



چطوری میتونم برای electron forge یک installer بنویسم
مثل همه اپ های دیگه که در زمان نصب میتونی تنظیمات رو انجام بدی یا مثلا مسیر نصب رو انتخاب کنی

سؤال خیلی خوبی پرسیدی 🍌

به صورت پیش فرض **Electron Forge** با پکیجیهایی مثل **electron-packager** و **electron-builder** یا **electron@** **forge/maker-squirrel** کار می کنه.
اما چیزی که تو می خواهی (مثل wizard نصبی با انتخاب مسیر نصب یا گزینه های سفارشی) دیگه در سطح **installer** هست، نه فقط build معمولی.
راهکارها:

1. استفاده از electron-builder

electron-builder نسبت به Forge خیلی انعطاف پذیرتره برای ساخت **installer**.
می تونی داخل **package.json** یا **electron-builder.yml** تنظیمات اضافه کنی:

json

```
"build": {
  "appId": "com.example.app",
  "productName": "MyApp",
  "directories": {
    "output": "dist"
  },
  "nsis": {
    "oneClick": false,
    "allowToChangeInstallationDirectory": true,
    "createDesktopShortcut": true,
    "createStartMenuShortcut": true
  },
  "win": {
    "target": "nsis"
  }
}
```

- **oneClick: false** → یعنی wizard نصب مرحله به مرحله ساخته میشه
- **allowToChangeInstallationDirectory: true** → کاربر می تونه مسیر نصب رو تغییر بده

بعد با این دستور بساز:

bash

```
npx electron-builder --win nsis
```

2. استفاده از NSIS (Nullsoft Scriptable Install System)

اگر می‌خواهی خیلی سفارشی بشه (فرم انتخاب زبان، تنظیمات خاص، نصب سرویس‌ها و ...) می‌تونی خروجی electron forge رو بسازی (.exe یا .zip) و بعد خودت یک **NSIS script** برای installer بنویسی. Forge بهت فایل خام (app build) میده، بعد تو با NSIS تبدیلش می‌کنی به installer. یک نمونه ساده اسکریپت NSIS:

```
nsis

OutFile "MyAppInstaller.exe"
InstallDir "$PROGRAMFILES\MyApp"

Page directory
Page instfiles

Section "Install"
    SetOutPath $INSTDIR
    File /r "path\to\your\electron\build\*.*)"
    CreateShortcut "$DESKTOP\MyApp.lnk" "$INSTDIR\MyApp.exe"
SectionEnd
```

3. ترکیب Forge + electron-builder

Forge قابلیت داره که maker های مختلف استفاده کنی. می‌تونی `electron-forge/maker-squirrel` یا حتی `electron-forge/maker-nsis` نصب کنی و داخل `forge.config.js` تنظیمش کنی. ولی در عمل خیلی‌ها مستقیم از `electron-builder` استفاده می‌کنن چون امکانات نصب حرفه‌ای‌تر میده.

👉 پیشنهاد من:

اگر چیزی مثل نرم‌افزارهای مرسوم ویندوز (Next, Install path, Options) می‌خواهی → برو سراغ `electron-builder` با `target nsis`. اگر installer خیلی خاص می‌خواهی (مثل نصب مازول‌ها یا اجرا شدن اسکریپت‌های اختصاصی) → باید NSIS رو مستقیم بنویسی.

می‌خواهی من به نمونه کامل کانفیگ `electron-builder` بنویسم که داخل پروژه Forge خودت بذاری و مستقیم installer با انتخاب مسیر نصب بسازه؟



با توضیحاتی که دادی به نظر میاد استفاده از NSIS میتونه گزینه مناسبی باشه
چطوری میتونم بنویسمش؟

خیلی عالی 🌹 انتخاب درستی کردی. NSIS (Nullsoft Scriptable Install System) برای ایجاد **installer حرفه‌ای** فوق‌العاده است و دقیقاً همون امکاناتی رو میده که دنبالش (انتخاب مسیر نصب، تیک زدن گزینه‌ها، اجرای اسکریپت‌های سفارشی و ...).

🔧 مراحل ساخت Installer با NSIS برای اپ Electron

1. نصب NSIS

- [دانلود: NSIS official](#)
- بعد از نصب، دستور makensis باید توی CMD/PowerShell در دسترس باشه.

2. خروجی گرفتن از Electron Forge

ابتدا اپ رو با Forge بساز:

bash

```
npm run make
```

این دستور معمولا خروجی رو داخل مسیر

bash

```
out/make/
```

قرار میده (مثلا out/make/squirrel.windows/x64/MyApp-1.0.0.exe یا یک پوشه کامل از app).

3. نوشتن یک NSIS Script ساده

یک فایل مثلا installer.nsi بساز:

nsis

```
; اسم فایل خروجی نصب‌کننده
OutFile "MyAppInstaller.exe"

; مسیر نصب پیش‌فرض
InstallDir "$PROGRAMFILES\MyApp"

; صفحه‌ها
Page directory      ; صفحه انتخاب مسیر نصب
Page instfiles      ; صفحه نصب فایل‌ها

; بخش نصب
Section "Install"
    ; تعیین مسیر
    SetOutPath $INSTDIR

    ; کپی همه فایل‌های برنامه
    File /r "out\make\*.*)"

    ; ساخت شورتکات روی دسکتاپ
    CreateShortcut "$DESKTOP\MyApp.lnk" "$INSTDIR\MyApp.exe"
SectionEnd

; بخش حذف نصب
Section "Uninstall"
    Delete "$DESKTOP\MyApp.lnk"
    RMDir /r "$INSTDIR"
SectionEnd
```

4. ساخت Installer

بعد از نوشتن اسکریپت:

```
makensis installer.nsi
```

```
bash
```

این دستور MyAppInstaller.exe رو می‌سازه که یک wizard کامل نصبه.

5. امکانات اضافه (اختیاری)

• گزینه‌های نصب (Checkbox)

می‌تونن سفارشی اضافه کنی مثل:

```
nsis
```

```
Page components
```

بعد داخل Section ها چند بخش مجزا تعریف کنی (مثلا "Create Desktop Shortcut", "Install Extra Tools").

• انتخاب زبان

با LoadLanguageFile می‌تونن فایل زبان اضافه کنی.

• Startup و Registry

می‌تونن کلید رجیستری بسازی یا اپ رو به startup ویندوز اضافه کنی.

جمع‌بندی: 

1. خروجی Forge رو بگیر.

2. یک .nsi بساز.

3. با makensis خروجی installer بساز.

می‌خواهی من به اسکریپت NSIS پیشرفته‌تر برات آماده کنم که:

- انتخاب مسیر نصب داشته باشه ☒
- شورتکات دسکتاپ و استارت منو بسازه ☒
- uninstall هم داشته باشه ☒
- گزینه‌ی checkbox برای "Run after install" بده ☒



makensis : The term 'makensis' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.

At line:1 char:1

+ makensis

+ ~~~~~

+ CategoryInfo : ObjectNotFound: (makensis:String) [], CommandNotFoundException

+ FullyQualifiedErrorId : CommandNotFoundException

اما این اخطار رو دریافت میکنم

آها! این خطا به این معنیه که **NSIS نصب هست یا نصب نشده/وارد PATH سیستم نشده**. وقتی PowerShell می‌گه **makensis** شناخته نشد، یعنی ویندوز نمی‌دونه کجا دنبال فایل اجرایی **makensis.exe** بگرده.

رفع مشکل

1. اطمینان از نصب NSIS

- برو به سایت رسمی: [NSIS Download](#)
- آخرین نسخه رو نصب کن (معمولاً مسیر پیش فرض نصب:

java

C:\Program Files (x86)\NSIS\

یا

makefile

C:\Program Files\NSIS\

(

داخل اون پوشه باید فایل **makensis.exe** باشه.

2. اضافه کردن به PATH

برای اینکه بتونی توی PowerShell یا CMD اجراش کنی:

1. دکمه **Start** → **Environment Variables** سرچ کن

2. وارد بخش **Environment Variables ...** شو

3. در بخش **System variables** → روی **Path** دابل کلیک کن

4. دکمه **New** رو بزن و مسیر نصب NSIS رو اضافه کن، مثلاً:

java

C:\Program Files (x86)\NSIS\

5. ذخیره کن → پنجره‌ها رو ببند → یک PowerShell جدید باز کن

3. تست

الان باید این دستور جواب بده:

powershell

```
makensis -VERSION
```

اگر درست باشه باید چیزی مثل این نشون بده:

```
v3.09
```

4. جایگزین موقت

اگر نمی‌خواهی PATH رو تغییر بدی، می‌تونی مستقیم از مسیر نصب اجراش کنی:

powershell

```
& "C:\Program Files (x86)\NSIS\makensis.exe" installer.nsi
```

می‌خواهی من قدم‌به‌قدم نشونت بدم چطوری همین `installer.nsi` که داری رو کامپایل کنیم (با مثال عملی روی همون خروجی Forge)؟



خب این مشکل حل شد

مراحل ساخت `installer.nsi` رو خیلی ساده و با جزئیات کامل توضیح بده

خیلی خوب — قدم‌به‌قدم و ساده با جزئیات کامل توضیح میدم تا بتونی `installer.nsi` رو بنویسی و خروجی `exe` نصبی رو بسازی.

نمونه کامل `installer.nsi` (قابل کپی/چسبوندن)

این مثال از Modern UI (MUI2) استفاده می‌کنه، امکانات معمول مثل انتخاب مسیر نصب، صفحه‌های لایسنس، صفحه‌های Components، شورتکات دسکتاپ/استارت منو، و گزینه "Run after install" رو داره. فقط جای متغیرها (`MyApp`)، مسیر فایل‌ها، نام `exe` و ... رو با مقادیر خودت عوض کن.

nsis

```
; -----  
; installer.nsi - نمونه ساده ولی کامل برای آپ -  
; -----  
  
!include "MUI2.nsh" ; Modern UI 2
```

```

; ----- اطلاعات پایه -----
Name "MyApp" ; نام نمایش داده شده در پنجره ها
OutFile "MyApp-Installer.exe" ; نام فایل خروجی نصب
InstallDir "$PROGRAMFILES\MyCompany\MyApp" ; مسیر پیش فرض نصب
RequestExecutionLevel admin ; نیاز به دسترسی ادمین برای نوشتن
SetCompressor lzma ; فشرده سازی بهتر

; آیکون نصب (اختیاری)
; Icon "installer.ico"
; UninstallIcon "uninstaller.ico"

; ----- صفحات نصب (MUI) -----
!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_LICENSE "license.txt" ; اگر لایسنس داری
!insertmacro MUI_PAGE_COMPONENTS ; انتخاب کامپوننت ها / گزینه ها
!insertmacro MUI_PAGE_DIRECTORY ; انتخاب پوشه نصب
!insertmacro MUI_PAGE_INSTFILES

; Finish در صفحه "run the app after install" گزینه
!define MUI_FINISHPAGE_RUN "$INSTDIR\MyApp.exe"
!insertmacro MUI_PAGE_FINISH

; (uninstall) صفحات حذف نصب
!insertmacro MUI_UNPAGE_CONFIRM
!insertmacro MUI_UNPAGE_INSTFILES

!insertmacro MUI_LANGUAGE "English" ; اگر فایل زبان داری "Persian" یا

; ----- متغیرها / داکيومنتيشن -----
!define APP_NAME "MyApp"
!define COMPANY "MyCompany"
!define VERSION "1.0.0"
!define EXE_NAME "MyApp.exe"

; ----- بخش نصب -----
Section "Install Files" SecFiles
; مسیر استخراج فایل ها
SetOutPath "$INSTDIR"

; را بنویس ... Electron Forge/packager/ اینجا مسیر خروجی ساخته شده توسط
; است out/make یا dist/win-unpacked مثال: اگر خروجی تو
File /r "path\to\your\built\win-unpacked\*" ; <-- این مسیر را تغییر بده

; ایجاد فولدر در استارت منو و شورتکات
CreateDirectory "$SMPROGRAMS\${COMPANY}\${APP_NAME}"
CreateShortCut "$SMPROGRAMS\${COMPANY}\${APP_NAME}\${APP_NAME}.lnk" "$INSTDIR\${EXE_NAME}"
CreateShortCut "$DESKTOP\${APP_NAME}.lnk" "$INSTDIR\${EXE_NAME}"

; نوشتن فایل حذف نصب داخل پوشه نصب
WriteUninstaller "$INSTDIR\Uninstall.exe"

; Programs & Features ثبت کردن در رجیستری برای نمایش در
WriteRegStr HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
"DisplayName" "${APP_NAME}"
WriteRegStr HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
"UninstallString" "$INSTDIR\Uninstall.exe"
WriteRegStr HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
"DisplayVersion" "${VERSION}"
WriteRegStr HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}" "Publisher"
"${COMPANY}"
WriteRegStr HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
"InstallLocation" "$INSTDIR"
SectionEnd

; ----- بخش حذف نصب -----
Section "Uninstall"
; پاک کردن شورتکات ها
Delete "$DESKTOP\${APP_NAME}.lnk"
Delete "$SMPROGRAMS\${COMPANY}\${APP_NAME}\${APP_NAME}.lnk"

; حذف فایل حذف نصب (برای اینکه خود حذف نصب هم حذف بشه)
Delete "$INSTDIR\Uninstall.exe"

; حذف کل دایرکتوری نصب به صورت بازگشتی

```

```
RMDir /r "$INSTDIR"
```

```
Uninstall حذف کلید رجیستری ;
DeleteRegKey HKLM "Software\Microsoft\Windows\CurrentVersion\Uninstall\${APP_NAME}"
SectionEnd
```

توضیح مرحله به مرحله (هر بلوک چی کار می‌کنه)

1. `"include "MUI2.nsh!"`

افزونهٔ Modern UI2 رو میاره تا صفحات آماده (Welcome, License, Directory, Finish و ...) داشته باشی بدون اینکه دستی طراحی کنی.

2. **تعاریف پایه** (`Name`, `OutFile`, `InstallDir`)

- `Name` اسم نمایشیسه.
- `OutFile` اسم exe نصاب.
- `InstallDir` مسیر پیش فرض نصب (معمولاً `Program Files`).
- `RequestExecutionLevel admin` باعث میشه ویندوز از کاربر اجازهٔ ادمین بگیره تا بتونی توی `Program Files/Registry` بنویسی.

3. **فشرده سازی** (`SetCompressor lzma`)

فایل خروجی کوچکتتر میشه. اختیاریه.

4. **تعریف صفحات با** `insertmacro!`

هر `*_MUI_PAGE` یک صفحهٔ آماده اضافه می‌کنه: خوش آمد، لایسنس، انتخاب کامپوننت، انتخاب مسیر نصب، و صفحهٔ نصب. برای صفحهٔ لایسنس، فایل متنی `license.txt` کنار `installer.nsi` باشه.

5. **گزینهٔ "Run after install"**

`define MUI_FINISHPAGE_RUN "$INSTDIR\MyApp.exe!` باعث میشه در صفحهٔ `Finish` یک تیک برای اجرای برنامه بعد نصب نمایش داده بشه.

6. **بخش نصب** (`"Section "Install Files"`)

- `SetOutPath "$INSTDIR` مسیر استخراج فایل‌ها را مشخص می‌کند.
- `File /r "..."` فایل‌ها را به صورت `recursive` داخل `$INSTDIR` کپی می‌کند — این مسیر باید مسیر خروجی اپ ساخته شده توسط `Forge/packager` باشه.
- `CreateShortCut` برای شورتکات دسکتاپ و استارت منو.
- `WriteUninstaller` یک `Uninstall.exe` داخل پوشهٔ نصب می‌سازد.
- `WriteRegStr` رجیستری را پر می‌کند تا اپ در `Programs & Features` نمایش داده شود.

7. **بخش حذف نصب** (`"Section "Uninstall"`)

شورتکات‌ها و فایل‌ها را حذف می‌کند و کلید رجیستری پاک می‌شود.

چگونه بسازم و تست کنم

1. خروجی اپ Electron را آماده کن (با `npm run make` یا `electron-builder` / `electron-packager`). مسیر به EXE تولیدی را پیدا کن (مثلاً `out/make/win-unpacked` یا `dist/win-unpacked`).

2. مسیر `*.*\File /r "path\to\your\built\win-unpacked` را داخل `installer.nsi` اصلاح کن.

3. از PowerShell یا CMD اجرا کن:

```
powershell
makensis installer.nsi
```

یا اگر `makensis` در `PATH` نیست:

```
powershell
& "C:\Program Files (x86)\NSIS\makensis.exe" installer.nsi
```

4. فایل `MyApp-Installer.exe` تولید میشه — حالا روش دوبار کلیک کن و نصب رو تست کن:

- مسیر نصب رو عوض کن بین درست کپی میشه یا نه.
- شورتکات دسکتاپ و استارت منو بسازن یا نه.
- گزینه "Run after install" کار کنه یا نه.
- بعد از Uninstall از Control Panel یا اجرای `Uninstall.exe` تست کن که همه چیز پاک بشه.

نکات و ترفندهای مفید

- **آزمون در ماشین مجازی:** قبل از انتشار در محیط تمیز (VM یا ماشین تست) امتحان کن تا مشکلات مجوز/فایل قفل نشدن مشخص بشه.
- **کد ساینینگ (Code Signing):** برای جلوگیری از هشدارهای SmartScreen و دریافت اعتماد بهتر، نصاب و exe رو امضا کن (Windows signtool).
- **مسائل رجیستری 64/32 بیتی:** اگر نیاز داری که مقدار رجیستری در view مشخص نوشته بشه، از `SetRegView 64` استفاده کن.
- **کمپوننت‌ها (Components):** با `insertmacro MUI_PAGE_COMPONENTS!` و تعریف بخش‌های مجزا، گزینه‌های نصب مثل "Install CLI tool" یا "Create Desktop Shortcut" به کاربر نمایش داده میشه.
- **کامپایل خودکار در pipeline:** می‌تونی در `package.json` اسکریپت بزاری:

```
json
{
  "scripts": {
    "make": "electron-forge make",
    "build:installer": "npm run make && makensis installer.nsi"
  }
}
```

- **مسیرها و فاصله‌ها:** اگر مسیر نصب یا فایل‌های منبع شامل space باشن، حواست باشه که بعضی مقادیر در رجیستری نیاز به quoting دارن. معمولاً NSIS متغیرها را درست هندل می‌کنه ولی اگر مشکلی دیدی میشه حولش راهنمایی کنم.

اشتباهات رایج

- نگذاشتن مسیر `File /r` درست → فایل‌ها داخل `installer` نیستن.

- اجرا نکردن makensis با دسترسی مناسب یا نصب نبوده NSIS (خطای makensis not recognized).
- فراموش کردن WriteUninstaller (در نتیجه Uninstall.exe نیست).
- تست نکردن در حالت non-admin (نصب به Program Files نیاز به admin دارد).

اگر خواستی این اسکریپت رو دقیقاً برای پروژه خودت سفارشی کنم، فقط بفرست:

1. مسیر فولدر خروجی (مثلاً out/make/win-unpacked یا dist/win-unpacked)

2. نام فایل exe دقیق (مثلاً MyApp.exe)

من برات installer.nsi رو با مقادیر جاگذاری شده و چند توضیح عملی تولید می‌کنم تا فقط makensis installer.nsi بزنی و خروجی بگیری.



```
LicenseData: open failed "license.txt"
Usage: LicenseData local_file_that_has_license_text | license_lang_string
Error in macro MUI_PAGEDECLARATION_LICENSE on macroline 17
Error in macro MUI_PAGE_LICENSE on macroline 6
Error in script "installer.nsi" on line 20 -- aborting creation process
این اخطار رو دریافت میکنم
```

🔥 خب این خطا خیلی ساده‌ست:

NSIS بهت میگه نمی‌تونه فایل license.txt رو پیدا کنه.
تو اسکریپت این خط وجود داشت:

```
nsis

!insertmacro MUI_PAGE_LICENSE "license.txt"
```

یعنی انتظار داره کنار installer.nsi فایل به اسم license.txt وجود داشته باشه.

🛣️ راه‌حل‌ها

1. اگر فایل لایسنس داری

- یک فایل متنی ساده بساز (UTF-8 بدون BOM باشه بهتره).
- اسمشو بذار license.txt و بذارش تو همون فولدری که installer.nsi هست.
- مثلاً:

```
java

My Project
Copyright (c) 2025 MyCompany
Licensed under MIT License.
```

بعد دوباره makensis installer.nsi رو بزنی.

2. اگر لایسنس نداری یا نمی‌خواهی نمایش داده بشه
خیلی راحت اون خط رو حذف کن یا کامنت کن:

```
nsis
```

```
; !insertmacro MUI_PAGE_LICENSE "license.txt"
```

اینطوری صفحه License توی wizard حذف میشه.

✓ بهترین کار برای تست

الان چون داری تازه اسکریپت رو تست می‌کنی، پیشنهاد می‌کنم اون خط رو موقتاً **کامنت** کنی. وقتی خواستی نسخه نهایی بدی، می‌تونی `license.txt` اضافه کنی.

می‌خواهی من نسخه بدون لایسنس همین `installer.nsi` رو برات بازنویسی کنم که مستقیم کار کنه؟



```
File /r "E:\Lerning\Programing\Js\electron-vite\out\make\squirrel.windows\x64\electron-vite-1.0.0 Setup.exe" ;
```

من ایم مسیر اپ رو بهش دادم اما این اخطار رو بهم میده

```
File: "path\to\your\built\win-unpacked\*.*)" -> no files found.
```

```
Usage: File [/nonfatal] [/a] ([/r] [/x filespec [...]] filespec [...] |
```

```
/oname=outfile one_file_only)
```

```
Error in script "installer.nsi" on line 48 -- aborting creation process
```

آها گرفتم 🌹

مشکل اینجاست که تو بخش `File /r` باید فایل‌های **unpack شده (محتویات برنامه)** رو به NSIS بدی، نه فایل `installer` خروجی Forge (مثل `Setup.exe`).

چرا این اتفاق میفته؟

- `electron-forge make --target squirrel` برات یک **installer (Setup.exe)** می‌سازه.
- ولی ما داریم با NSIS یک **installer جدید** می‌سازیم. Installer داخل Installer منطقی نیست 😊
- NSIS باید فایل‌های **unpack شده** اپ رو بگیره (یعنی فولدری که شامل `YourApp.exe` و بقیه `dll`, `resources` و ... هست).

راه درست پیدا کردن فایل‌ها

بعد از `npm run make`، تو پوشه `out/make` باید یک چیزی شبیه این باشه:

```

out/
└─ make/
   └─ squirrel.windows/
      └─ x64/
         └─ electron-vite-1.0.0 Setup.exe
└─ win-unpacked/
   └─ electron-vite.exe
   └─ resources/
   └─ ...

```

پیش‌فرض فُرج هست installer این همون ←
!اینو نیاز داریم ←

📌 تو باید از win-unpacked استفاده کنی.
اونجا تمام فایل‌های لازم برای اجرای برنامه‌ت وجود دارن.

اصلاح خط NSIS

به جای:

nsis

```
File /r "E:\Lerning\Programing\Js\electron-vite\out\make\squirrel.windows\x64\electron-vite-1.0.0 Setup.exe"
```

بذار:

nsis

```
File /r "E:\Lerning\Programing\Js\electron-vite\out\make\win-unpacked\*.*)"
```

تست دوباره

1. مسیر درست فولدر win-unpacked رو پیدا کن.

2. اون رو توی File /r بذار.

3. بعد:

powershell

```
makensis installer.nsi
```

بزنی باید بدون خطا نصاب ساخته بشه.

می‌خوای من برات یک نسخه‌ی جمع‌وجور installer.nsi فقط با **Directory + Install + Uninstall** بسازم (خیلی ساده برای شروع)، تا مطمئن بشی مشکل مسیر حل شده و همه‌چی درست کار می‌کنه، بعد بریم سراغ امکانات اضافه مثل شورتکات و Finish page؟



من باز هم همون ارور رو دریافت میکنم

به نظرت عجیب نیست

باوجودی که من مسیر رو تغییر دادم اما باز هم داخل مات ارور مسیر قبلی رو بهم میده!!!!

آها 🤩 خیلی نکته‌ی خوبی گرفتی.

اینکه بعد از تغییر دادن مسیر توی `installer.nsi`، باز هم خطا مسیر قبلی رو نشون میده، نشون میده که **NSIS هنوز داره نسخه قدیمی اسکریپت رو می‌خونه** (یا فایل اشتباه رو اجرا می‌کنی).

۱. مطمئن شو همون فایل رو داری ویرایش می‌کنی

- چک کن واقعاً داری همون `installer.nsi` که باهاش `makensis` رو صدا می‌زنی ویرایش می‌کنی.
- خیلی وقت‌ها آدم چند نسخه کپی کرده (مثلاً یکی توی `electron-vite\installer.nsi` و یکی توی دسکتاپ) و اشتباهی یکی دیگه اجرا میشه.

۲. مسیر کامل رو به `makensis` بده

به جای اینکه بری داخل فولدر و بزنی:

powershell

```
makensis installer.nsi
```

اینطوری اجرا کن:

powershell

```
makensis "E:\Lerning\Programing\Js\electron-vite\installer.nsi"
```

اینجوری مطمئن میشی دقیقاً همون فایل داره کامپایل میشه که تو تغییرش دادی.

۳. مسیر `File /r` دوباره بررسی کن

فرض کنیم خروجی Forge به این شکله:

csharp

```
E:\Lerning\Programing\Js\electron-vite\out\make\win-unpacked\
```

پس داخل `installer.nsi` باید دقیقاً باشه:

nsis

```
File /r "E:\Lerning\Programing\Js\electron-vite\out\make\win-unpacked\*.*"
```

- ستاره‌ها (*) مهمن چون NSIS باید همه فایل‌ها رو بگیره.
- آخر مسیر حتماً \ قبل از *.* باشه.

۴. لاگ بگیر

برای اینکه مطمئن بشی تغییراتت خونده شده، همون اول فایل `installer.nsi` اینو اضافه کن:

nsis

```
MessageBox MB_OK "NSIS is compiling THIS script!"
```

وقتی `makensis` رو بزنی، اگر این پیام ظاهر بشه یعنی فایل درست داری اجرا میشه.
اگر ظاهر نشد → داری فایل اشتباه رو کامپایل می‌کنی.

به نظرم اول با هم تست کنیم:
آیا می‌خوای من یک نسخه خیلی مینیمال `installer.nsi` (فقط شامل `File /r` و خروجی ساده) برات بنویسم که مطمئن بشیم مشکل از مسیر یا از فایل اشتباه بودن هست؟



```
E:\Lerning\Programing\Js\electron-vite\out\electron-vite-win32-x64\*.*
```

دادم که حاوی فایلهای برنامه من هستش installer من الان این مسیر رو به

لیستش هم به این صورته

```
[
'chrome_100_percent.pak',
'chrome_200_percent.pak',
'd3dcompiler_47.dll',
'electron-vite.exe',
'ffmpeg.dll',
'icudtl.dat',
'libEGL.dll',
'libGLESv2.dll',
'LICENSE',
'LICENSES.chromium.html',
'locales',
'resources',
'resources.pak',
'snapshot_blob.bin',
'Squirrel-Unset.log',
'Squirrel.exe',
'v8_context_snapshot.bin',
'version',
'vk_swiftshader.dll',
'vk_swiftshader_icd.json',
'vulkan-1.dll'
```

]

رو اجرا کنه همه چی درست میشه electron-vite.exe فایل NSIS اگر در نهایت

اما نمیدونم چطوری باید بهش بفهمونم

درست میگم یا دارم اشتباه میکنم؟

دقیقاً درست فهمیدی 🧠

کاری که ما از NSIS می‌خوایم اینه که:

1. **کل محتویات فولدر اپت** (همون electron-vite-win32-x64) رو کپی کنه داخل `INSTDIR$` (پوشه انتخابی کاربر در زمان نصب).

2. بعد یک **شورتکات یا دستور اجرا** بذاره که فایل اصلی (electron-vite.exe) رو باز کنه.

📄 نمونه اسکریپت ساده برای شرایط تو

چون خروجی اپت توی این مسیره:

csharp

E:\Lerning\Programing\Js\electron-vite\out\electron-vite-win32-x64\

اسکریپت NSIS باید چیزی مثل این باشه:

nsis

```
!include "MUI2.nsh"

Name "Electron Vite App"
OutFile "ElectronViteInstaller.exe"
InstallDir "$PROGRAMFILES\ElectronViteApp"
RequestExecutionLevel admin

!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_INSTFILES
!insertmacro MUI_PAGE_FINISH

!insertmacro MUI_LANGUAGE "English"

Section "Install"
    ; مسیر مقصد نصب
    SetOutPath "$INSTDIR"

    ; همه فایل‌های اپ رو کپی کن
    File /r "E:\Lerning\Programing\Js\electron-vite\out\electron-vite-win32-x64\*.*"

    ; شورتکات روی دسکتاپ
    CreateShortCut "$DESKTOP\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"

    ; شورتکات در Start Menu
    CreateDirectory "$SMPROGRAMS\Electron Vite"
    CreateShortCut "$SMPROGRAMS\Electron Vite\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"

    ; فایل Uninstaller
    WriteUninstaller "$INSTDIR\Uninstall.exe"
SectionEnd
```

```

Section "Uninstall"
Delete "$DESKTOP\Electron Vite.lnk"
Delete "$SMPROGRAMS\Electron Vite\Electron Vite.lnk"
RMDir "$SMPROGRAMS\Electron Vite"

RMDir /r "$INSTDIR"
SectionEnd

```

توضیحات مهم

- `**.*\File /r "E:\...\electron-vite-win32-x64`
این خط کل فایل‌های اون فولدر رو میاره داخل installer و هنگام نصب میریزه توی `.INSTDIR`.
- `CreateShortCut`
باعث میشه وقتی کاربر نصب کرد، یک شورتکات روی دسکتاپ و استارت‌منو ایجاد بشه که مستقیماً `electron-vite.exe` رو اجرا کنه.
- `WriteUninstaller`
یه `Uninstall.exe` می‌سازه که بعداً اپ رو کامل پاک کنه.

تست

1. `makensis installer.nsi` بزن.
 2. نصب رو تست کن.
- پوشه پیش‌فرض: `C:\Program Files\ElectronViteApp`
 - باید همه فایل‌ها کپی بشن.
 - شورتکات روی دسکتاپ ایجاد بشه.
 - با دوبار کلیک روی شورتکات، `electron-vite.exe` اجرا بشه.

سؤال: 🗨 می‌خواهی من همین اسکرپت رو به مقدار کامل‌تر کنم که توی **Control Panel (Programs & Features)** هم اسم برنامه‌ت بیاد (مثل برنامه‌های عادی ویندوز)؟



عالی شد و همه چی داره خوب کار میکنه
فقط چرا بعد از نصب اپ رو به صورت اتوماتیک باز نمیکنه؟

آفرین 🙌 خوشحالم که درست کار کرده.
اینکه بعد از نصب برنامه به صورت خودکار باز نمیشه طبیعیه، چون توی اسکرپت ما فقط شورتکات ساختیم و نگفتیم که **بعد از اتمام نصب** اپ اجرا بشه.

✓ **راه‌حل: اجرای اپ بعد از نصب**

1. استفاده از Finish Page (پیشنهادی)

یه تیک توی آخرین صفحه نصاب میاد که میگه "Run Electron Vite" → کاربر اگر بخواد، برنامه بلافاصله اجرا میشه. کافیه این خطوط رو توی `installer.nsi` اضافه کنی (زیر صفحات MUI):

```
nsis

; با گزینه اجرای برنامه Finish صفحه
!define MUI_FINISHPAGE_RUN "$INSTDIR\electron-vite.exe"
!insertmacro MUI_PAGE_FINISH
```

پس به جای چیزی که الان داری (فقط `!insertmacro MUI_PAGE_FINISH`), این دوتا خط رو بذار. بعد از نصب، توی صفحه آخر یه چک باکس میاد: ☒ Run Electron Vite

2. اجرای مستقیم (بدون چک باکس)

اگر می‌خواهی همیشه بعد از نصب اپ اجرا بشه، می‌تونی آخر بخش نصب اینو بذاری:

```
nsis

Exec "$INSTDIR\electron-vite.exe"
```

یعنی بخش `Section "Install"` اینطوری میشه:

```
nsis

Section "Install"
  SetOutPath "$INSTDIR"
  File /r "E:\Lerning\Programing\Js\electron-vite\out\electron-vite-win32-x64\*.*)"

  CreateShortCut "$DESKTOP\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"
  CreateDirectory "$SMPROGRAMS\Electron Vite"
  CreateShortCut "$SMPROGRAMS\Electron Vite\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"

  WriteUninstaller "$INSTDIR\Uninstall.exe"

  ; اجرای اپ بعد از نصب
  Exec "$INSTDIR\electron-vite.exe"
SectionEnd
```

- ♦ اگر بخوای انتخاب دست کاربر باشه → راه اول (چک باکس در Finish Page).
- ♦ اگر بخوای همیشه باز بشه → راه دوم (Exec مستقیم).

می‌خواهی من نسخه نهایی `installer.nsi` رو برات بازنویسی کنم که هم شورتکات بسازه، هم داخل Control Panel نمایش داده بشه، و هم توی Finish Page تیک "Run after install" داشته باشه؟



چطوری میتونم درکنار این installer موارد دیگری هم نصب کنم مثلا به کاربر این امکان رو بدم nodejs هم نصب کنه

آها، حالا میخوای Installer خودت رو چند مرحله‌ای و چند کامپوننتی کنی، یعنی کاربر موقع نصب بتونه انتخاب کنه که علاوه بر اپ، چیزهای دیگه مثل **Node.js** یا ابزارهای جانبی هم نصب بشه. در NSIS دو راه معمول هست:

1 استفاده از Components Page (چک‌باکس انتخابی)

با MUI2 می‌تونی بخش‌های مختلف نصب رو تعریف کنی. هر بخش (Section) می‌تونه شامل فایل‌ها یا یک اسکریپت نصب باشه. کاربر در صفحه Components می‌تونه تیک بزنه یا نزنه.

مثال ساده:

```
nsis

!include "MUI2.nsh"

Name "Electron Vite App"
OutFile "ElectronViteInstaller.exe"
InstallDir "$PROGRAMFILES\ElectronViteApp"
RequestExecutionLevel admin

!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_COMPONENTS ; صفحه انتخاب کامپوننت‌ها
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_INSTFILES
!define MUI_FINISHPAGE_RUN "$INSTDIR\electron-vite.exe"
!insertmacro MUI_PAGE_FINISH
!insertmacro MUI_LANGUAGE "English"

; ----- بخش اصلی اپ -----
Section "Install Electron Vite" SecApp
  SetOutPath "$INSTDIR"
  File /r "E:\Lerning\Programing\Js\electron-vite\out\electron-vite-win32-x64\*.*)"
  CreateShortCut "$DESKTOP\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"
  CreateDirectory "$SMPROGRAMS\Electron Vite"
  CreateShortCut "$SMPROGRAMS\Electron Vite\Electron Vite.lnk" "$INSTDIR\electron-vite.exe"
  WriteUninstaller "$INSTDIR\Uninstall.exe"
SectionEnd

; ----- بخش نصب Node.js -----
Section "Install Node.js" SecNode
  ; نصب نیست Node.js چک می‌کنیم که
  ReadRegStr $0 HKLM "SOFTWARE\Node.js" "InstallPath"
  StrCmp $0 "" +2
  Goto NodeInstalled

  ; Node.js installer دانلود یا اجرای
  ; اصلی گذاشتی installer رو کنار nodejs فرض کنید فایل
  ExecWait "$EXEDIR\node-v20.7.0-x64.msi" /quiet /norestart

NodeInstalled:
SectionEnd
```

توضیحات مهم:

- `insertmacro MUI_PAGE_COMPONENTS!` → صفحه‌ای با چک‌باکس میاره که هر `Section` بهش مربوطه.
- هر `Section` یک کامپوننته:
- `Install Electron Vite` → اپ خودت
- `Install Node.js` → نصب `Node`
- `ExecWait` → دستور نصب `Node.js` رو اجرا می‌کنه (مثلاً `MSI` یا `EXE`). `/quiet /norestart` باعث میشه بدون دخالت کاربر نصب بشه.
- `EXEDIR$` → مسیر `Installer` فعلی، پس `Node installer` کنار `installer` اصلی باشه.

2 نصب برنامه‌های جانبی از اینترنت

اگر نمی‌خواه `Node.js` رو همراه `Installer` بذاری، می‌تونی **NSIS** دانلود کنه و نصب کنه:

```
nsis

Section "Install Node.js" SecNode
; دانلود Node.js
nsisdl::download "https://nodejs.org/dist/v20.7.0/node-v20.7.0-x64.msi" "$TEMP\nodejs.msi"
Pop $0
StrCmp $0 "success" 0 +2
ExecWait "$TEMP\nodejs.msi" /quiet /norestart'
SectionEnd
```

⚠ نکته: برای این روش کاربر باید اینترنت فعال داشته باشه.

✓ جمع‌بندی

- با **Components Page** می‌تونی چند گزینه نصب به کاربر بدهی.
- هر بخش می‌تونه فایل‌های خودت یا یک `installer` خارجی (`Node`, `Git`, `Redis` و ...) اجرا کنه.
- بعد از نصب، `Finish Page` هنوز می‌تونه گزینه "Run Electron Vite" داشته باشه.

اگر بخوای، می‌تونم برات یک نسخه کامل **installer.nsi** بسازم که:

1. اپ `Electron` نصب بشه
 2. `Node.js` هم با چک‌باکس نصب بشه
 3. شورتکات‌ها بسازه
 4. `Finish Page` با گزینه "Run app" داشته باشه
- می‌خواهی برات آماده کنم؟



اگر بخوام یک اسکریپت جاوااسکریپتی اجرا بکنم چطور؟

آها، می‌خوای Installer بتونه یک فایل .js اجرا کنه، یعنی یه اسکریپت جاوااسکریپت که Node.js اجراش کنه. دو نکته مهم وجود داره:

1 نیاز به Node.js نصب شده

اگر اسکریپت .js رو با Node اجرا می‌خوای، مطمئن شو که Node.js روی سیستم کاربر نصب شده. در غیر اینصورت NSIS نمی‌تونه فایل .js رو اجرا کنه.

2 اجرای فایل JS با NSIS

روش ساده با Exec یا ExecWait

فرض کنیم فایل script.js کنار Installer هست یا داخل فولدر نصب:

```
nsis

Section "Run JS Script"
    ; داخل فولدر نصب JS مسیر فایل
    ExecWait "$INSTDIR\node-vite-script.js"
SectionEnd
```

⚠ مشکل: این فقط فایل .js رو اجرا میکنه اگر Node توی PATH باشه.

روش درست (Explicit Node)

```
nsis

Section "Run JS Script"
    ; داخل فولدر نصب JS مسیر فایل
    ; باشه PATH باید نصب شده باشه و مسیرش توی Node.js
    ExecWait '"node" "$INSTDIR\script.js"'
SectionEnd
```

- "node" → فرض می‌کنیم Node توی PATH هست.
- "\$INSTDIR\script.js" → مسیر فایل JS در فولدر نصب.
- ExecWait → صبر می‌کنه اسکریپت اجرا بشه و بعد Installer ادامه بده.

نکته مهم

اگر Node.js ممکنه نصب نباشه:

1. ابتدا Node.js رو نصب کن (مثل روش Component Page یا نصب از اینترنت).

2. بعد ExecWait با Node اجرا کن.

مثال ترکیبی:

```
nsis

Section "Install Node.js and Run Script"
; نصب Node.js
ExecWait "$EXEDIR\node-v20.7.0-x64.msi" /quiet /norestart'

; Node با JS اجرای فایل
ExecWait "$INSTDIR\script.js"
SectionEnd
```

⚠ بهتره صریح Node رو صدا بزنی:

```
nsis

ExecWait "$INSTDIR\nodejs\node.exe" "$INSTDIR\script.js"
```

اگر Node داخل پوشه خودش نصب شده باشه.

💡 جمع‌بندی:

- js به تنهایی توسط ویندوز اجرا نمیشه، Node.js لازم داره.
- مسیر Node رو مشخص کن یا مطمئن شو توی PATH هست.
- ExecWait "node path\to\script.js" امن‌ترین راهه.

اگر بخوای، می‌تونم برات یک **مثال کامل installer.nsi** بسازم که:

1. اپ Electron نصب بشه
2. Node.js نصب بشه (اختیاری)
3. اسکریپت js اجرا بشه
4. شورتکات‌ها بسازه
5. Finish Page با تیک Run App داشته باشه

میخواهی برات آماده کنم؟



خب از این مشکل بگذریم و بریم سراغ نکته مهمتر
چطوری میتونم این installer رو برای لینوکس هم داشته باشم؟

آها، رسیدیم به بخش مهم 🌍

درست است: NSIS فقط برای **ویندوز** کار می‌کنه. برای **لینوکس** باید از روش‌های دیگری استفاده کنیم، چون لینوکس **سیستم نصب متفاوتی** دارد و معمولاً بسته‌های نصبی استاندارد هستند: .rpm (Fedora/RedHat)، deb (Debian/Ubuntu)، یا AppImage.

♦ روش‌های مرسوم برای ساخت Installer لینوکس

1 AppImage (توصیه شده برای توزیع‌های مختلف)

- یک فایل اجرایی واحد که تمام وابستگی‌ها را دارد.
- کاربر می‌تواند فقط `chmod +x MyApp.AppImage` کند و اجرا کند، بدون نیاز به نصب رسمی.
- بسیار مشابه `exe installer` Windows عمل می‌کند.

مراحل سریع:

1. خروجی لینوکس اپ خودت را بساز (با `electron-builder` یا `electron-forge`):

```
bash
npm run make
```

یا با `electron-builder`:

```
bash
electron-builder --linux AppImage
```

2. یک فایل `AppImage` تولید می‌شود (`dist/YourApp-x86_64.AppImage`).

3. کاربر می‌تواند فایل را اجرا کند:

```
bash
chmod +x YourApp-x86_64.AppImage
./YourApp-x86_64.AppImage
```

2 بسته DEB (برای Debian/Ubuntu)

- یک بسته `deb` استاندارد که با `dpkg -i` نصب می‌شود.
- به راحتی می‌توان `Shortcut` و `Menu entry` ایجاد کرد و `uninstall` هم دارد.

مراحل سریع با `electron-builder`:

```
json
{
  "build": {
    "appId": "com.example.myapp",
    "linux": {
      "target": ["deb"],
      "category": "Utility"
    }
  }
}
```

bash

```
electron-builder --linux deb
```

- خروجی: `dist/*.deb`

- نصب روی Ubuntu/Debian

bash

```
sudo dpkg -i YourApp_1.0.0_amd64.deb
```

3 بسته RPM (برای Fedora/RedHat)

مشابه DEB ولی برای توزیع‌های RPM:

json

```
"linux": {
  "target": ["rpm"]
}
```

سپس:

bash

```
electron-builder --linux rpm
```

- نصب روی Fedora/RedHat

bash

```
sudo rpm -i YourApp-1.0.0.x86_64.rpm
```

4 Snap یا Flatpak

- Snap و Flatpak برای توزیع‌های مختلف قابل استفاده هستند.
- معایب: نیاز به runtime مخصوص، اما راحت برای انتشار در چند توزیع.

◆ جمع‌بندی سریع

- برای توزیع گسترده لینوکس و ساده‌ترین تجربه کاربر → **AppImage**
- اگر می‌خواهی کاربران Ubuntu/Debian راحت نصب کنند → **deb**.
- اگر کاربران Fedora/RedHat داری → **rpm**.

💡 نکته مهم:

NSIS مخصوص ویندوز است، بنابراین برای لینوکس حتماً از **electron-builder** یا **AppImage** استفاده می‌کنیم. Installer لینوکس معمولاً **wizard** گرافیکی مرحله به مرحله مثل **Windows** ندارد، مگر اینکه بخواهی با ابزارهایی مثل **makeself** اسکریپت **bash** بسازی که **wizard** ساده شبیه **Windows** داشته باشد.

اگر بخوای، می‌تونم برات یک نسخه **AppImage** آماده برای پروژه **Electron** تو بسازم و مراحلش رو دقیق نشون بدم، به طوری که روی اکثر توزیع‌های لینوکس اجرا بشه بدون نیاز به نصب اضافی. میخوای این کارو انجام بدیم؟