npm install --save-dev electron

In pakage.json

    "scripts": {

    …rest of the code

        "electron": "electron electron/main.cjs",

    },

Create a folder electron/main.cjs and paste this

const { app, BrowserWindow } = require("electron");

const path = require("path");

const { spawn } = require("child\_process");

let processes = [];

function runArtisan(command) {

  const proc = spawn("php", ["artisan", ...command], {

    cwd: path.join(\_\_dirname, ".."), // go to Laravel root

    shell: true,

  });

  proc.stdout.on("data", (data) => {

    console.log(`[${command.join(" ")}]: ${data}`);

  });

  proc.stderr.on("data", (data) => {

    console.error(`[${command.join(" ")} ERROR]: ${data}`);

  });

  processes.push(proc);

  return proc;

}

function createWindow() {

  const win = new BrowserWindow({

    width: 1200,

    height: 800,

    webPreferences: {

      nodeIntegration: false,

    },

  });

  // Start Laravel backend + workers

  runArtisan(["serve"]);         // Web server

  runArtisan(["queue:work"]);    // Queue worker

  runArtisan(["reverb:start"]);  // Reverb server

  // Load Laravel app into Electron window

  win.loadURL("http://127.0.0.1:8000");

  win.on("closed", () => {

    // Kill all background processes when Electron closes

    processes.forEach((p) => p.kill());

  });

}

app.whenReady().then(createWindow);

app.on("window-all-closed", () => {

  if (process.platform !== "darwin") {

    processes.forEach((p) => p.kill());

    app.quit();

  }

});

Now to run:

**npm run electron**

**To build app**

{

    "name": "attendance-system",

    "version": "1.0.0",

    "description": "Attendance System desktop app with Laravel, React, and Electron",

"author": "Your Name",

    "main": "electron/main.cjs",

    "private": true,

    "type": "module",

    "scripts": {

    ….rest of code

        "electron": "electron electron/main.cjs"

    },

    "devDependencies": {

    ……rest of code

    },

"dependencies": {

….…rest of code

    },

    "optionalDependencies": {

    …rest of code

    },

    "build": {

        "appId": "com.attendance.desktop",

        "productName": "Attendance System",

        "directories": {

            "output": "dist"

        },

        "files": [

             "\*\*/\*",

            "!node\_modules/@esbuild/\*-ppc64\*",

            "!node\_modules/@esbuild/\*-arm\*",

            "!node\_modules/@esbuild/\*-s390x\*",

            "!node\_modules/@esbuild/\*-mips\*",

            "!node\_modules/.cache/\*\*",

            "!vendor/\*\*",

            "!node\_modules/.cache/\*\*"

        ],

        "win": {

            "target": "nsis",

            "icon": "electron/icon.ico"

        }

    }

}

**For with splash screen:**

const { app, BrowserWindow } = require("electron");

const path = require("path");

const { spawn } = require("child\_process");

const http = require("http");

let processes = [];

let splashWindow;

let mainWindow;

function runArtisan(command) {

  const proc = spawn("php", ["artisan", ...command], {

    cwd: path.join(\_\_dirname, ".."), // Laravel root

    shell: true,

  });

  proc.stdout.on("data", (data) => {

    console.log(`[${command.join(" ")}]: ${data}`);

  });

  proc.stderr.on("data", (data) => {

    console.error(`[${command.join(" ")} ERROR]: ${data}`);

  });

  processes.push(proc);

  return proc;

}

// Function to check if Laravel server is running

function checkLaravelReady(url, callback) {

  const tryConnect = () => {

    console.log("Checking Laravel server...");

    http

      .get(url, (res) => {

        console.log("Laravel status:", res.statusCode);

        if (res.statusCode === 200) {

          callback(true);

        } else {

          setTimeout(tryConnect, 1000);

        }

      })

      .on("error", () => {

        console.log("Laravel not ready yet...");

        setTimeout(tryConnect, 1000);

      });

  };

  tryConnect();

}

function createSplash() {

  splashWindow = new BrowserWindow({

    width: 400,

    height: 300,

    frame: false,

    transparent: true,

    alwaysOnTop: true,

  });

  splashWindow.loadFile(path.join(\_\_dirname, "splash.html"));

}

function createMainWindow() {

  console.log("Opening main window...");

  mainWindow = new BrowserWindow({

    width: 1200,

    height: 800,

    webPreferences: {

      nodeIntegration: false,

    },

  });

  mainWindow.loadURL("http://127.0.0.1:8000");

  mainWindow.on("closed", () => {

    processes.forEach((p) => p.kill());

  });

  if (splashWindow) {

    splashWindow.close();

    splashWindow = null;

  }

}

app.whenReady().then(() => {

  createSplash();

  // Start Laravel backend + workers

  runArtisan(["serve"]); // Web server

  runArtisan(["queue:work"]);   // Queue worker

  runArtisan(["reverb:start"]); // Reverb server

  // Wait until Laravel is ready before showing main window

  checkLaravelReady("http://127.0.0.1:8000/login", () => {

    createMainWindow();

  });

});

app.on("window-all-closed", () => {

  if (process.platform !== "darwin") {

    processes.forEach((p) => p.kill());

    app.quit();

  }

});