

# **REMOTE FILE MANAGEMENT SOFTWARE**

## **PROJECT REPORT**

*Submitted by*

**DHINOO [JEC20CS043]**

**CHRISTY GEORGE JOSEPH [JEC20CS040]**

**ALLAN GEO P [JEC20CS019]**

*to*

**The APJ Abdul Kalam Technological University**

*in partial fulfillment of the requirements for the award of the Degree  
of*

**BACHELOR OF TECHNOLOGY (B.TECH)**

*in*

**COMPUTER SCIENCE & ENGINEERING**

Under the guidance of

**Mr.SHAIJU PAUL**



**JUNE 2023**

**Department of Computer Science & Engineering**

**JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY**

**THRISSUR 679 531**

## DECLARATION

We the undersigned hereby declare that the project report "Remote File Management Software", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Mr.Shaiju Paul. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in this submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously used by anybody as a basis for the award of any degree, diploma or similar title of any other University.

Signature &

Name Of Students

DHINOO

(JEC20CS043)

CHRISTY GEORGE

JOSEPH

(JEC20CS040)

ALLAN GEO P

(JEC20CS019)

Place:CHERUTHURUTHY

Date

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**JYOTHI ENGINEERING COLLEGE, CHERUTHURUTHY**

**THRISSUR 679 531**



**CERTIFICATE**

This is to certify that the report entitled Title **REMOTE FILE MANAGEMENT SOFTWARE** submitted by **DHINOO(JEC20CS043),CHRISTY GEORGE JOSEPH (JEC20CS040), ALLAN GEO P(JEC20CS019)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in **Computer Science & Engineering** is a bonafide record of the project work carried out by them under my/our guidance and supervision.. This report in any form has not been submitted to any other University or Institute for any purpose.

Ms.Aswathy Wilson &

Ms.Sobha Xavier

Project Coordinators

Mr.Shaiju Paul

Project Supervisor

Dr.Saju P John

Head Of The Dept

## **DEPARTMENT VISION**

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

## **DEPARTMENT MISSION**

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department
- To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect

## **PROGRAMME EDUCATIONAL OBJECTIVES (PEO's)**

- The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.
- The graduates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.
- The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

## **PROGRAMME OUTCOMES (PO's)**

1. Ability to apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Ability to Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Ability to design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Ability to use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Ability to create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. Ability to apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Ability to understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ability to apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. Ability to communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Ability to recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **PROGRAMME SPECIFIC OBJECTIVES (PSO's)**

- An ability to apply knowledge of data structures and algorithms appropriate to computational problems.
- An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.
- An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.
- An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and trade-offs involved in design choices.

## ACKNOWLEDGEMENT

We take this opportunity to express our heartfelt gratitude to all respected personalities who had guided, inspired and helped us in the successful completion of this seminar. First and foremost, we express our thanks to **The Lord Almighty** for guiding us in this endeavour and making it a success.

We take immense pleasure in thanking the **Management** of Jyothi Engineering College and **Dr. Jose P Therattil**, principal, Jyothi Engineering College for having permitted us to carry out this project. Our sincere thanks to **Dr. Saju P John**, Head of the Department of Computer Science and Engineering for permitting us to make use of the facilities available in the department to carry out the project successfully.

We express our sincere gratitude to **Ms. Aswathy Wilson & Ms. Sobha Xavier** project Coordinator for their invaluable supervision and timely suggestions. We are very happy to express our deepest gratitude to our mentor **Mr. Shaiju Paul**, Assistant Professor, Department of Computer Science and Engineering, Jyothi Engineering College for his able guidance and continuous encouragement.

Last but not least we extend our gratefulness to all teaching and non teaching staffs who directly or indirectly involved in the successful completion of this project and to all our friends who have patiently extended all sorts of help for accomplishing this undertaking.

## **ABSTRACT**

A remote file management software is a computer program that allows users to manage and access their files remotely from a computer or other device over a network or the internet. It enables users to perform various file-related tasks such as creating, editing, sharing, moving, copying, compressing and deleting files without having to be physically present at the location of the files. Remote file management software is commonly used by businesses and individuals to manage and share files across multiple devices, locations, and users. Our proposed remote file management software is an innovative solution that offers advanced file organization, server-side compression, and scalability. It allows users to efficiently manage their files with customizable folders and tags, as well as advanced search capabilities that make it easy to find specific files. Additionally, the server-side compression feature helps reduce the storage requirements for files, freeing up space and increasing efficiency. In comparison to market leaders in this space, such as Dropbox and Google Drive, the new software offers a more comprehensive file organization and search functionality, as well as an innovative server-side compression feature.



# CONTENTS

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Objectives . . . . .	1
1.3 Organization Of the Project . . . . .	1
<b>2 LITERATURE SURVEY</b>	<b>3</b>
2.1 An Efficient Data Compression Scheme in InterPlanetary File System(IEEE) .	3
2.2 Data Consistency in Multi-cloud storage systems with passive servers and Non-Communicating Clients (IEEE) . . . . .	4
2.3 An Incentive-Compatible Mechanism for Decentralized Storage Network (IEEE) . . . . .	5
<b>3 METHADODOLOGY</b>	<b>6</b>
3.1 Existing Systems . . . . .	6
3.1.1 Disadvantages Of Existing System . . . . .	6
3.2 Problem Statement . . . . .	6
3.3 Proposed System . . . . .	7

3.4	Requirement Analysis . . . . .	7
3.4.1	Functional . . . . .	7
3.4.2	Non Functional . . . . .	7
3.5	Modules . . . . .	8
3.5.1	User Management Module . . . . .	8
3.5.2	Request Handler Module . . . . .	8
3.5.3	Upload Module . . . . .	8
3.5.4	Download Module . . . . .	8
3.5.5	Compressor Module . . . . .	8
3.5.6	Decompressor Module . . . . .	9
3.6	System Requirements And Specifications . . . . .	9
3.6.1	HTML . . . . .	9
3.6.2	CSS . . . . .	9
3.6.3	Python . . . . .	9
3.6.4	Bootstrap . . . . .	10
3.6.5	Pycharm . . . . .	10
3.6.6	Flask . . . . .	10
3.7	Data Flow Diagram . . . . .	10
3.7.1	Level 0 . . . . .	10
3.7.2	Level 1 . . . . .	11
3.7.3	Level 2 . . . . .	11
3.8	ER Diagram . . . . .	12
3.9	Use Case Diagram . . . . .	13
3.10	Implementation . . . . .	13

3.10.1	User Authentication and Authorization . . . . .	13
3.10.2	File Upload and Storage . . . . .	16
3.10.3	File Management . . . . .	16
3.10.4	File Compression . . . . .	17
3.10.5	Architecture Diagram . . . . .	19
3.10.6	Importing Libraries . . . . .	20
3.10.7	UI . . . . .	20
<b>4</b>	<b>RESULTS &amp; DISCUSSION</b>	<b>22</b>
<b>5</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>28</b>
5.1	Conclusion . . . . .	28
5.2	Future Scope . . . . .	28
<b>6</b>	<b>REFERENCES</b>	<b>29</b>

## List of Figures

3.1	level 0 . . . . .	11
3.2	level 1 . . . . .	11
3.3	level 2 . . . . .	12
3.4	ER Diagram . . . . .	12
3.5	Use Case Diagram . . . . .	13
3.6	User Registration . . . . .	14
3.7	User Login . . . . .	14
3.8	Access Control . . . . .	15
3.9	Logout . . . . .	15
3.10	File Upload And Storage . . . . .	16
3.11	File Management . . . . .	17
3.12	File Management . . . . .	17
3.13	File Compression . . . . .	18
3.14	Architecture Diagram . . . . .	19
3.15	UI . . . . .	21
4.1	Home page . . . . .	22
4.2	About page . . . . .	22
4.3	Services page . . . . .	23

4.4	Signup page . . . . .	23
4.5	Login Page . . . . .	24
4.6	DashBoard . . . . .	24
4.7	Upload File . . . . .	25
4.8	Mobile View(RESPONSIVE) . . . . .	26
4.9	Multiple View . . . . .	27
4.10	Logout page . . . . .	27

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

Remote file management software refers to applications or tools that enable users to manage files and folders located on remote servers or networked storage devices. These software solutions provide a convenient way to access, organize, transfer, compress and manipulate files without physically being present at the storage location.

### 1.2 Objectives

The objectives of remote file management software are to enable remote access and management of files and folders located on remote servers, simplify file organization and transfer, enhance file sharing and collaboration capabilities, ensure data security through encryption and access controls, streamline file operations for improved efficiency, provide versioning and control features, enhance productivity and enable efficient remote work, and offer monitoring and reporting functionalities to track file system activities and optimize file management processes.

### 1.3 Organization Of the Project

The report is organised as follows:

- . Chapter 1:Introduction-Gives an introduction to"remote file management"
- . Chapter 2:Literature Survey-Summarizes the various existing techniques that helped us in achieving the desired result.
- . Chapter 3:Methodology-Methods which are used in this project.

- . Chapter 4:Results and Discussion-The results of work and discussion.
- . Chapter 5:Conclusion Future scope-The chapter gives a conclusion of the overall work along with the future scope of implementation.
- . Chapter 6:References-Include the references for the project

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 An Efficient Data Compression Scheme in InterPlanetary File System(IEEE)

In this paper, we investigate the procedure of operations in IPFS. We find that the performance of IPFS is worse as the data size increases. To handle this issue, we present IPFSz which is a variant of IPFS to enable data compression functionality for better I/O performance and storage space consumption.

(1). In this scenario, the IPFSz uses the Compression API instead of UnixFS API because of compression requests. After then, the Compression API initiates the compression module to compress the data

( 2 ). In the compression module, the compression detector checks whether the request type is compression or decompression.

( 3 ). If the request type is compression, the detector calls compressor. The compressor compresses data.

#### ADVANTAGES:

- . compress/decompress the data during I/O operations by using a compression algorithm, manage the states of data.
- . Efficient file access control by managing the files in a single server.

#### DISADVANTAGES:

- . Quality of compressed image degrades with high ratio of compression. User can not get back original image after compression



## **2.2 Data Consistency in Multi-cloud storage systems with passive servers and Non-Communicating Clients (IEEE)**

( 1 ).Multi-cloud storage systems are becoming more popular due to the ever-expanding amount of consumer data. This growth is accompanied by increasing concerns regarding security, privacy, and reliability of cloud storage solutions

( 2 ). Multi-cloud storage systems can detect conflicts and preserve consistency through utilizing a centralized coordination point (e.g., server) that receives and logs the modification requests from the different clients (append-log).

( 3 ).One of the fundamental synchronization features is the ability to detect data conflicts and maintain data consistency . In general, data conflicts occur when multiple clients attempt to modify the same file at the same time.The cloud consumer typically has multiple computers sharing and concurrently accessing (reading and writing) the data

### **ADVANTAGES:**

- . Define data consistency in multi-cloud storage systems, identify how they can be violated, and introduce a new method that probably maintains the data consistency in these systems
- . Provide accessibility from anywhere and across different devices

### **DISADVANTAGES:**

- . The lack of communication between clients is a significant challenge to the consensus-based approach.
- . An additional and essential challenge that faces multi-cloud storage systems is the heterogeneity of consistency models followed by different providers.

### **2.3 An Incentive-Compatible Mechanism for Decentralized Storage Network (IEEE)**

The dominance of a few big companies in the storage market arising various concerns including single point of failure, privacy violation, and oligopoly. To eliminate the dependency on such a centralized storage architecture, several Decentralized Storage Network (DSN) schemes such as Filecoin, Sia, and Storj have been introduced. The outcome of this paper is a new incentive-compatible mechanism designed carefully for the blockchain-based DSN. The proposed mechanism utilizes different tools including game-theory, smart-contract, oracle network, and Merkle tree to improve the security and performance of storage verification in DSN

#### **ADVANTAGES:**

- . The primary goal of DSN's mechanism design is to ensure that the storage provider stores the client's data and returns it upon the client's request following the SLA.
- . Less burden to higher management
- . Prevent service denying attack
- . Moreover, our scheme prevents the dishonest storage provider to deliver PoS to the DSN while refusing the storage service to the client.

#### **DISADVANTAGES:**

- . Not appropriate for huge companies
- . Chance of conflict and misunderstanding
- . Difficult to maintain proper coordination

## **CHAPTER 3**

### **METHADODOLOGY**

#### **3.1 Existing Systems**

Google Cloud Storage: Google Cloud Storage is another cloud-based storage solution that allows you to store and access your files from anywhere.

Dropbox: Dropbox is a popular file hosting service that allows you to store, access, and share your files from anywhere.

Our software offers a more comprehensive file organization and search functionality, as well as an innovative server-side compression feature that reduces storage requirements.

##### **3.1.1 Disadvantages Of Existing System**

In existing systems there is loss of data takes place while compress files. Our software offers a more comprehensive file organization and search functionality, as well as an innovative server-side compression feature that reduces storage requirements. They can result in a loss of quality, particularly noticeable in media files. There is also a slight risk of file corruption during compression or decompression, leading to data loss.

#### **3.2 Problem Statement**

The proposed problem that remote file management software aims to solve is the challenge of efficiently managing files across multiple devices, locations, and users. With the increasing amount of data generated by individuals and organizations, the need for a centralized and accessible solution for managing files has become critical.

### **3.3 Proposed System**

The proposed solution is a remote file management software that provides users with a centralized and secure platform for managing their files. The software is designed to be accessible from any device with an internet connection, enabling users to access and manage their files from anywhere. Additionally, the server-side compression feature helps to reduce the storage requirements for files, freeing up space and increasing efficiency.

### **3.4 Requirement Analysis**

#### **3.4.1 Functional**

- . User Management
- . File Management
- . File Organization
- . Searching
- . Sharing
- . File Compression

#### **3.4.2 Non Functional**

- . Performance
- . Scalability
- . User Interface
- . Compatibility
- . Availability

## **3.5 Modules**

### **3.5.1 User Management Module**

The user module in a compression app serves as the interface through which users interact with the application's compression functionalities. It provides intuitive controls and displays relevant information, such as file sizes, compression ratios.

### **3.5.2 Request Handler Module**

The request handler module in a compression app acts as the intermediary between the user module and the core compression functionality. It receives requests and commands from the user module, processes them, and coordinates the necessary operations to fulfill those requests.

### **3.5.3 Upload Module**

The upload module in a compression app facilitates the process of uploading files from various sources into the application for compression. It provides functionality for users to select and upload files from their local storage, import files from cloud storage services.

### **3.5.4 Download Module**

The download module in a compression app facilitates the retrieval and saving of compressed files to various destinations after the compression process is complete. This ensures that the installed modules do not interfere with other projects or the system-wide Python installation.

### **3.5.5 Compressor Module**

The compressor module in a compression app is the core component responsible for executing the actual compression algorithms and reducing the size of files. The compressor module

utilizes a lossless compression technique to efficiently reduce the file size.

### **3.5.6 Decompressor Module**

The decompressor module in a compression app is a vital component responsible for unpacking or restoring compressed files back to their original format. The decompressor module ensures the accurate and efficient restoration of the files while preserving their integrity and minimizing data loss.

## **3.6 System Requirements And Specifications**

### **3.6.1 HTML**

When working on a project using HTML, it is important to plan the structure and content of your webpage, create the necessary HTML elements and tags, and add metadata and CSS stylesheets to enhance the appearance and functionality of the webpage.

### **3.6.2 CSS**

CSS (Cascading Style Sheets) is a crucial component in project making with HTML. It allows you to control the visual presentation of your web pages, including aspects like layout, colors, fonts, and animations. It allows developers to customize colors, fonts, sizes, spacing, and other design aspects of web pages. CSS works by selecting HTML elements using selectors and applying specific styles to them.

### **3.6.3 Python**

Python is a versatile and widely used programming language that is well-suited for project development. It offers a rich ecosystem of libraries and frameworks, making it ideal for various applications.

### **3.6.4 Bootstrap**

Bootstrap is a popular front-end framework that simplifies web development by providing a collection of pre-built CSS styles, JavaScript components, and responsive design templates. It allows developers to create modern and responsive websites quickly and efficiently.

### **3.6.5 Pycharm**

PyCharm is a popular integrated development environment (IDE) for Python programming that provides a comprehensive set of tools and features to streamline the development process. It offers a user-friendly interface, advanced code editor with powerful debugging capabilities

### **3.6.6 Flask**

Flask is a lightweight and versatile web framework for Python that allows developers to quickly build web applications. It provides a simple and flexible structure, enabling easy routing, handling requests and responses, and rendering templates. It also supports extensions that add additional functionality such as authentication, database integration, and API development. Flask's simplicity, flexibility.

## **3.7 Data Flow Diagram**

### **3.7.1 Level 0**

In a Data Flow Diagram (DFD), Level 0 represents the highest level of abstraction. It provides an overview of the system or process being modeled. The Level 0 DFD is useful for understanding the overall scope and boundaries of the system, as well as the high-level interactions between the system and external entities.

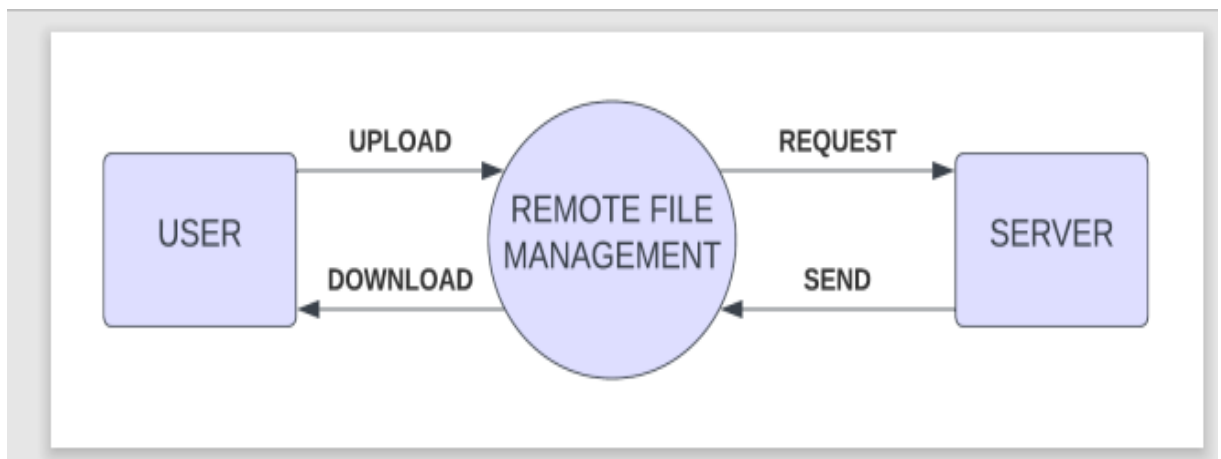


Figure 3.1: level 0

### 3.7.2 Level 1

In a Data Flow Diagram (DFD), Level 1 provides a more detailed view of the system or process that was represented at Level 0.

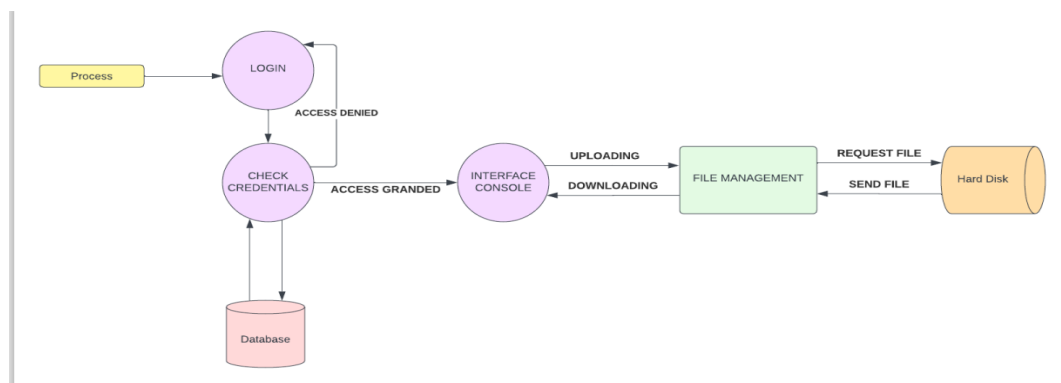
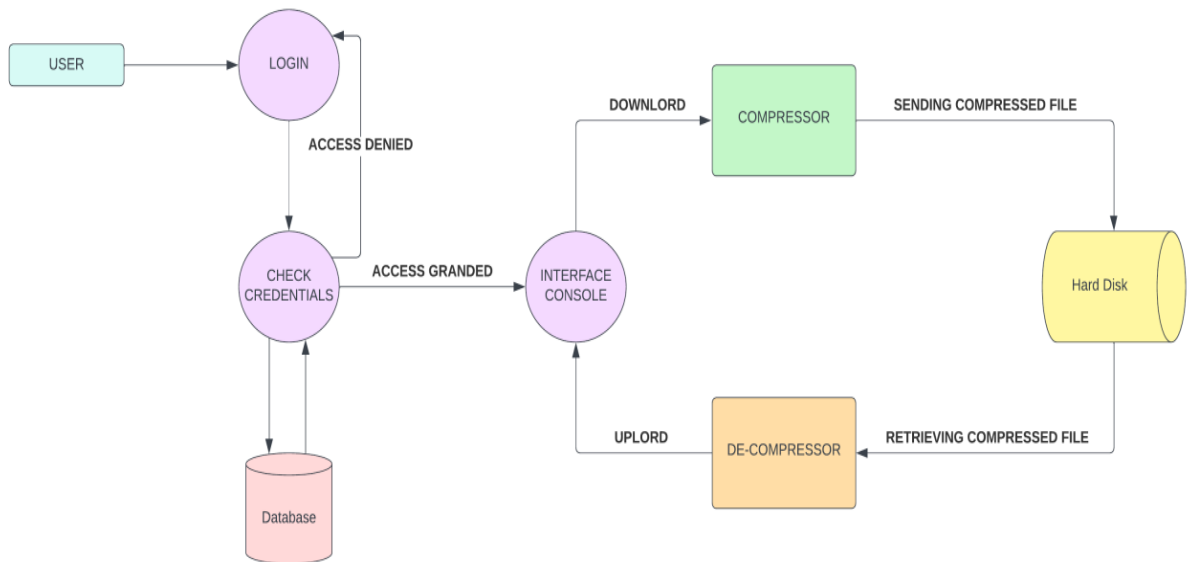


Figure 3.2: level 1

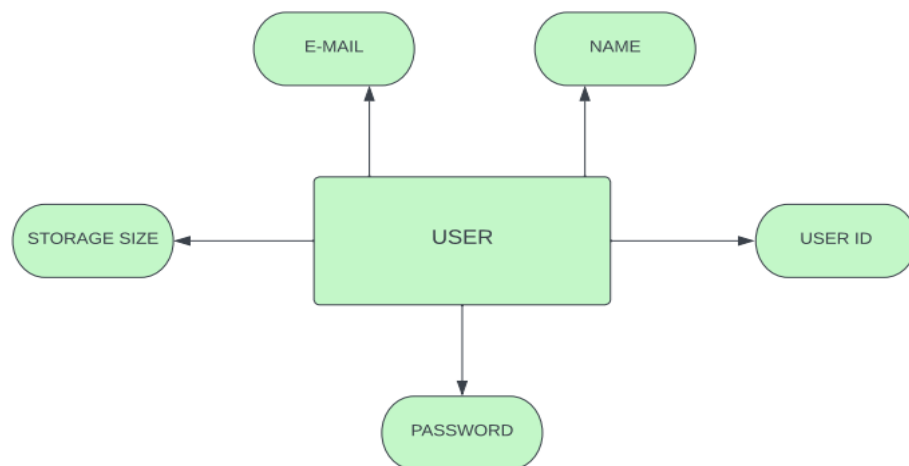
### 3.7.3 Level 2

Level 2 provides an even more detailed view of the system or process than Level 1. It further decomposes the subprocesses or functions from Level 1.



**Figure 3.3: level 2**

### 3.8 ER Diagram

**Figure 3.4: ER Diagram**

### 3.9 Use Case Diagram

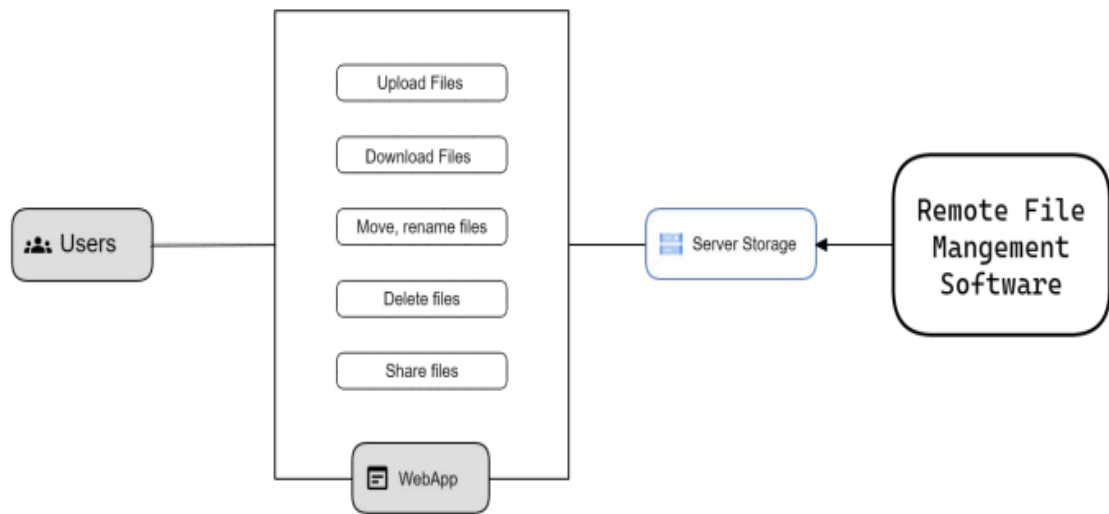


Figure 3.5: Use Case Diagram

### 3.10 Implementation

Developing a complete remote file management software involves multiple components and functionalities. Here is a high-level overview of the implementation steps and features that can be included:

#### 3.10.1 User Authentication and Authorization

Implementing user registration and login functionality

##### User Registration

1. Create a registration form where users can input their credentials, username and password.
2. On form submission, validate and securely store the user's credentials in a database.

```
@app.route('/signup', methods=['POST'])
def signup1():
    usr = request.form['uname']
    psw = request.form['psw']
    if usr==" " or psw==" ":
        return "All fields must be filled";
    global user_details
    for d in user_details:
        if d["user"] == usr:
            return "User exists try another name"
    user_details.append({"user": usr, "password": psw})
    f = open('user_details.txt', 'w')
    json.dump(user_details, f)
    f.close()
    os.mkdir(path=main_path + r'/' + usr)
    return redirect('/login')
```

Figure 3.6: User Registration

## User Login

1. Create a login form where users can enter their credentials.
2. On form submission, validate the provided credentials against the stored values in the database and If the credentials are valid, generate a new unique token for the user.
3. Store the token as a cookie in the user's browser, setting an expiration time if desired and associate the token with the user in the server-side storage

```
@app.route('/login', methods=['POST'])
def login1():
    usr = request.form['uname']
    psw = request.form['psw']
    log = {'user': usr, 'password': psw}
    if log in user_details:
        global Sessions
        Sessions.append(usr)
        token=str(random.randint(100000, 999999))
        Session_Tokens.append(token)
        resp = make_response(redirect('/dashboard'))
        resp.set_cookie('user', usr)
        resp.set_cookie('token', token)
        return resp
    else:
        return "Incorrect User Name or password Try Again"
```

Figure 3.7: User Login

## Access Control

1. For protected resources or pages, check if the user has a valid token in their cookie.
2. On each request, retrieve the token from the user's cookie and validate it against the server-side storage and if the token is valid and matches the user, grant access to the requested resource.
3. If the token is invalid or expired, redirect the user to the login page.

```
@app.route('/dashboard')
def dashboard():
    get_user = request.cookies.get('user')
    get_token = request.cookies.get('token')
    if get_user in Sessions and get_token in Session_Tokens:
        return render_template('/dashboard.html', user=get_user.upper())
    else:
        return redirect('/login')
```

Figure 3.8: Access Control

## Logout

1. Provide a logout functionality where users can terminate their session.
2. On logout, remove the token from the user's cookie and invalidate it on the server-side.
3. Redirect the user to the login page.

```
@app.route('/signout', methods=['GET'])
def signout():
    get_user = request.cookies.get('user')
    get_token = request.cookies.get('token')
    if get_user in Sessions and get_token in Session_Tokens:
        Sessions.remove(get_user)
        Session_Tokens.remove(get_token)
        resp = make_response(redirect('/login'))
        resp.set_cookie('user', '', expires=0)
        resp.set_cookie('token', '', expires=0)
        return resp
    else:
        return redirect('/login')
```

Figure 3.9: Logout

### 3.10.2 File Upload and Storage

- Allows users to upload files from their local devices.
- Implementing server-side file storage and organization.

```
@app.route('/upload', methods=['POST'])
def upload1():
    get_user = request.cookies.get('user')
    get_token = request.cookies.get('token')
    if get_user in Sessions and get_token in Session_Tokens:
        path = main_path + r'\\' + get_user + r'\\'
        f = request.files.getlist('folder')
        for file in f:
            vari = file.filename
            vari = vari.replace('/', '\\')
            makit = vari.split('\\')
            if not os.path.exists(path + makit[0]):
                os.mkdir(path + makit[0])
            varit = vari.rsplit('\\', 1)
            varity = varit[len(varit) - 2]
            if os.path.exists(path + varity):
                filer = open(os.path.join(path, vari), "wb")
                filer.write(zlib.compress(file.read()))
                filer.close()
            else:
                os.makedirs(path + varity)
                filer = open(os.path.join(path, vari), "wb")
                filer.write(zlib.compress(file.read()))
                filer.close()
        return redirect('/dashboard')
    else:
        return 'ERROR'
```

Figure 3.10: File Upload And Storage

### 3.10.3 File Management

- Provide a user interface to browse, view, and manage files and folders allow users to upload and delete files and folders.
- Implemented search functionality to find files based on name and types.

```

@app.route('/search/<path:filename>', methods=['GET', 'POST'])
def search(filename):
    get_user = request.cookies.get('user')
    get_token = request.cookies.get('token')
    if get_user in Sessions and get_token in Session_Tokens:
        path = main_path + r'\\' + get_user
        var = []
        var += search_folder(path, filename, '')
        var += [{"name": "no"}]
        return var
    else:
        return "ERROR"

def search_folder(path, filename, search_term, carry):
    carry += filename + '/'
    if filename != '':
        path = path + r'\\' + filename
        var = []
        for i in os.listdir(path):
            if i.lower().startswith(search_term.lower()):
                if os.path.isfile(path + r'\\' + i):
                    var += [{"name": "" + i + "", "size": "" + str(os.path.getsize(path + r'\\' + i)) + "", "type": "" + pathlib.Path(
                        path + r'\\' + i).suffix + "", "path": "" + carry + i + ""},]
                else:
                    var += [{"name": "" + i + "", "size": "" + str(
                        get_folder_size(path + r'\\' + i)) + "", "type": "folder", "path": "" + carry + i + ""},]
            if not os.path.isfile(path + r'\\' + i):
                var += search_folder(path, i, search_term, carry)
        return var

```

Figure 3.11: File Management

```

def sortfiletypes(path, filename, search_term, carry):
    carry += filename + '/'
    if filename != '':
        path = path + r'\\' + filename
        var = []
        for i in os.listdir(path):
            if os.path.isfile(path + r'\\' + i):
                if pathlib.Path(path + r'\\' + i).suffix in search_term:
                    var += [{"name": "" + i + "", "size": "" + str(os.path.getsize(path + r'\\' + i)) + "", "type": "" + pathlib.Path(
                        path + r'\\' + i).suffix + "", "path": "" + carry + i + ""},]
                else:
                    var += sortfiletypes(path, i, search_term, carry)
        return var

```

Figure 3.12: File Management

### 3.10.4 File Compression

File compression is implemented by using the zlib algorithm, which is a widely used compression algorithm that provides lossless data compression. At its core, zlib uses the DEFLATE algorithm, which combines LZ77 (a sliding window compression algorithm) and Huffman coding (a variable-length prefix coding algorithm). Steps involved in zlib compression:

## Deflate Compression

1. The data to be compressed is divided into a series of blocks.
2. Within each block, LZ77 compression is applied to find repeated sequences of data. Instead of storing the complete repeated sequence, LZ77 uses a combination of a backward reference (offset) and a length to represent the repeated data.
3. The compressed LZ77 data is then passed through Huffman coding, where fixed and dynamic Huffman tables are used to assign shorter codes to frequently occurring symbols and longer codes to less frequent symbols so it reduces the size of compressed data.

```
int ZEXPORT compress2(dest, destlen, source, sourcelen, level)
Bytef *dest;
uLongf *destlen;
const Bytef *source;
uLong sourcelen;
int level;
{
    z_stream stream;
    int err;
    const uInt max = (uInt)-1;
    uLong left;

    left = *destlen;
    *destlen = 0;

    stream.zalloc = (alloc_func)0;
    stream.zfree = (free_func)0;
    stream.opaque = (voidpf)0;

    err = deflateInit(&stream, level);
    if (err != Z_OK) return err;

    stream.next_out = dest;
    stream.avail_out = 0;
    stream.next_in = (z_const Bytef *)source;
    stream.avail_in = 0;

    do {
        if (stream.avail_out == 0) {
            stream.avail_out = left > (uLong)max ? max : (uInt)left;
            left -= stream.avail_out;
        }
        if (stream.avail_in == 0) {
            stream.avail_in = sourcelen > (uLong)max ? max : (uInt)sourcelen;
            sourcelen -= stream.avail_in;
        }
        err = deflate(&stream, sourcelen ? Z_NO_FLUSH : Z_FINISH);
    } while (err == Z_OK);

    *destlen = stream.total_out;
    deflateEnd(&stream);
    return err == Z_STREAM_END ? Z_OK : err;
}

/* =====
*/
int ZEXPORT compress(dest, destlen, source, sourcelen)
Bytef *dest;
uLongf *destlen;
const Bytef *source;
uLong sourcelen;
{
    return compress2(dest, destlen, source, sourcelen, Z_DEFAULT_COMPRESSION);
}
```

Figure 3.13: File Compression



## Compression Ratio and Efficiency

1. The compression ratio achieved by zlib depends on the nature of the input data. Highly repetitive or redundant data tends to compress better, while already compressed data or random data may not compress significantly.
2. The efficiency of zlib compression is attributed to the combination of LZ77 and Huffman coding. LZ77 finds repeated patterns and replaces them with references, and Huffman coding assigns shorter codes to more frequent symbols, resulting in effective compression.

## Decompression

1. The zlib algorithm also supports decompression of the compressed data.
2. The compressed data is processed in reverse: Huffman decoding is applied to obtain the compressed LZ77 data, and then LZ77 decompression reconstructs the original data by resolving the backward references.

### 3.10.5 Architecture Diagram

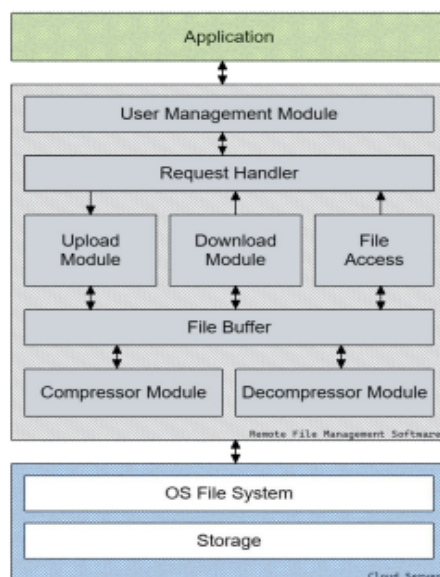
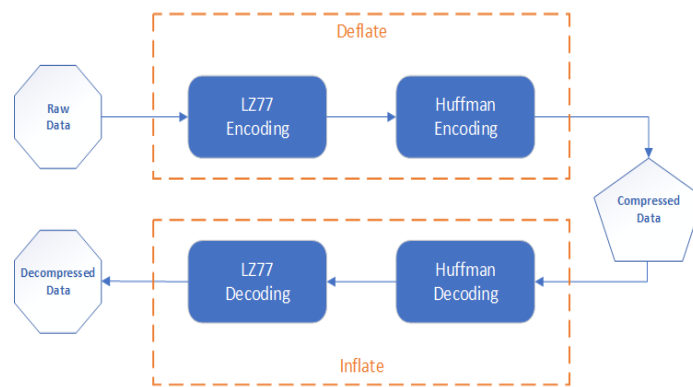


Figure 3.14: Architecture Diagram





### 3.10.6 Importing Libraries

- Font Awesome library

- Font Awesome icons library provides 519 free scalable vector icons. This library is completely free for both personal and commercial use. Originally designed for Bootstrap, these icons can be customized easily.

- Flask

- Flask is a popular Python web framework that allows you to build web applications. It is lightweight, flexible, and easy to use. Flask follows the WSGI (Web Server Gateway Interface) specification, which means it can work with any WSGI-compliant web server.

- Zlib

- The zlib library is a software library for data compression. It provides functions and algorithms for compressing and decompressing data using the zlib compression algorithm, which is based on the Deflate compression algorithm.

### 3.10.7 UI

The UI is created using HTML and CSS in Bootstrap and Tailwind framework.

User interface simplifies the entire model for the user. Helps the user to use the entire system

without knowing how it works

```
<!DOCTYPE html>
<html lang="en">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css">
<head...>
<script src="https://cdn.tailwindcss.com"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/js/all.min.js"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script src="https://unpkg.com/@popperjs/core@2.9.1/dist/umd/popper.min.js" charset="utf-8"></script>
<script...>
<body class="flex h-screen bg-white text-black">
<div class="flex flex-col flex-1 w-full overflow-y-auto">
  <header class="z-10 py-4 lg:h-20 h-28 bg-[#1e4356]"...>
    <div id="full" class="h-full overflow-y-auto">
      <div id="info0" class="h-12 sticky top-0 drop-shadow mb-6 ">
      </div>
      <div id="infolist" class="m-2 grid grid-cols-1 gap-4"></div>
      <div id="info" class="m-2 grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4"></div>
      <div id="infomob" class="m-2 grid grid-cols-2 gap-4"></div>
      <div id="contextMenu" class="hidden absolute bg-white text-xl shadow-2xl rounded py-2">
      <ul class="list-none">
        <li id="uploadFileItem" class="px-4 py-2 hover:bg-cyan-50 hover:cursor-pointer"><i class="fa-solid fa-file-arrow-up"></i> Upload File</li>
        <li id="uploadFolderItem" class="px-4 py-2 hover:bg-cyan-50 hover:cursor-pointer"><i class="fa-solid fa-upload"></i> Upload Folder</li>
      </ul>
      </div>
      <div id="contextMenuItem" class="hidden absolute bg-white text-xl shadow-2xl rounded py-2">
      <ul class="list-none">
        <li id="openItem" class="px-4 py-2 hover:bg-cyan-50 hover:cursor-pointer"><i class="fa-solid fa-up-right-from-square"></i> Open</li>
        <li id="itemDetails" class="px-4 py-2 hover:bg-cyan-50 hover:cursor-pointer"> <i class="fa-solid fa-info"></i> Details</li>
        <li id="deleteItem" class="px-4 py-2 hover:bg-cyan-50 hover:cursor-pointer"><i class="fa-solid fa-trash"></i> Delete</li>
        <li id="downloadItem" class="px-4 hidden py-2 hover:bg-cyan-50 hover:cursor-pointer"><i class="fa-solid fa-download"></i> Download</li>
      </ul>
      </div>
      <div id="info1" class="p-4"></div>
      <div id="alertbox" class="fixed hidden inset-0 flex items-center justify-center">
    <div class="bg-white shadow-2xl p-8 m-2 rounded-md relative"...>
  </div>
</div>
</div>
</div>
</body>
```

Figure 3.15: UI

## CHAPTER 4

# RESULTS & DISCUSSION

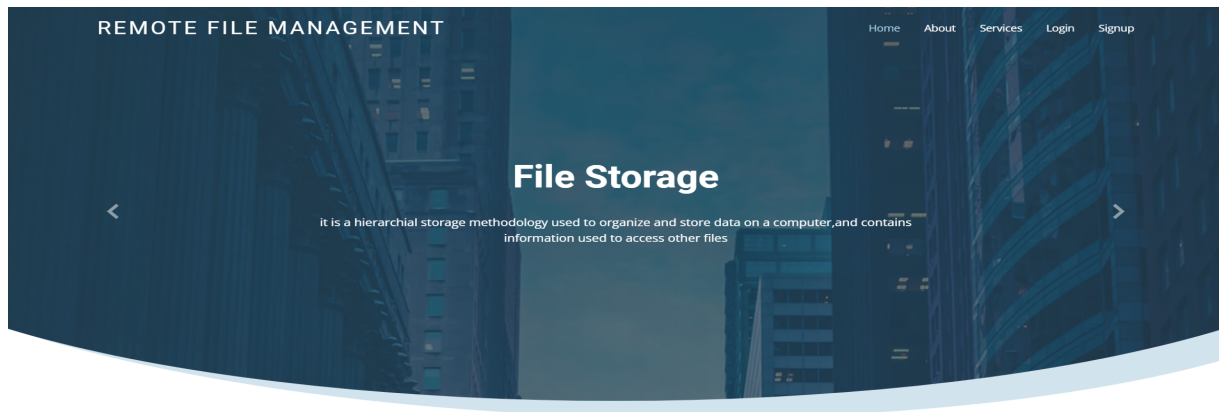


Figure 4.1: Home page

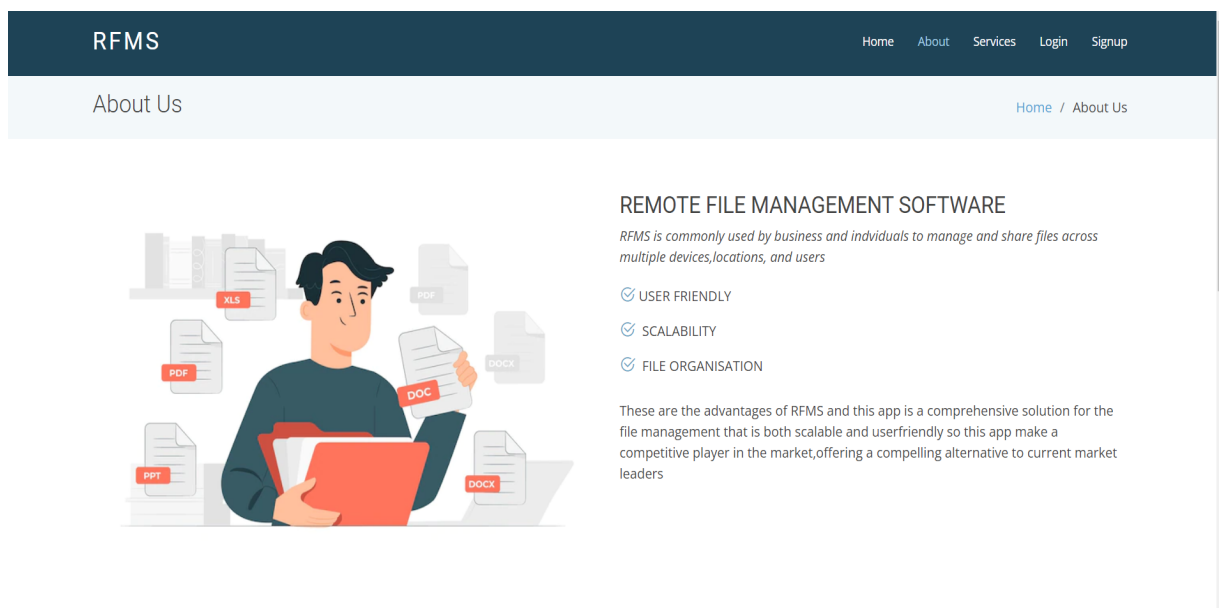
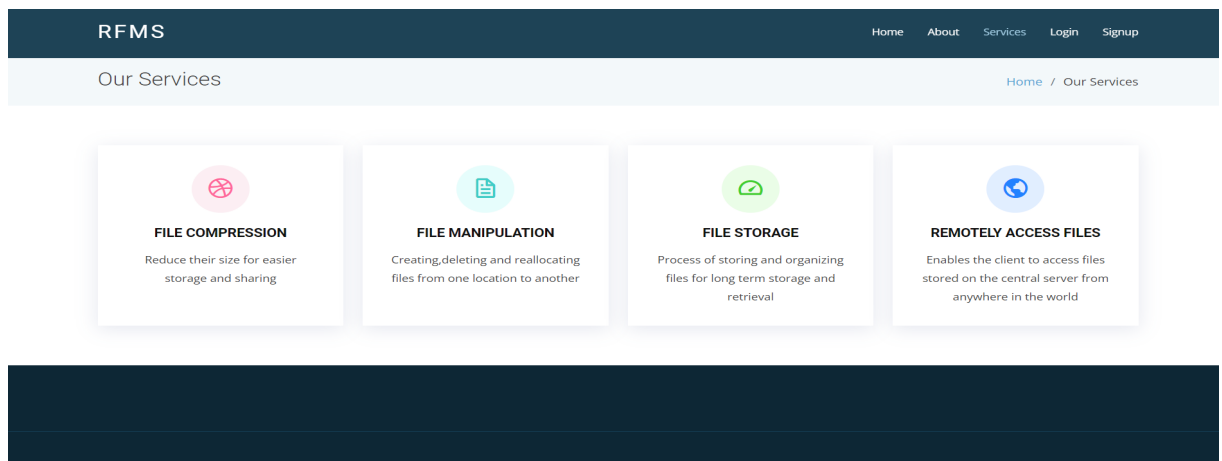
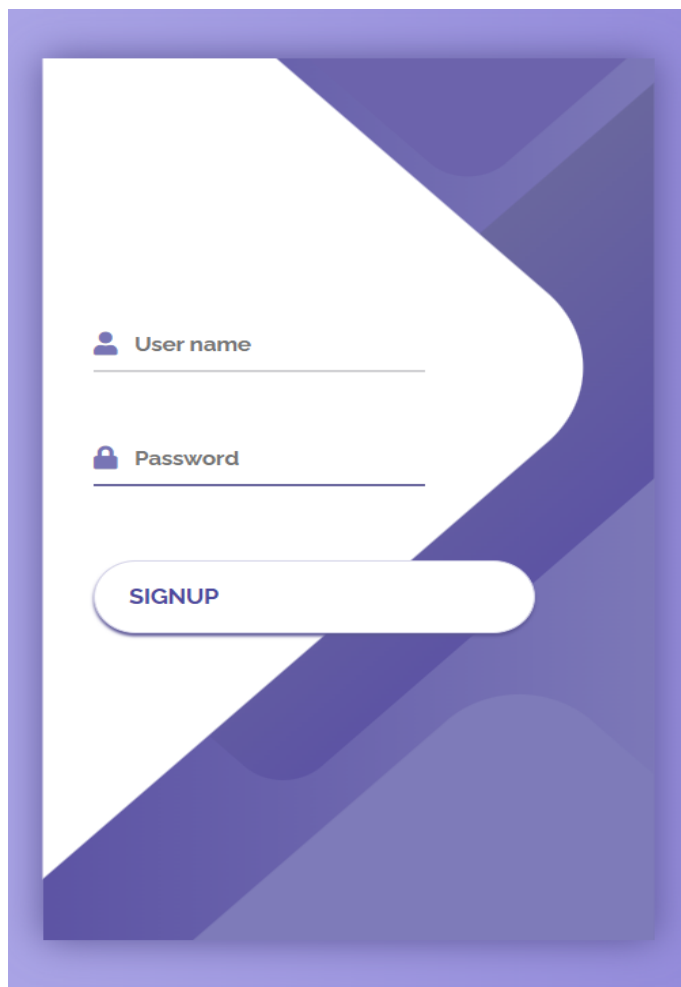


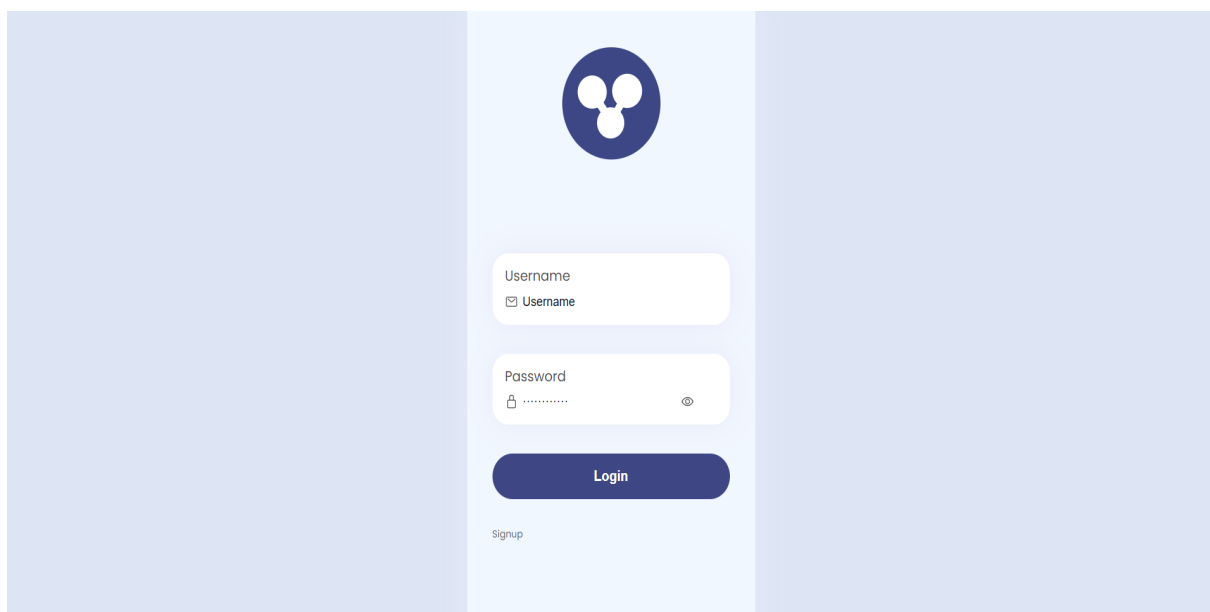
Figure 4.2: About page



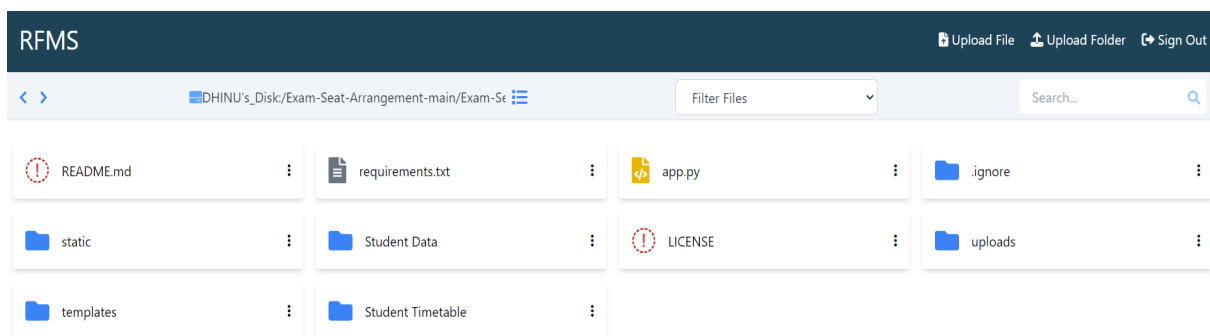
**Figure 4.3: Services page**



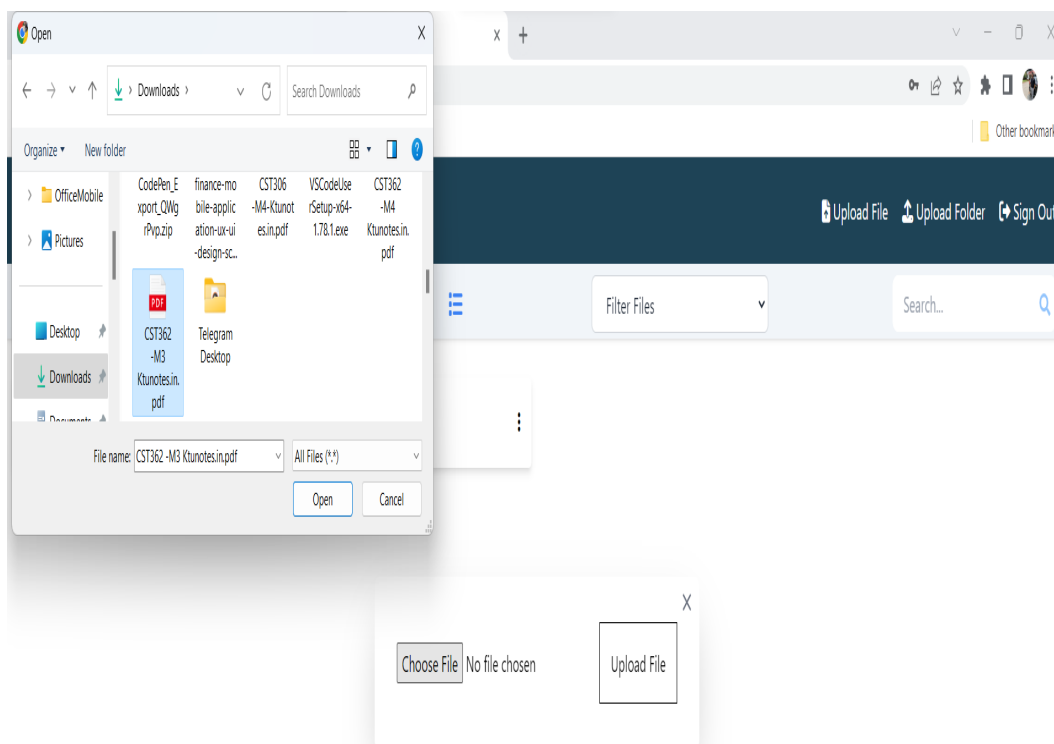
**Figure 4.4: Signup page**

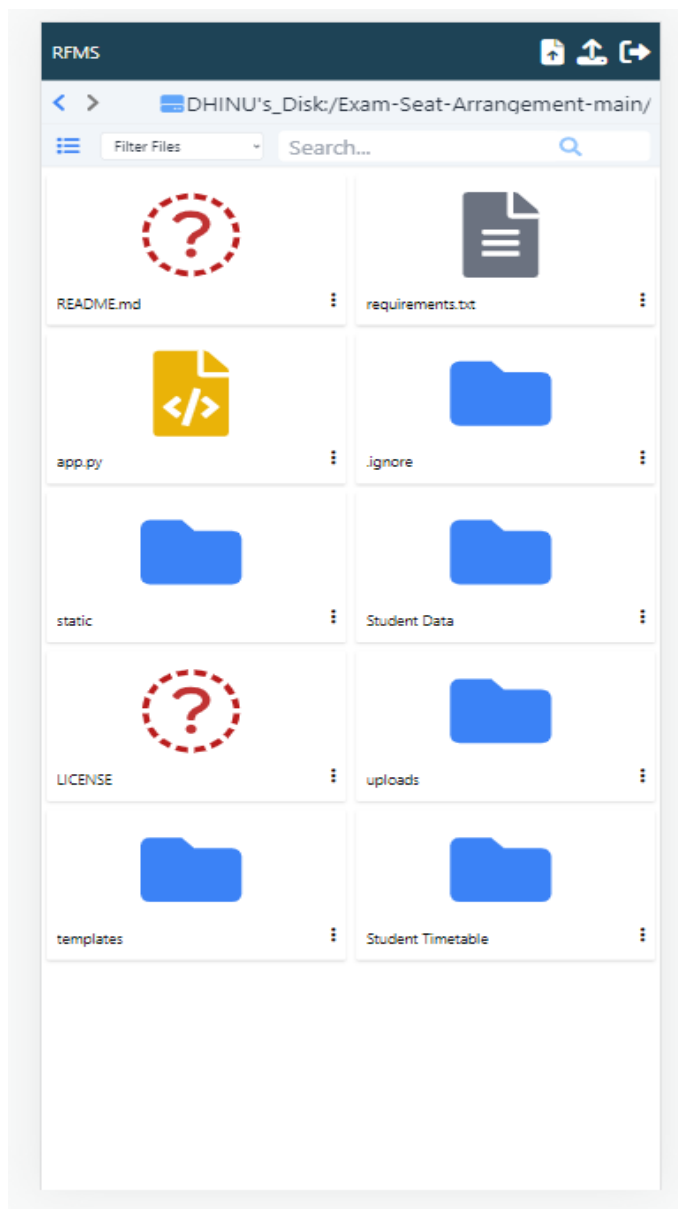


**Figure 4.5: Login Page**

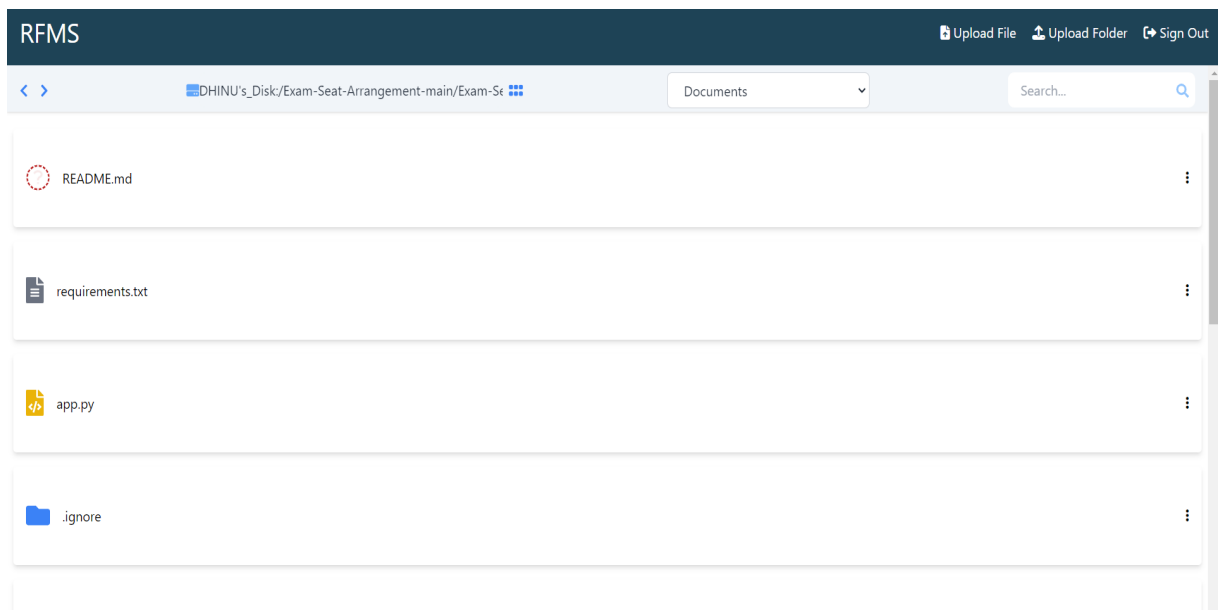


**Figure 4.6: DashBoard**

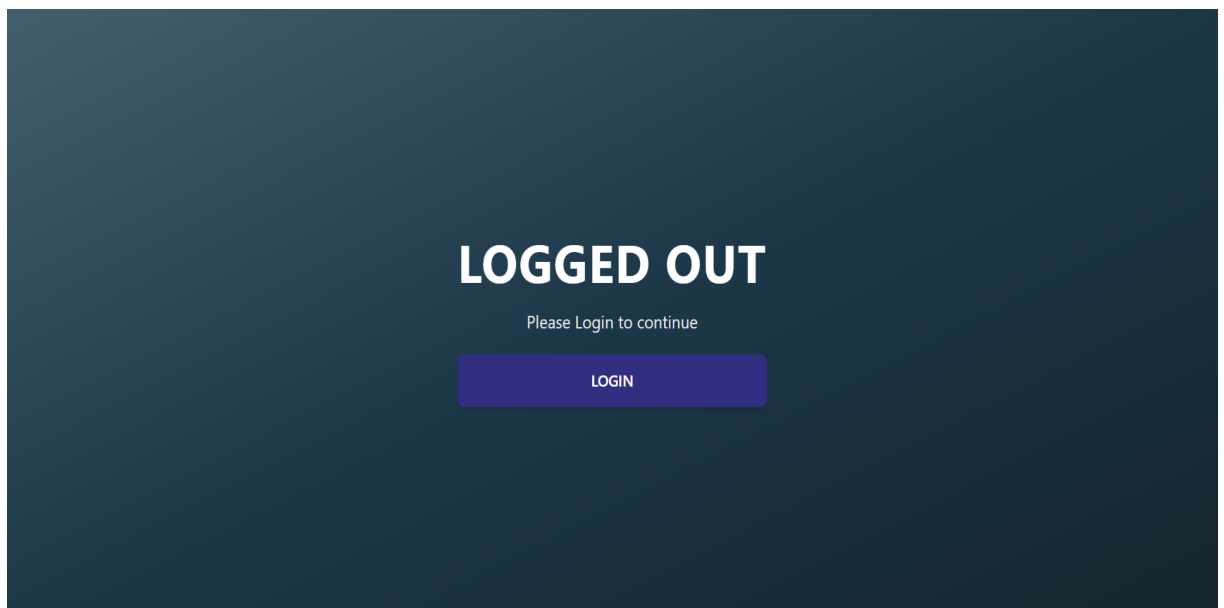
**Figure 4.7: Upload File**



**Figure 4.8: Mobile View(RESPONSIVE)**



**Figure 4.9: Multiple View**



**Figure 4.10: Logout page**



## **CHAPTER 5**

### **CONCLUSION & FUTURE SCOPE**

#### **5.1 Conclusion**

In conclusion, the development of our remote file management software has proven to be a significant success, providing users with a highly efficient and secure solution for accessing, organizing and compressing files remotely. Our software has greatly simplified file management tasks for individuals and businesses alike. By enabling seamless collaboration, efficient file sharing, and advanced security measures, our remote file management software has not only enhanced productivity but also ensured the protection of sensitive data. Overall, our software has effectively addressed the challenges associated with remote file management

#### **5.2 Future Scope**

The future scope for our remote file management software is promising. We envision incorporating advanced artificial intelligence algorithms to provide intelligent file organization and predictive search capabilities, further enhancing user productivity. Additionally, expanding the software's compatibility with emerging technologies such as blockchain and decentralized storage systems would offer enhanced security and data integrity. Furthermore, integrating collaboration features with real-time document editing and video conferencing capabilities would enable seamless teamwork and foster remote collaboration.

## CHAPTER 6

## REFERENCES

- [1] IEEE: <https://ieeexplore.ieee.org/Xplore/home.jsp>
- [2] Flask documentation: <https://flask.palletsprojects.com/en/2.2.x/>
- [3] Stackoverflow: <https://stackoverflow.com/>
- [4] WebDAV: <https://tools.ietf.org/html/rfc4918>
- [5] AWS: <https://docs.aws.amazon.com/>
- [6] Python: <https://docs.python.org/3/>
- [7] Google Drive: <https://developers.google.com/drive>
- [8] DropBox: [https://www.dropbox.com/developers/ documentation](https://www.dropbox.com/developers/documentation)