**CPU Scheduling Algorithms**

**Technologies Used :**

1. HTML
2. CSS
3. JavaScript
4. Google Charts : <https://developers.google.com/chart> – for charts
5. ChartJS : <https://www.chartjs.org/> - for charts
6. ElectronJS : <https://www.electronjs.org/> - for building desktop app.

**ElectronJS**

Using ElectronJS, we can make desktop application for Windows, Mac or Linux using Web Technologies like HTML, CSS and JavaScript.

It renders the HTML page to make the application.

It means “if we can build a website then we can build a desktop app”.

**Procedure :**

1. First make the website with HTML, CSS and JavaScript.
2. Install NodeJS in the system from official website : <https://nodejs.org/> (for Windows) or using terminal (for Mac or Linux) :

sudo apt install nodejs

1. Check the status of installation of nodejs using terminal :

node -v

1. Install npm using terminal :

sudo apt install npm

1. Check the status of installation of npm using terminal :

npm -v

1. Make package.json file using terminal :

npm init

1. Enter the details of the package as asked in the terminal.
2. Install ElectronJS in the directory where the HTML, CSS and JavaScript files are present using terminal :

npm install --save-dev electron

1. Check the status of installation of electron by checking electron entry in devDependencies in package.json file.
2. Remove the test script from package.json.
3. Add this script to the package.json.

"start": "electron ."

1. Make main.js file in the same directory containing the code for window and for rendering the HTML code to the window and contains code for taskbar for window.
2. Check the status of app using terminal :

npm start

1. The application building is done.
2. Add icons for the app in the same directory.
3. Make a folder with name assets.
4. Make a subfolder with name icons.
5. Make 3 subfolders with names mac, win and png (for Mac, Windows and Linux apps respectively).
6. Rename the icon to icon.png, icon.ico and icon.icns and put it in the respective folders.
7. Now to build the package, first add this scripts to the package.json file :

"package-mac": "electron-packager . --overwrite --platform=darwin --arch=x64 --icon=assets/icons/mac/icon.icns --prune=true --out=release-builds",

"package-win": "electron-packager . scheduling-algorithms --overwrite --asar=true --platform=win32 --arch=ia32 --icon=assets/icons/win/icon.ico --prune=true --out=release-builds --version-string.CompanyName=CE --version-string.FileDescription=CE --version-string.ProductName=\"CPU Scheduling Algorithms\"",

“package-linux": "electron-packager . scheduling-algorithms --overwrite --asar=true --platform=linux --arch=x64 --icon=assets/icons/png/icon.png --prune=true –out=release-builds"

1. Change the name of the file and app and icons in the scripts.
2. Install electron-packager using terminal to package the electron app :

npm install -g electron-packager

1. Package the app using terminal :

npm run package-win (for Windows)

npm run package-mac (for Mac)

npm run package-linux (for Linux)

1. A release-builds folder will be created in the same directory.
2. The application has been created.
3. It can be launched from the app file in release-builds folder.

(For Linux only)

1. For creating a .deb file to install the application, make a debian.json file in the same directory (can be found in the code files).
2. Run the command in the terminal :

electron-installer-debian --src release-builds/scheduling-algorithms-linux-x64/ --arch amd64 --config debian.json

With user specific name.

1. The .deb file has been created in the release-builds folder.
2. The application can be installed using the file.

After the application has been made, it can be distributed as required.

The website repository is at <https://github.com/vasu-gondaliya/cpu-scheduling-algorithms>

The website is hosted at <https://vasu-gondaliya.github.io/cpu-scheduling-algorithms/>

**Further Enhancements:**

1. Feature to save the input for future use and upload saved input.
2. Feature to download the output.
3. Design can be enhanced.
4. More visualizations can be added.
5. Mobile View of the website can be made.