Hope you subscribe. Thanks

---- flask_server.py ---- in flask_app folder

```
import sys

import os

from flask import Flask, render_template



if getattr(sys, 'frozen', False):

    template_folder = os.path.join(sys._MEIPASS, 'templates')

    print(template_folder)

    app = Flask(__name__, template_folder=template_folder)

else:

    app = Flask(__name__)


@app.route("/")

def hello():

    return render_template("main.html")


if __name__ == "__main__":

    app.run(host='127.0.0.1', port=5000)
```

---- main.html --- in flask_app/templates folder

```
<!DOCTYPE html>

<html lang="en">

<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h2>Hello welcome to my flask application</h2>
</body>
</html>
```

------ main.js --------- in electron folder

```
// Modules to control application life and create native browser window
const {app, BrowserWindow} = require('electron')
const path = require('path')

function createWindow () {

  var executablePath = 'flask_server.exe';
  var child = require('child_process').exec;

  child(executablePath, function (err, data) {
    if (err) {
      console.error(err);
      return;
    }
    console.log(data.toString());
  });
```

```
// Create the browser window.
const mainWindow = new BrowserWindow({
  width: 800,
  height: 600,
  webPreferences: {
    preload: path.join(__dirname, 'preload.js')
  }
})

const urlExist = require("url-exist");

(async () => {
    const exists = await urlExist("http://127.0.0.1:5000/");
    // Handle result
    mainWindow.loadURL('http://127.0.0.1:5000/')
})();



// and load the index.html of the app.
mainWindow.loadFile('index.html')

// Open the DevTools.
// mainWindow.webContents.openDevTools()

mainWindow.on('closed', function(){

  // kill the server on exit

  const { exec } = require('child_process');
```

```
    exec('taskkill /f /t /im flask_server.exe', (err, stdout, stderr) => {
      if (err) {
        console.log(err)
        return;
      }

      // the *entire* stdout and stderr (buffered)
      console.log(`stdout: ${stdout}`);
      console.log(`stderr: ${stderr}`);
    });

  });

}

// This method will be called when Electron has finished
// initialization and is ready to create browser windows.
// Some APIs can only be used after this event occurs.
app.whenReady().then(createWindow)

// Quit when all windows are closed.
app.on('window-all-closed', function () {
  // On macOS it is common for applications and their menu bar
  // to stay active until the user quits explicitly with Cmd + Q
  if (process.platform !== 'darwin') app.quit()
})

app.on('activate', function () {
  // On macOS it's common to re-create a window in the app when the
  // dock icon is clicked and there are no other windows open.
  if (BrowserWindow.getAllWindows().length === 0) createWindow()
})

// In this file you can include the rest of your app's specific main process
// code. You can also put them in separate files and require them here.
```

----- package.json ---- in electron folder

```
{
  "name": "electron-quick-start",
  "version": "1.0.0",
  "description": "A minimal Electron application",
  "main": "main.js",
  "scripts": {
    "start": "electron ."
  },
  "repository": "https://github.com/electron/electron-quick-start",
  "keywords": [
    "Electron",
    "quick",
    "start",
    "tutorial",
    "demo"
  ],
  "author": "GitHub",
  "license": "CC0-1.0",
  "devDependencies": {
    "electron": "^8.2.1",
    "electron-builder": "^22.4.1"
  },
  "dependencies": {
    "url-exist": "^2.0.1"
  }
}

// Hope you subscribe. Thanks
```