**A Project Report**

**on**

**Easy Share**

Submitted in partial fulfillment of the requirement of Project – VI

(BCA378CO)

Of

Bachelor of Computer Application 6th Semester

**Submitted to**

Purbanchal University

Biratnagar, Nepal

**Submitted By**

Lenish Yesmali Magar (352818)

Puran Gupta(352824)

Manish Kumar Shrestha(352819)

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

Aug 4, 2023

**A project Reports**

**on**

**Easy Share**

Submitted in partial fulfillment of the requirement of Project – VI

(BCA378CO)

Of

Bachelor of Computer Application 6th Semester

**Submitted to**

Purbanchal University

Biratnagar, Nepal

**Submitted By**

Lenish Yesmali Magar (352818)

Puran Gupta (352824)

Manish Kumar Shrestha(352819)

**Project Supervisor**

Rubim Shrestha

**KANTIPUR CITY COLLEGE**

Putalisadak, Kathmandu

Aug 4, 2023

**CERTIFICATE OF APPROVAL**

This is to certify the project entitled “**Easy Share**” is a project work Submitted by

Lenish Yesmali Magar (352818)

Puran Gupta (345158)

Manish Kumar Shrestha (352819)

In fulfillment for the degree of Bachelor in Computer Application

………………….

Kiran Khanal

Project Supervisor

Department of IT

………………….

Saroj Pandey

Head of Department

Department of IT

# Abstract

EasySent is an innovative solution for sharing files, designed to make file sharing easier in today's digital world. Traditional file-sharing methods can be tricky, with complex logins and limitations on file size. EasySent aims to change this by offering a simple, secure, and user-friendly approach.

At the core of EasySent is its ability to make file sharing effortless. Users can pick a file and create a special link for sharing, without the need for complicated logins. Additionally, EasySent allows for real-time communication and quick file transfers, making collaboration and content sharing smooth and efficient.

The platform uses advanced technology like Node.js, Express, Socket.IO, React with Vite, and Tailwind CSS. These technologies ensure that the user interface is responsive and easy to use while also maintaining security and scalability. EasySent even gives users the option to set when shared links expire, adding an extra layer of control.

Contents

[Abstract i](#_Toc144579190)

[Chapter 1: INTRODUCTION 1](#_Toc144579191)

[1.1 Project introduction 1](#_Toc144579192)

[1.2 Overview 1](#_Toc144579193)

[1.3 Problem Statement 2](#_Toc144579194)

[1.4 Objective of the Project 2](#_Toc144579195)

[1.5 Significance of The Project 2](#_Toc144579196)

[1.6 Features of the Project 3](#_Toc144579197)

[1.7 Scope and Limitation 3](#_Toc144579198)

[1.7.1 Scope 3](#_Toc144579199)

[1.7.2 Limitation 3](#_Toc144579200)

[Chapter 2: Literature Overview 4](#_Toc144579201)

[2.1 WeTransfer 4](#_Toc144579202)

[2.2 MediaFire 4](#_Toc144579203)

[Chapter 3: Methodology 6](#_Toc144579204)

[3.1 Software Development Life Cycle 6](#_Toc144579205)

[3.2 Tool and Technology Used 6](#_Toc144579206)

[3.3 Integrated Development Environment 7](#_Toc144579207)

[Chapter 4: System Analysis 8](#_Toc144579208)

[4.2 Requirement Analysis 8](#_Toc144579209)

[4.2.1 Functional requirement 8](#_Toc144579210)

[4.2.2 Non-Functional Requirement 9](#_Toc144579211)

[4.3 Feasibility Study 10](#_Toc144579212)

[4.3.1 Technical feasibility 10](#_Toc144579213)

[4.3.2 Schedule feasibility 10](#_Toc144579214)

[4.3.3 Economic feasibility 10](#_Toc144579215)

[4.4 Gantt Chart 10](#_Toc144579216)

[Chapter 5: SYSTEM DESIGN 11](#_Toc144579217)

[5.1 Context Level Diagram 11](#_Toc144579218)

[5.2 DFD level 1 11](#_Toc144579219)

[5.3 Use Case Diagram 12](#_Toc144579220)

[5.4 Relational Diagram 12](#_Toc144579221)

[CHAPTER 6: SYSTEM DEVELOPMENT AND IMPLEMENTATION 13](#_Toc144579222)

[6.1 Programming Platform 13](#_Toc144579223)

[6.2 Test Plan 14](#_Toc144579224)

[6.3 Features to be Tested 14](#_Toc144579225)

[6.4 Testing Tool 14](#_Toc144579226)

[6.5 Test cases 15](#_Toc144579227)

[6.5.1 Login 15](#_Toc144579228)

[6.5.1 Register 15](#_Toc144579229)

[6.5.1 Send and Receive File 15](#_Toc144579230)

[6.5.1 Add or Remove Files. 16](#_Toc144579231)

[6.5.1 Logs 16](#_Toc144579232)

[6.5.1 Delete Generated UniqueID 16](#_Toc144579233)

[Chapter 7 conclusion and Future Enhancement 17](#_Toc144579234)

[7.1 Conclusion 17](#_Toc144579235)

# Chapter 1: INTRODUCTION

## Project introduction

EasySent is a platform designed to streamline the process of sharing files. It offers a straightforward approach where users can select a file and generate a unique ID URL for sharing purposes. If both the sender and receiver are logged in, the platform automatically stores the shared URL and the received URL in MongoDB, providing a convenient way to access shared content. However, even if the sender or receiver is not logged in, the file transfer will work as usual, with the only difference being that the shared or received URL won't be stored in MongoDB.

The development of EasySent relied on technologies such as Node.js, Express, Socket.io, JavaScript, Vite, React.js, and Tailwind CSS. These technologies were chosen to ensure a user-friendly and efficient file sharing experience. EasySent aims to simplify the file sharing process, offering both convenience and flexibility to users, whether they are logged in or not.

## Overview

EasySent is a user-friendly file-sharing platform designed to simplify the process of sharing files. Users can effortlessly select a file and generate a unique ID URL for sharing purposes, making the sharing process straightforward and accessible to all. Notably, EasySent accommodates both logged-in and non-logged-in users, ensuring seamless file transfers regardless of user authentication.

The platform's technology stack includes Node.js, Express, Socket.io, JavaScript, Vite, React.js, and Tailwind CSS, chosen for their ability to provide a user-friendly and efficient file-sharing experience. This versatile platform caters to a diverse range of users, whether they're collaborating at work, sharing personal files, or engaging in creative content exchanges. EasySent mission is to make file sharing hassle-free, offering a reliable solution that adapts to various sharing scenarios.

## Problem Statement

In today's digital age, effectively sharing files can be a challenge for both individuals and organizations. Traditional file-sharing methods often involve complex authentication processes, strict file size limits, or unreliable transfer methods, leading to inefficiencies and user frustrations during content exchange. Additionally, the demand for real-time collaboration and instant file transfers is crucial for maintaining smooth workflows and effective communication among users.

To address these common challenges, the EasySent project aims to create a user-centered file-sharing platform. The platform's goal is to simplify the process of sharing files in real-time, offering a solution that is both seamless and accessible. EasySent caters to a wide range of users, including individuals, small businesses, and organizations. It achieves this by integrating Socket.IO for instant communication, utilizing the capabilities of Node.js and Express for a robust backend infrastructure, and employing React with Vite for a responsive frontend. Ultimately, EasySent's objective is to provide a user-friendly platform that meets the diverse file-sharing needs of users across various scenarios. It aspires to empower users with effortless content sharing experiences while eliminating the complexities associated with traditional methods.

## Objective of the Project

* **To Make File Sharing Faster:** EasySent should help people share files more quickly, especially when they need to work together right away.
* **To Keep Things Easy to Use:** We aim to ensure that EasySent is easy to understand and simple to use for everyone.
* **To Let Everyone, Share Files Easily:** EasySent should work for everyone, whether they're logged in or not, making sure that sharing files is easy and safe for all.

## Significance of The Project

* **Enhanced Collaboration and Communication:** Improve collaboration and communication among users through real-time file sharing and dynamic discussions with Socket.IO integration.
* **Simplified File Sharing:** Provide a user-friendly platform for quick file sharing with unique shareable links, eliminating the need for complex authentication.

## Features of the Project

* **Shareable Links:** After successful file uploads, EasySent generates unique shareable links for each file. Users can easily share these links with others, granting direct access to the shared files without requiring complex authentication.
* **Link Expiration:** EasySent automatically sets shared links to expire after 24 hours. This ensures that file access is restricted by default, enhancing security and managing file availability.
* **Efficient File Transfer:** With real-time communication and optimized server-side handling, file transfers on EasySent are efficient, allowing users to send and receive files without unnecessary delays.
* **MongoDB Database for URL Storage:** EasySent includes a MongoDB database that stores URLs when either the sender or receiver is logged in. This feature ensures that shared URLs are securely retained, allowing both parties to conveniently access shared content and providing a reliable record of shared files.

## Scope and Limitation

### 1.7.1 Scope

* **Real-time File Sharing:** Users can share files instantly and engage in real-time communication through Socket.IO integration.
* **User-Friendly Interface:** The platform will have an intuitive and user-friendly interface, making it easy for users of all technical backgrounds to upload and share files.
* **Shareable Links:** Unique shareable links will be generated for each uploaded file, simplifying the sharing process without complex authentication.

### 1.7.2 Limitation

* **File Size Limitations:** Due to server and network constraints, there may be limitations on the size of files that can be uploaded and shared.
* **Security Risks:** While efforts will be made to secure data transmission, no system is entirely immune to security risks. Users are advised to share sensitive files cautiously.
* **User Authentication:** Without login or authentication, direct file sharing may lack user accountability and history tracking for shared files.

# Chapter 2: Literature Overview

## 2.1 WeTransfer

WeTransfer is a popular file sharing website that focuses on transferring large files quickly and effortlessly. Users can upload files up to a certain size (usually up to 2GB for free accounts) and send them to recipients via email. The service provides a simple and straightforward user interface, making it ideal for one-time large file transfers.

**Advantages:**

* **Quick and Simple File Transfer:** WeTransfer specializes in fast and easy large file transfers without the need for sign-ups or installations.
* **No Storage Limitations:** WeTransfer does not store files permanently, freeing up space on the server after the transfer is completed.
* **No Account Required:** Users can send files without creating an account, providing a straightforward and hassle-free experience.

**Disadvantages:**

* **Limited File Retention Period:** Files on WeTransfer are available for download for a limited time, usually up to seven days, after which they are deleted.
* **File Size Limitation:** WeTransfer has size limitations for free users (usually 2GB), and larger files require a paid subscription.
* **Lack of File Management Features:** WeTransfer is designed for one-time transfers and lacks file organization and management features like cloud storage services.

## 2.2 MediaFire

MediaFire offers cloud storage and file sharing services with both free and premium plans. Users can upload, share, and manage files and folders, and MediaFire provides robust privacy and security features. It is commonly used for hosting files for direct download links on websites and forums.

**Advantages:**

* **Free Storage:** MediaFire offers 10GB of free storage, making it suitable for users with moderate storage needs.
* **Direct Download Links:** MediaFire is often used to host files for direct download links on websites and forums.
* **File Versioning:** MediaFire keeps file versions, allowing users to revert to previous versions if needed.

**Disadvantages:**

* **Advertisements and Captchas:** Free users may encounter advertisements and captchas during file downloads.
* **Limited Collaboration Features**: MediaFire is primarily a cloud storage service and lacks advanced collaboration features found in other platforms like Google Drive and Dropbox.
* **File Size Limitations:** MediaFire has file size limitations for free users (usually 2GB), and larger files require a paid subscription.

# Chapter 3: Methodology

## 3.1 Software Development Life Cycle

The Easy Sent project was developed using the Agile Software Development Life Cycle (SDLC) methodology. It involved dividing the project into smaller iterations, allowing for frequent feedback, customer collaboration, and adaptability to changing requirements. Regular meetings and user testing sessions ensured alignment with project goals, resulting in a user-centric and feature-rich file-sharing platform that effectively met user needs. The Agile SDLC's flexibility and responsiveness will be key factors in delivering a successful and user-friendly application.

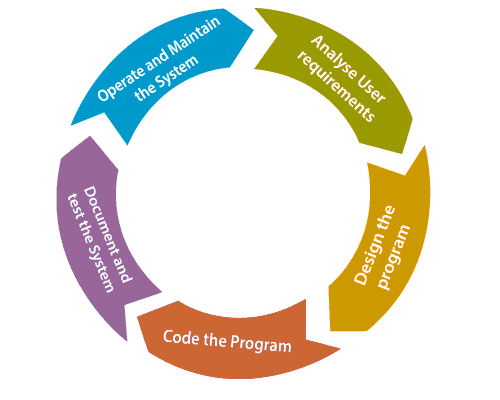


Figure 3.1 Agile Software Development Life Cycle

## 3.2 Tool and Technology Used

* **Node.js:** A JavaScript runtime used for building the backend server and handling server-side operations.
* **Express:** A fast and minimalist web application framework for Node.js, used to create the backend server and handle HTTP requests.
* **Socket.IO:** A JavaScript library enabling real-time bidirectional communication between the server and clients, utilized for instant file transfers and dynamic discussions.
* **React with Vite:** React is a JavaScript library used for building user interfaces, and Vite is a fast build tool for modern web applications. Together, they were used to develop the frontend of the platform.
* **Tailwind CSS:** A utility-first CSS framework providing pre-built classes for rapid styling, used for designing and styling the user interface.
* **MongoDB:** A NoSQL database used for storing data, including shared URLs and other relevant information in EasySent's backend.

## 3.3 Integrated Development Environment

The Integrated Development Environment (IDE) used to develop the EasySent project was Visual Studio Code (VS Code), a popular and lightweight code editor widely used by developers.

VS Code provided an extensive set of tools and extensions for web development, making it well-suited for working with Node.js, React, and other technologies used in the project. In addition to VS Code, the development team adopted Git for version control, utilizing platforms like GitHub to host the Git repositories. Git enabled effective collaboration among team members, allowing them to track changes, manage code versions, and address any issues that arose during development.

The integration between VS Code and Git platforms like GitHub facilitated efficient code management and streamlined collaboration, enabling the team to work cohesively and deliver a robust and responsive file-sharing platform. The combined use of these tools provided a powerful development environment, ensuring effective project management and the successful delivery of the EasySent application.

# Chapter 4: System Analysis

## 4.2 Requirement Analysis

### 4.2.1 Functional requirement

|  |  |  |
| --- | --- | --- |
| **NO** | **Requirement** | **Function** |
| R1 | File Upload and Sharing | * Users can upload files of various formats to the platform. * The platform generates unique shareable links for each uploaded file. * Users can share the links with others to allow them to download the files directly. |
| R2 | File Size Limitations | * The platform sets limitations on file sizes for efficient handling and data storage. |
| R3 | User Registration and Login | * Users can register with the platform using email. * User should be able to login using Registered Email. |
| R4 | Add File or Remove File | * User should be able to add file or Remove file form Generated uniqueID. |
| R5 | Share File and Receive files without Login | * User should be able to send and receive files without login. |
| R6 | Send and Received Logs | * If user is Logged in the system should automatically stored send and Received URLs. |
| R8 | QR | * When URL with UniqueID is generated, the system should display QR code. |
| R9 | Expire | * The generated UniqueID should automatically expires in 24hr. |
| R10 | Delete Sent Unique ID | * When sender/user is logged in Sender should be able to delete generated UniqueID and remove URL from the database and file form the server. |

### 4.2.2 Non-Functional Requirement

|  |  |  |
| --- | --- | --- |
| **No** | **Requirement** | **Function** |
| R1 | Performance | * The platform should have low latency and quick response times during file uploads, downloads, and real-time communication. * It should be capable of handling multiple concurrent users without compromising performance. |
| R2 | usability | * The user interface should be intuitive, user-friendly, and accessible to users with varying technical backgrounds. * File sharing and management tasks should be straightforward and easy to perform. |
| R3 | Reliability | * The platform should be reliable and available for users at all times, with minimal downtime for maintenance and updates. |
| R4 | Compatibility | * The platform should be compatible with modern web browsers and devices to ensure a seamless user experience across different environments. |

## 4.3 Feasibility Study

### 4.3.1 Technical feasibility

In this type of feasibility study, the technical aspects of the project were analyzed. The various technical aspects such as hardware and software were taken into consideration during the development of this project.

### 4.3.2 Schedule feasibility

Schedule feasibility was done to know whether the project could be completed before the deadline or not. This feasibility study was used to allocate the time for separate module development in the system.

### 4.3.3 Economic feasibility

Since we don’t have to purchase any license for the software and all the software which are used to develop this website is opensource code software and this project is developed to meet our academic project, therefore there is no any cost.

## 4.4 Gantt Chart

# Chapter 5: SYSTEM DESIGN

## 5.1 Context Level Diagram

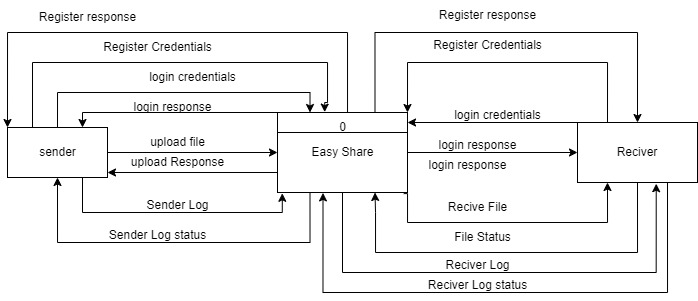


Fig 5.2 Context Level Diagram

## 5.2 DFD level 1

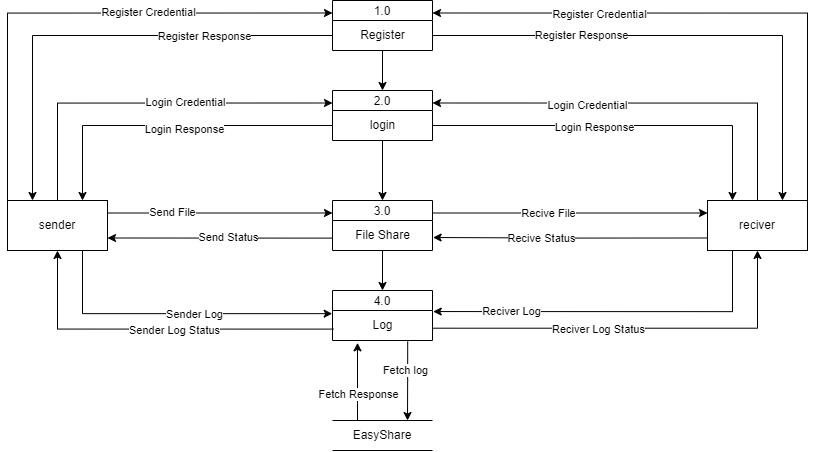


Fig 5.3 DFD level 1

## 5.3 Use Case Diagram

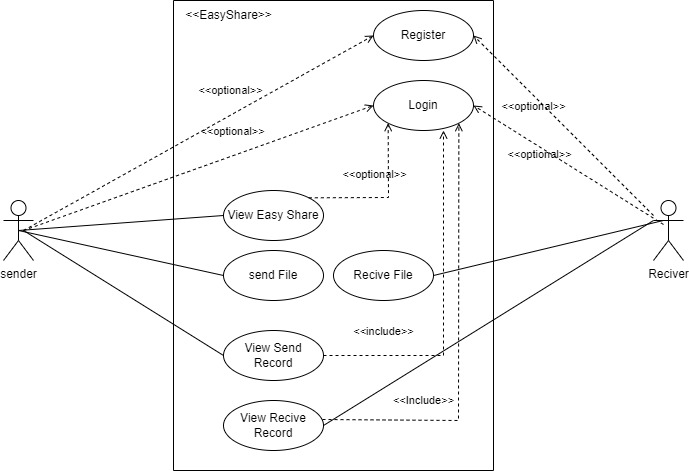
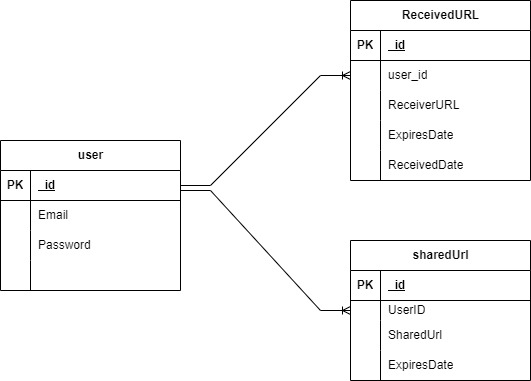


Fig 5.4 Use Case Diagram

## 5.4 Relational Diagram



## 5.5 Sender Flow Chart

### 5.5.1 Sender Flow Chart

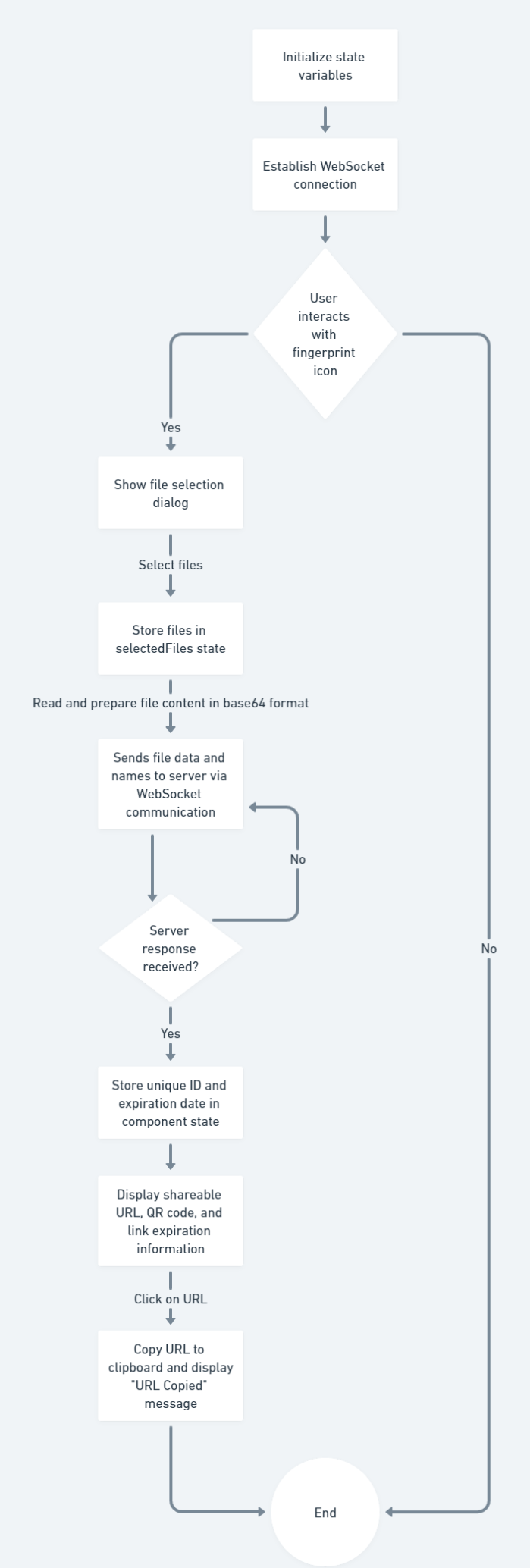


Fig 5.5.1 Sender Flow Chart

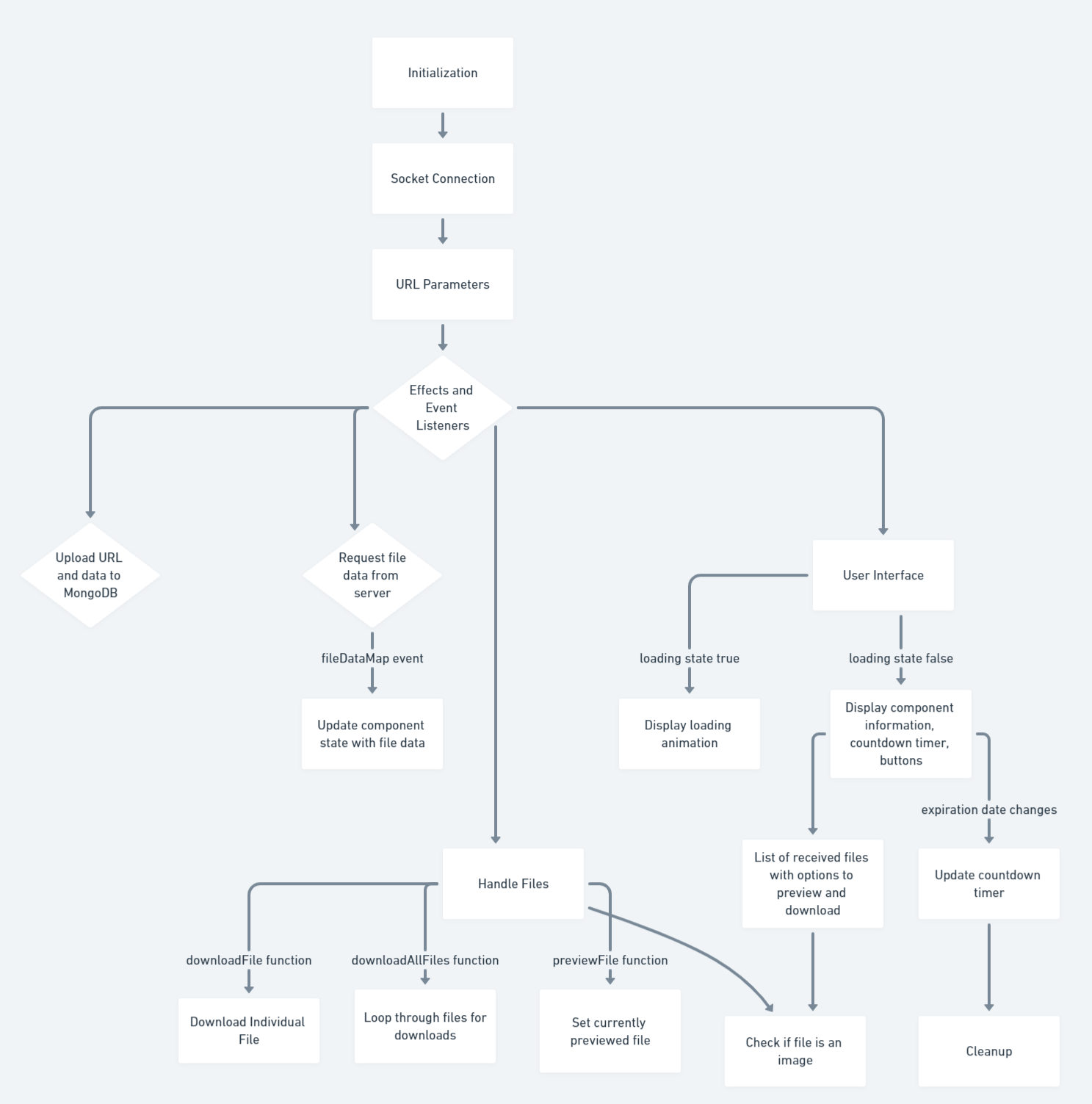
5.5.1 Receiver Flowchart

Fig 5.5.2 Receiver Flow Chart

### 5.5.2 User Profile Flowchart

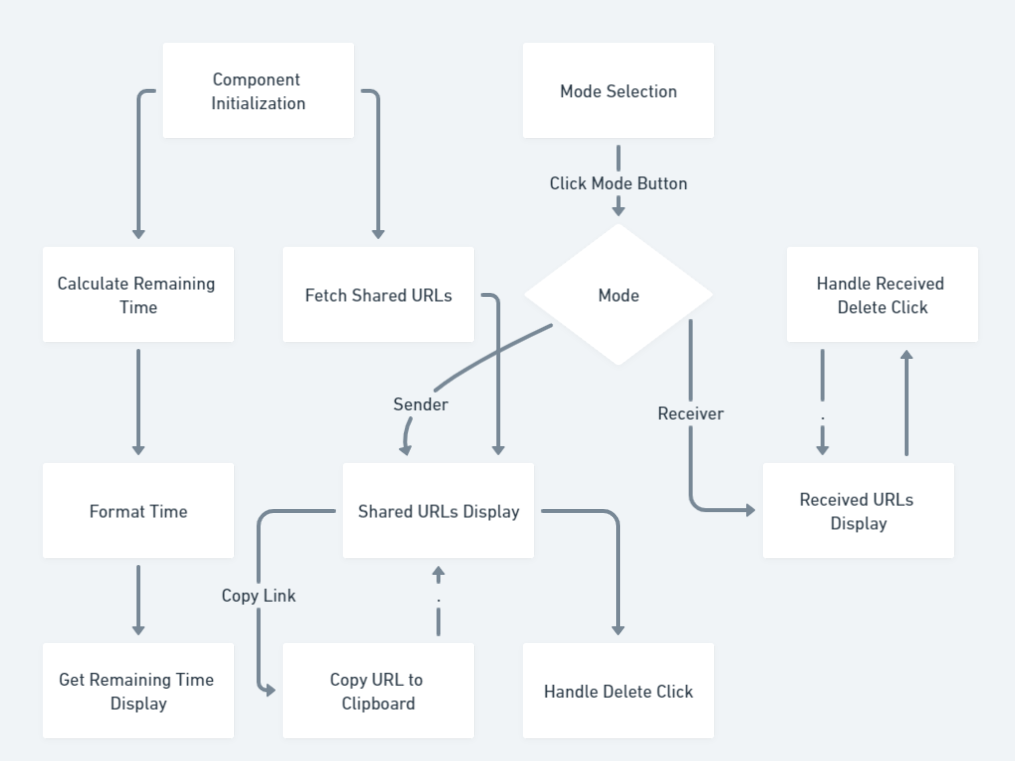


Fig 5.5.3 User Profile Flow Chart

# CHAPTER 6: SYSTEM DEVELOPMENT AND IMPLEMENTATION

## 6.1 Programming Platform

The EasySent application was developed using a combination of programming platforms and technologies. Here's a breakdown of the key elements used in EasySent's development:

1. **Backend Server:**

* **Node.js:** Used as the primary runtime environment for the backend server. Node.js allows for efficient server-side JavaScript development.
* **Express:** A fast and minimalist web application framework for Node.js. Express is used to create the backend server and handle HTTP requests.

1. **Real-Time Communication:**

* **Socket.IO:** A JavaScript library for enabling real-time bidirectional communication between the server and clients. Socket.IO is utilized for instant file transfers and dynamic discussions.

1. **Frontend Development:**

* **React with Vite:** React is a popular JavaScript library for building user interfaces. Vite is a fast build tool for modern web applications. Together, they were used to develop the frontend of the EasySent application. React provides a responsive and dynamic user interface.

1. **User Interface Styling:**

* **Tailwind CSS:** A utility-first CSS framework that provides pre-built classes for rapid styling. Tailwind CSS is used for designing and styling the user interface, ensuring efficiency in the styling process.

1. **Database:**

* **MongoDB:** A NoSQL database used for data storage in EasySent. It stores user information, shared URLs, and received URLs. MongoDB is known for its flexibility and scalability, making it suitable for various types of data.

1. **Development Tools:**

* **Visual Studio Code (VS Code):** The integrated development environment (IDE) used for coding and managing the EasySent project. VS Code is known for its lightweight and extensible nature, making it popular among developers.

1. **Version Control:**

* **GitHub:** A web-based platform for hosting Git repositories. GitHub facilitates collaboration among team members by providing tools for code management and issue tracking.

These programming platforms and technologies were carefully selected to create EasySent, a user-friendly and efficient file-sharing platform that streamlines the process of sharing files in real-time. They provide the necessary tools and capabilities to handle various aspects of the application, from backend logic to frontend presentation and real-time communication.

## 6.2 Test Plan

Since there are lots of interconnected components the team thoroughly tested the site to ensure that there is no problem. In additional to that website was tested on different size device as well as to ensure reactive nature of the website.

## 6.3 Features to be Tested

* Login
* Register
* Sent and receive File
* Add and remove File form generated UniqueID (URL)
* Logs
* Delete generated UniqueID

## 6.4 Testing Tool

Testing was done manually to ensure there was proper co-ordination between the team members and was done by using GitHub. GitHub allowed for proper communication and provided with the progress of fellow team member to be noted, and also allowed for proper integration of the software.

## 6.5 Test cases

### 6.5.1 Login

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| LOG\_1 | User Login | User Input wrong username/ password | Display Invalid Email or Password. |
| LOG\_2 | User Login | User Input correct username and password | Display Login Successful and redirect to main page. |

Table 6.5.1 Login

### 6.5.1 Register

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| REG\_1 | User Register | User Input short password or missed to add number | Display Must contain at least 6 character and a number |
| REG\_2 | User Register | User Input password and confirm password doesn’t match | Display password do not match |
| REG\_3 | User Register | User Input correct necessary fields. | Display Sign up Success and redirect to login page. |

Table 6.5.1 Register

### 6.5.1 Send and Receive File

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| S&R\_1 | Send | When user finish selecting file. | Generate uniqueID URL. With QR code. |
| S&R\_2 | Receive | When user use Generated uniqueID URL or Scan QR. | Redirect to Receiver page when Receiver receives Files. |

Table 6.5.1 Send and Receive Files

### 6.5.1 Add or Remove Files.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| AR\_1 | Add | When Sender adds file in already generated URL. | Add file in same uniqueID. |
| AR\_2 | Remove | When senders remove file from the generated uniqueID. | Remove File form the generated uniqueID. |

Table 6.5.1 Add or Remove Files

### 6.5.1 Logs

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| LG\_1 | Sender | If the sender logged in when the uniqueID URL is generated automatically insert Sender URL in database. | Insert in Sent File which contain URL and sender ID. |
| LG\_2 | Receiver | When the Receiver is logged in and when receiver paste URL automatically insert in to received files database. | Insert in to received file database. |
| LG\_3 | Receiver | When Receiver use same URL multiple time. | Do not insert in to the database. |

Table 6.5.1 Logs

### 6.5.1 Delete Generated UniqueID

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **TEST CASE** | **USER INPUT** | **PASS CRITERIA** |
| DG\_1 | Delete | When the sender is logged in and click delete URL | Delete URL form the database and remove file form the server. |

Table 6.5.1 Delete Generated UniqueID

# Chapter 7 conclusion and Future Enhancement

## 7.1 Conclusion

EasySent is a user-friendly file-sharing platform designed for quick and real-time file transfers. It offers a straightforward approach where users can select a file and generate a unique ID URL for sharing purposes. Whether the sender and receiver are logged in or not, EasySent allows for efficient file transfers. If both parties are logged in, the shared URL and received file are automatically stored in MongoDB, providing convenient access to shared content. This platform was developed using a stack of technologies, including Node.js, Express, Socket.IO, JavaScript, React with Vite, and Tailwind CSS.

**7.2 Future Enhancement**

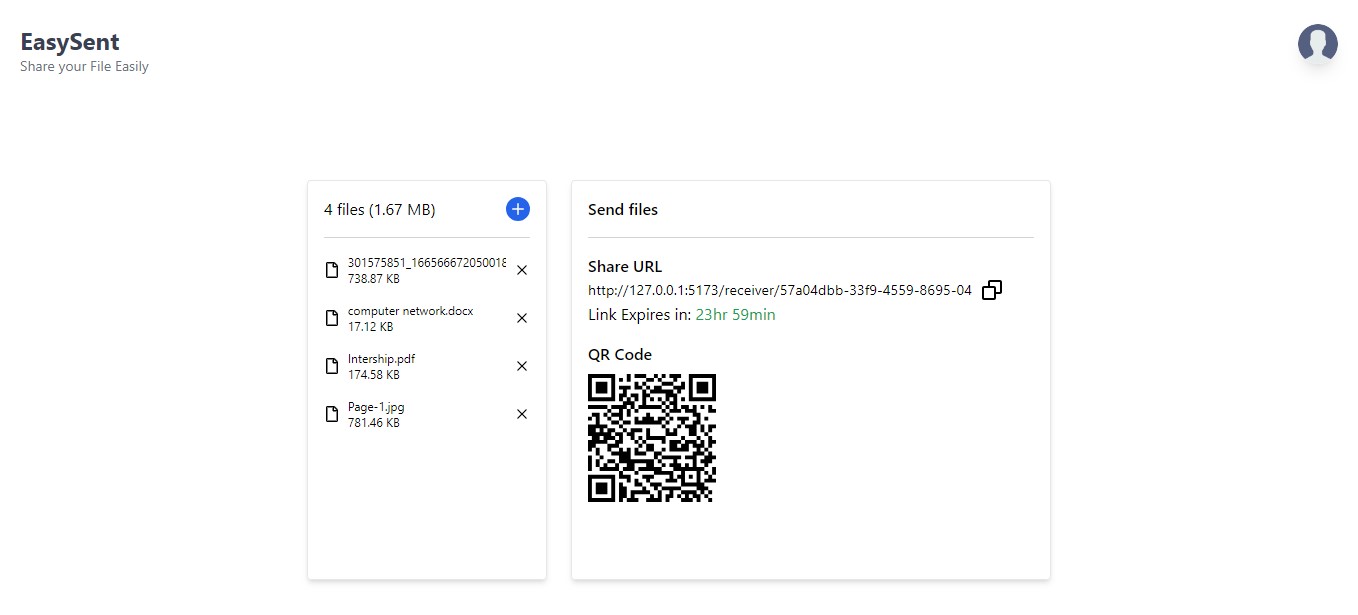
* Improve EasySent to allow users to send large-sized files seamlessly.
* Introduce the ability to create room where multiple users can send and receive files.
* Enable user to set an expiration time for the file they share through EasySent.

**REFRENCES**

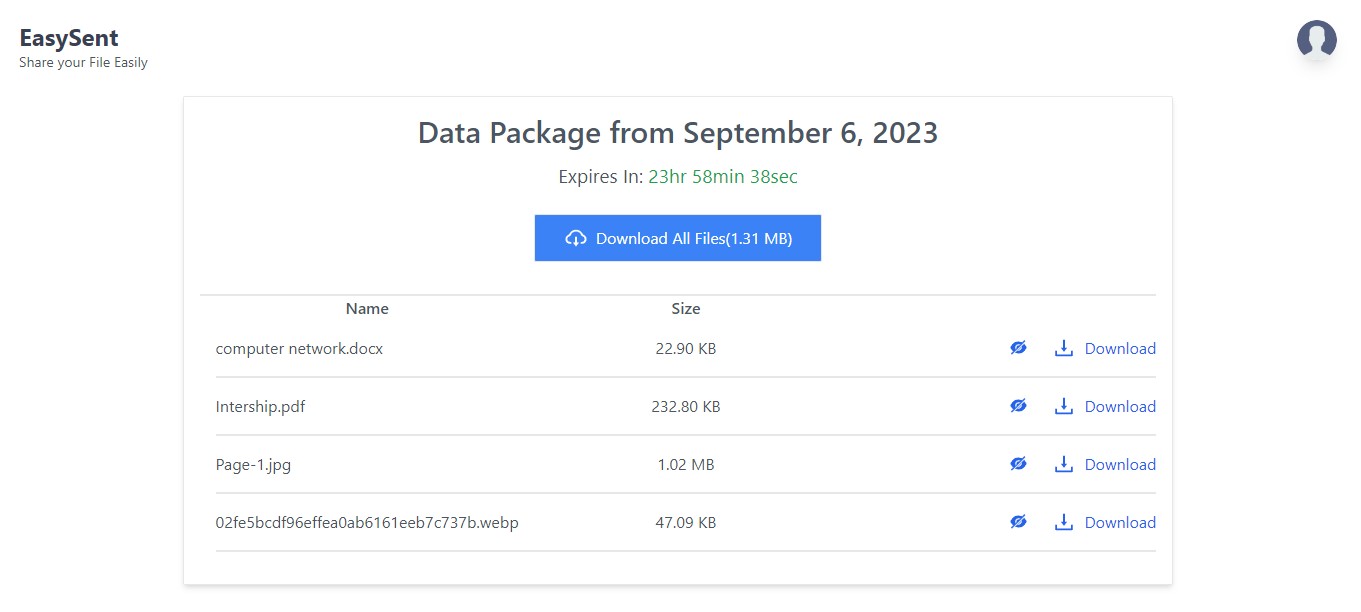
* Duckett, J. (2011). HTML & CSS: Design and build websites. John Wiley & Sons.
* Express. (n.d.). Express 5.x -API Reference. Retrieved form https://expressjs.com/en/5x/api.html

**Appendices**

**Sender Page**

****

**Receiver**

****