

filez++ (SSH File Manager)

RAHUL R - LKTE18MCA065

Department of Computer Applications
Rajiv Gandhi Institute of Technology, Kottayam

Guided By :
Prof. Jane George
Asst. Professor
Dept. of MCA

Introduction

- As remote working is going on companies have to use servers remotely.
- A Utility Tool for Browsing files from a Remote server using protocols like Secure Shell Host (SSH).
- The Software is an installable, Desktop Application developed using Electron.JS (Packaging Tool) by using Web Technologies including Node.JS, Express.JS, Angular, EJS.

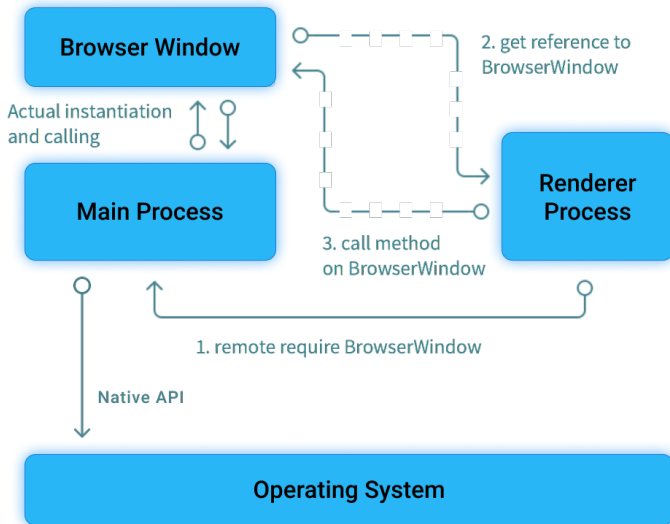
Existing System

- Cannot save connections
- Cannot connect to multiple ssh servers at once
- User Experience Issue, hard to use for naive users
- Server-side installation required
- Not Portable (User have to install the software in their PC)

Proposed System

- Add/Edit/Test/Delete Connections
- Connect to multiple SSH servers at once.
- GUI is easy to use, and highly customizable
- Most features of Native File Browsers like Windows File Explorer.
- Logs and User activity available.
- Server-side installation is not required
- Portable (Can run without installation on Users PC)
- Can run on Windows, Ubuntu, Fedora, Debian, MAC, etc.
- Logs and User activity available.
- Does not take much resources like RAM and CPU.

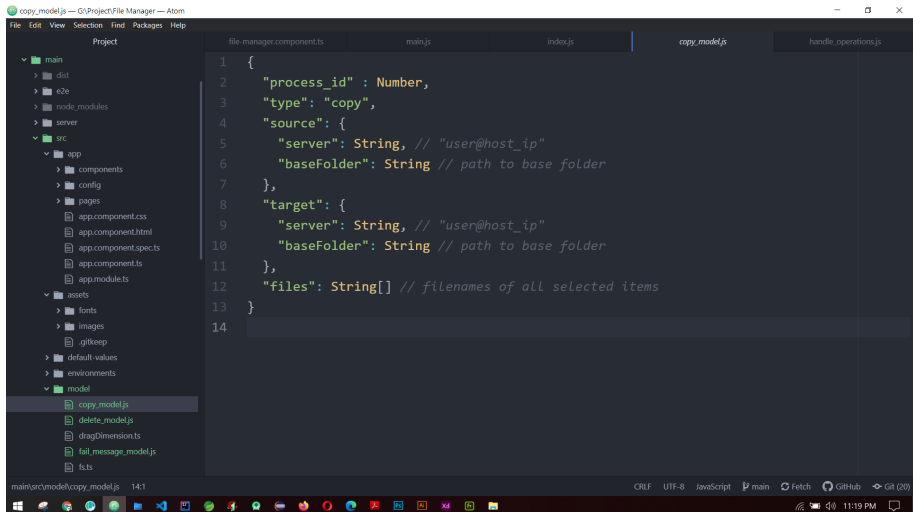
System Architecture



Product Backlog

SI No.	User Stories	Priority	Estimated Time
1	User can browse local files	2	4 days
2	User can create new file	2	2 days
3	User can create new folder	2	1 day
4	User can delete files and folders	2	7 days
5	User can rename files and folders	2	5 days
6	User can open files, folders and shortcut files	2	4 days
7	User can add, edit, delete, test ssh connections	1	13 days
8	User can browse SSH files	1	2 days
9	User can create new file/folder	1	1 day
10	User can delete/rename files/folders	1	5 days
11	User can open folders	1	3 days
12	User can copy/paste files/folders	1	7 days
13	User can cut/paste files/folders	1	1 day

Data Model



The screenshot shows the Atom editor interface. The left sidebar displays a project tree with the following structure:

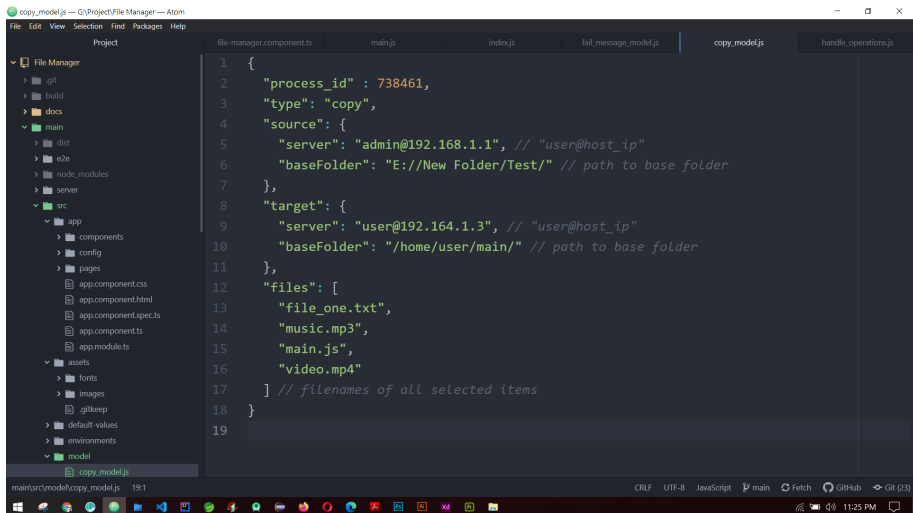
- main
 - dist
 - e2e
 - node_modules
 - server
 - src
 - app
 - components
 - config
 - pages
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - fonts
 - images
 - .gitkeep
 - default-values
 - environments
 - model
 - copy_model.js (selected)
 - delete_model.js
 - dragDimension.ts
 - fail_message_model.js
 - fs.ts

The main editor area shows the content of `copy_model.js`:

```
1 {
2   "process_id" : Number,
3   "type": "copy",
4   "source": {
5     "server": String, // "user@host_ip"
6     "baseFolder": String // path to base folder
7   },
8   "target": {
9     "server": String, // "user@host_ip"
10    "baseFolder": String // path to base folder
11  },
12  "files": String[] // filenames of all selected items
13 }
14
```

The status bar at the bottom indicates the file path `main/src/model/copy_model.js` at line 14, column 1. It also shows encoding settings (CRLF, UTF-8), language (JavaScript), and other utility icons.

Data Model

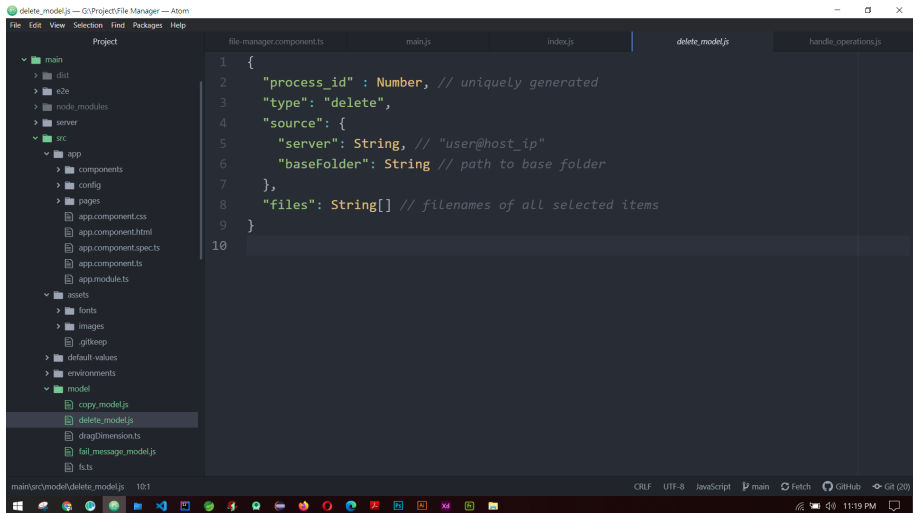


The screenshot shows the Atom editor interface. The title bar reads "copy_model.js — G:\Project\File Manager — Atom". The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The Project sidebar on the left shows a file tree with folders like .git, build, docs, main, dist, e2e, node_modules, server, and src. The src folder is expanded, showing subfolders like app, assets, fonts, images, and model. The model folder contains the file copy_model.js, which is selected. The main editor pane shows the content of copy_model.js, which is a JSON object with fields for process_id, type, source, target, and files. The source and target fields are commented with user@host_ip and baseFolder paths. The files field is an array of filenames.

```
1 {
2   "process_id" : 738461,
3   "type": "copy",
4   "source": {
5     "server": "admin@192.168.1.1", // "user@host_ip"
6     "baseFolder": "E://New Folder/Test/" // path to base folder
7   },
8   "target": {
9     "server": "user@192.164.1.3", // "user@host_ip"
10    "baseFolder": "/home/user/main/" // path to base folder
11  },
12  "files": [
13    "file_one.txt",
14    "music.mp3",
15    "main.js",
16    "video.mp4"
17  ] // filenames of all selected items
18 }
19
```

The status bar at the bottom shows the file path "main\src\model\copy_model.js" and the time "19:1". The right side of the status bar includes tabs for CRLF, UTF-8, JavaScript, and icons for main, Fetch, GitHub, and Git (23).

Data Model



The screenshot shows the Atom code editor interface. The left sidebar displays a file tree for a project named 'delete_model.js'. The tree structure is as follows:

- main
 - dist
 - e2e
 - node_modules
 - server
 - src
 - app
 - components
 - config
 - pages
 - app.component.css
 - app.component.html
 - app.component.spec.ts
 - app.component.ts
 - app.module.ts
 - assets
 - fonts
 - images
 - .gitkeep
 - default-values
 - environments
 - model
 - copy_model.js
 - delete_model.js (selected)
 - dragDimension.ts
 - fail_message_model.js
 - fs.ts

The main editor area shows the content of 'delete_model.js' with the following code:

```
1 {  
2   "process_id" : Number, // uniquely generated  
3   "type": "delete",  
4   "source": {  
5     "server": String, // "user@host_ip"  
6     "baseFolder": String // path to base folder  
7   },  
8   "files": String[] // filenames of all selected items  
9 }  
10
```

The status bar at the bottom indicates the file path 'main/src/model/delete_model.js' and the time '10:1'. The bottom of the screen shows a Windows taskbar with various application icons and the system clock '11:19 PM'.

Data Model

The screenshot shows the Atom editor interface. The title bar reads "fail_message_model.js — G:\Project\File Manager — Atom". The menu bar includes "File", "Edit", "View", "Selection", "Find", "Packages", and "Help". The left sidebar shows a project tree with folders like "main", "dist", "e2e", "node_modules", "server", "src", "assets", "default-values", "environments", and "model". The "model" folder is expanded, showing files like "copy_model.js", "delete_model.js", "dragDimension.ts", "fail_message_model.js" (selected), and "fs.ts". The main editor area has tabs for "file-manager.component.ts", "main.js", "index.js", "fail_message_model.js" (active), and "handle_operations.js". The active tab displays a TypeScript data model:

```
1 {  
2   "process_id" : Number,  
3   "status" : "failed",  
4   "error_message" : String, // for user to see  
5   "dev_log" : String // for developer  
6 }  
7
```

The status bar at the bottom shows the file path "main/src/model/fail_message_model.js" with a zoom level of "7:1". It also displays encoding "CRLF", "UTF-8", language "JavaScript", and various icons for "main", "Fetch", "GitHub", and "Git (20)". The Windows taskbar is visible at the very bottom.

Hardware Requirements

- Storage : 350 to 400 MB (or greater) free space
- RAM : 1 GB (or more), 2GB Recommended
- Mouse : Required
- Keyboard : Not mandatory if having Touch Screen, or on screen keyboard

Sprint Backlog

SI No.	Sprint	To Do	Status
1	Sprint 1	Analyze Features of Existing System	Completed
2		Develop UI Wireframe and Prototype	Completed
3		UI basic backbone development	Completed
4		Create Data Models	Completed
5	Sprint 2	SSH Connection establishing	Completed
6		SSH Connection testing	Completed
7		local file System accessing	Completed
8	Sprint 3	create files feature	Completed
9		create folders feature	Completed
10		delete files and folders feature	Completed
11		browse directoy contents feature	Completed
12	Sprint 4	GUI file management feature	Completed
13		Serach feature	Completed
14		Open files feature	Completed
15	Sprint 5	Rename feature	Completed
16		Packaging for Multiple Operating Systems	Completed
17	Sprint 6	Testing in multiple Operating Systems	Completed
18		File Copy-Paste Feature	Completed
19		File Cut-Paste Feature	Completed
20		Got To directory path feature	Completed

Version Control (Git)

github.com/27px/Remote-File-Manager

27px / Remote-File-Manager

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 1 tag

Go to file Add file Code

27px Added feature to open shortcut file in local system 59429b8 15 days ago 73 commits

docs	fixed edit path issue and added open file in system app feature	25 days ago
main	Added feature to open shortcut file in local system	15 days ago
.gitignore	implemented copy with skip and overwrite options	29 days ago
LICENSE	componentized files and folders	4 months ago
README.md	design changes	27 days ago
desktop.ini	renamed client->main, and moved server in to client(main)	2 months ago
todo.txt	Added feature to open shortcut file in local system	15 days ago

File_Manager

About

Server Based GUI File Manager

Readme

AGPL-3.0 License

Releases

1 tags

Create a new release

Packages

No packages published


Publish your first package

11:02 PM

Screenshot

File Manager Refresh

Add Connection +

My Computer 

Local
Local Server Accessing via SSH
localhost
Admin

RIT Server
RIT College Server
14.139.188.87
raahul

Demo Server
Some Open Server from Internet
test.rebex.net
demo

/

Search

C: 70%
105 GB used of 149 GB

D: 86%
172 GB used of 199 GB

E: 71%
249 GB used of 349 GB

F: 3%
2 GB used of 80 GB

G: 67%
91 GB used of 135 GB

5 items 0 items selected

Windows taskbar: 10:03 PM

Conclusion

- Developed a user friendly software inspired by FileZilla, without some of its limitations and adding some more Features.
- The Software is a Desktop application that can run on any platform and architecture.
- packaged app can be distributed and can be run without any additional dependencies or programming languages installed and is portable.
- Future modifications and scope could be adding support for System default recycle bin, including Online Storages like Google Drive, Dropbox etc.

