SMS-based system can be set up for such cases to make it easier to receive and place orders via SMS.

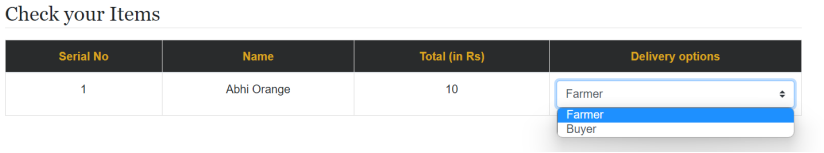


Figure 49: Delivery options during checkout.

Currently, there are two modes of delivery, either delivery by the farmer or pickup by the buyer. This can be expanded for a smoother faster delivery process by contacting external delivery services like Delhivery, India Post etc, for transporting goods to farther locations.

REFERENCES

[1] Bødker, M., Nielsen, L., & Orngreen, R. N. (2007). Enabling User Centered Design Processes in Open Source Communities. Lecture Notes in Computer Science, 10–18. doi:10.1007/978-3-540-73287-7_2

[2] Scacchi, W. (2007). Free/open source software development. Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering - ESEC-FSE '07. doi:10.1145/1287624.1287689

[3] Bødker, M., Nielsen, L., Orngreen, R.N. (2007). Enabling User Centered Design Processes in Open Source Communities. In: Aykin, N. (eds) Usability and Internationalization. HCI and Culture. UI-HCII 2007. Lecture Notes in Computer Science, vol 4559. Springer, Berlin, Heidelberg.

[4] IPEK—Institute of Product Engineering, Karlsruhe Institute of Technology (KIT), Kaiserstraße 10, 76131, Karlsruhe, Germany
Sebastian Helmstetter, Matthias Dörr, René Germann & Sven Matthiesen

[5] Lavelle, Claire & Konrad, Almudena. (2007). FriendlyRoboCopy: A GUI to RoboCopy for computer forensic investigators. Digital Investigation. 4. 16-23. 10.1016/j.diin.2007.01.001.

[6] Williams, Alex (July 9, 2012). "GitHub Pours Energies into Enterprise – Raises $100 Million From Power VC Andreessen Horowitz". *TechCrunch*. Archived from the original on September 19, 2020. Retrieved June 25, 2017. Andreessen Horowitz is investing an eye-popping $100 million into GitHub

[7] Nitnaware, Reshma. (2019). Basic Fundamental of Python Programming Language and The Bright Future. A Peer-Reviewed Journal About. VIII. 71-76.

[8]https://www.researchgate.net/publication/222426802_FreeOpen_Source_Software_Development_Recent_Research_Results_and_Methods

[9] Helmstetter, S., Dörr, M., Germann, R. *et al.* User-centered design of power tools: a generic process for evaluation of usability aspects. *Forsch Ingenieurwes* 86, 93–104 (2022). https://doi.org/10.1007/s10010-021-00546-y

[10] http://www.iocscience.org/ejournal/index.php/mantik/article/view/2533/2073

APPENDICES

1. Link to repository :

https://github.com/prishita-s/Developing_repository_of_user_centric_tools_using_FOSS

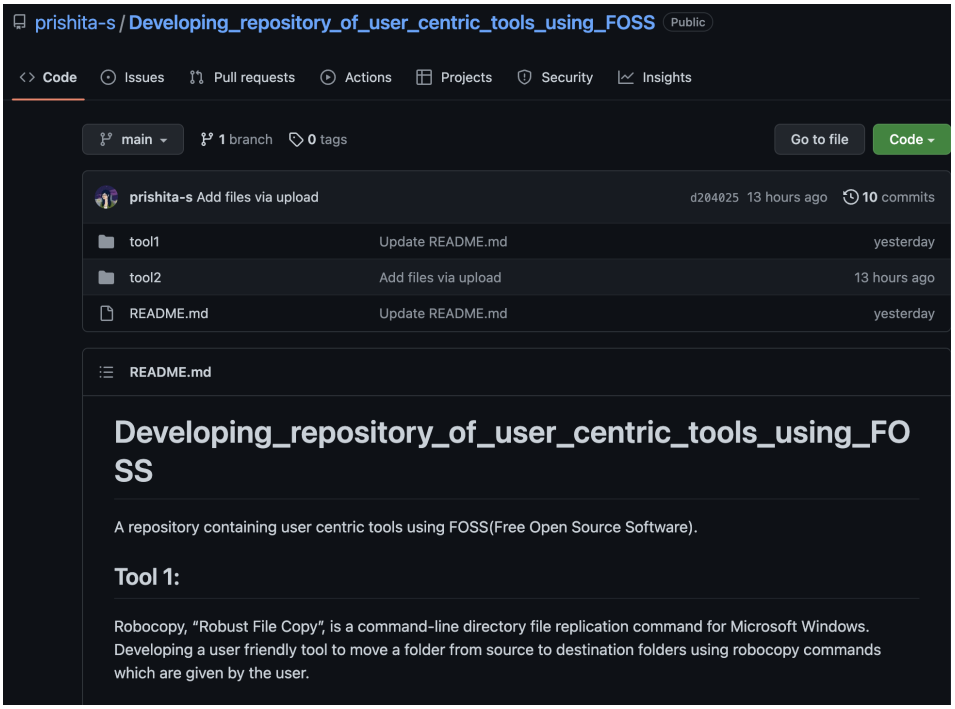


Figure 50: Repository Hosted on GitHub

2. Link to Tool 1 :

https://github.com/prishita-s/Developing_repository_of_user_centric_tools_using_FOSS/tree/main/tool1

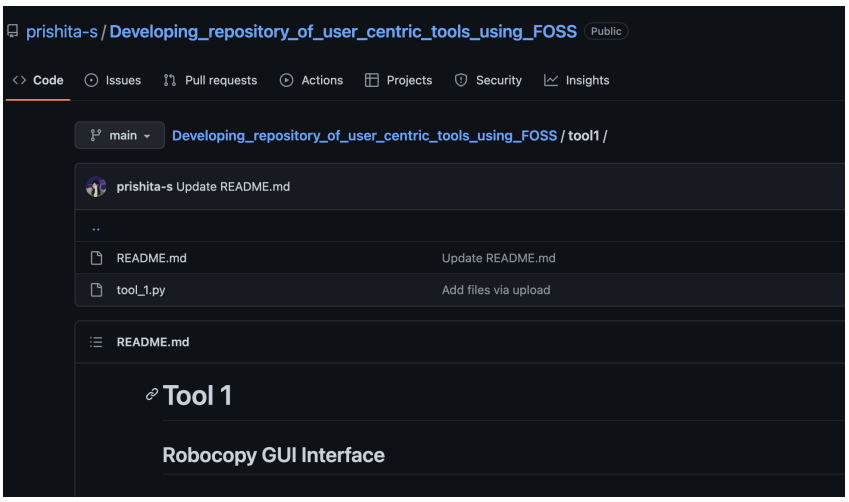


Figure 51: Tool 1 repository Hosted on GitHub

3. Link to Tool 2 :

https://github.com/prishita-s/Developing_repository_of_user_centric_tools_using_FOSS/tree/main/tool2

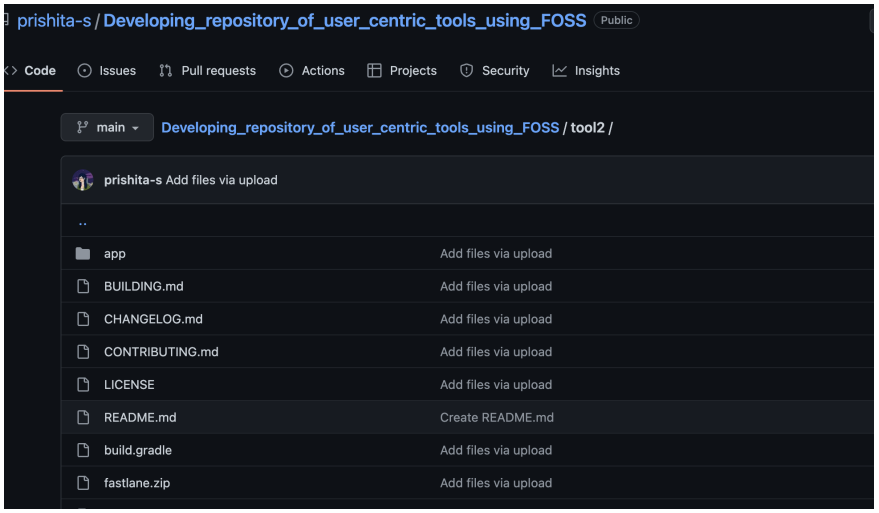


Figure 52: Tool 2 repository Hosted on GitHub

4. Link to Tool 3 :

https://github.com/prishita-s/Developing_repository_of_user_centric_tools_using_FOSS/tree/main/tool3

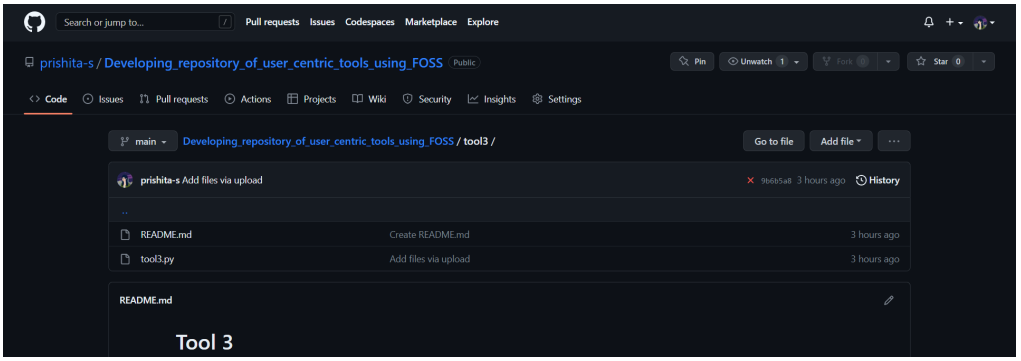


Figure 53: Tool 3 repository Hosted on GitHub

5. Link to Tool 4 :

https://github.com/prishita-s/Developing_repository_of_user_centric_tools_using_FOSS/tree/main/tool4

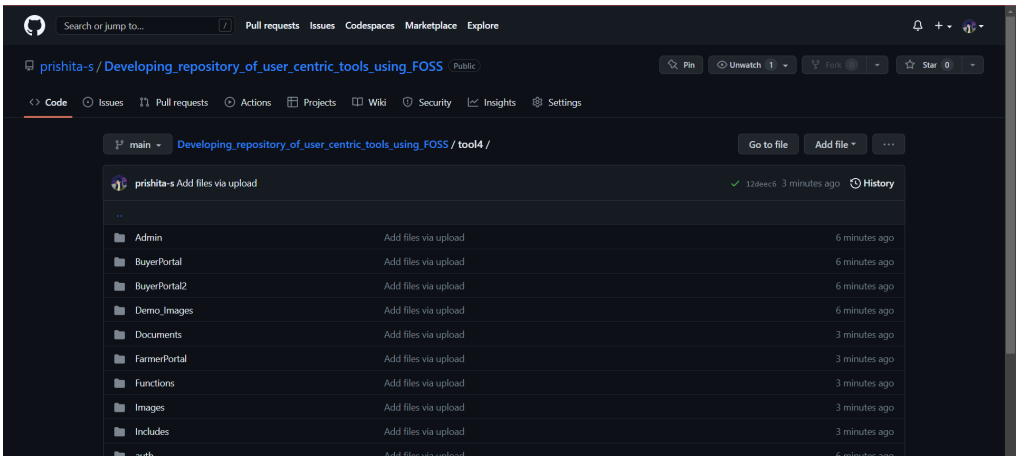


Figure 54: Tool 4 repository Hosted on GitHub