AI & Software Engineering Workshop

Շաբաթ 1, Օր 1. Կարգավորում և առաջին հաղորդագրություն — Windows

Edik Simonian, Ամառ 2026

Բարև, ես Edik-ն եմ 👋

- **Գլխավոր ծրագրային ինժեներ NASA JPL-ում** 🚀 — 9 տարի
- **Բակալավր UC Davis**-ից · **Համակարգչային գիտության մագիստրոս Georgia Tech**-ից
- Իմ **երկրորդ տարին**՝ դասավանդելով այստեղ՝ **TUMO Yerevan**-ում

Սկզբից ծանոթանանք լսարանի հետ

Շրջանով, **30 վայրկյան յուրաքանչյուրին՝**

- **Ո՞վ ես դու:** Քո անունն ու դասարանը
- **Ի՞նչ համակարգչային գիտության դասերի ես հաճախել:** Ցանկացած ծրագրավորման դասընթաց՝ դպրոցում, ակումբում կամ առցանց — կամ առայժմ ոչ մեկը
- **Ի՞նչ ես հույս ունենում սովորել այս դասից:** Հմտություն, նախագծի տեսակ, կամ պարզապես այն, ինչ քեզ գրավեց

Սխալ պատասխաններ չկան, նույնիսկ «դեռ ոչինչ»-ն ընդունելի է: Հենց այստեղ ենք պարզում, թե ինչ ենք կառուցելու մինչև ուրբաթ:

Հիմնական կանոններ

- **Ոչ YouTube, ոչ խաղեր, ոչ հեռախոսազանգեր** սեմինարի ընթացքում
- **Արագ վիկտորինաներ** յուրաքանչյուր սեմինարի սկզբին, և կրկին՝ ընդմիջումից հետո
- **5 որպեից ավելի մի՛ ուշացիր** — բաց թողած վիկտորինան այլևս չես կարող հանձնել
- **Ոչ մի կոդ մի՛ պատճենիր ChatGPT-ից** — օգտագործիր մեր տրամադրած Claude Code-ը և հասկացիր յուրաքանչյուր տող, որ ուղարկում ես
- **Ոչ մի անձնական տվյալ բոտին** — ոչ իրական անուններ, հասցեներ, հեռախոսներ կամ գաղտնաբառեր; զրույցները գնում են արտաքին AI ծառայության

Սկզբից տեսնենք, թե ինչպես է աշխատում

t.me/tele_pythonanywhere_bot

Ահա թե որտեղ կլինես մինչև **ուրբաթ**՝ քո սեփական բոտը, քո բնավորությունը, կենդանի՝ համացանցում:

Գրիր նրան հիմա, հետո մենք մեկը զրոյից կկառուցենք:

Ինչ ես կառուցելու այս շաբաթ

- **Telegram բոտ**՝ իրական LLM-ով աշխատող
- Իր սեփական **բնավորությամբ**՝ դու ես այն նախագծում
- Հատուկ **slash հրամաններ**՝ դու ես դրանք կոդավորում
- **Հիշողություն**, որ պահպանվում է վերագործարկումներից հետո
- **Կենդանի՝ համացանցում** մինչև ուրբաթ, աշխատող նույնիսկ երբ քո համակարգիչն անջատված է

Մինչև այսօրվա վերջը՝ բոտը պատասխանում է քեզ՝ քո մեքենայի վրա եղած կոդից:

Շաբաթը՝ օր առ օր

- **Օր 1, Կարգավորում և առաջին հաղորդագրություն** *(այսօր)*՝ token-ներ, `.\make.ps1 run`, քո առաջին `/start` խմբագրումը
- **Օր 2, Բնավորություն և հուշումներ**՝ համակարգային հուշում, որ բոտիդ ձայն է տալիս — և ծանոթացում Claude Code-ի հետ
- **Օր 3, Նոր հրամաններ**՝ ուղիղ եթերում կոդավորում ենք `/joke`, հետո դու քո սեփականն ես կառուցում՝ թեստով
- **Օր 4, Հիշողություն և տեղակայում**՝ հիշողություն, որ պահպանվում է վերագործարկումներից հետո, հետո՝ կենդանի PythonAnywhere-ի վրա
- **Օր 5, Կառուցիր քո սեփական բոտը**՝ նախագծիր ու թողարկիր հնարավորություն, որ ամբողջովին քոնն է, հետո՝ ցուցադրություն

Ամեն օր հիմնվում է նախորդի վրա — մինչև ուրբաթ այն քոնն է և առցանց:

Ի՞նչ է բոտը

Telegram-ը քո կոդի հետ խոսում է երկու եղանակով՝

- **Polling**՝ քո կոդը անընդհատ հարցնում է Telegram-ին *«նոր հաղորդագրություններ կա՞ն»*
  → այն, ինչ օգտագործում ենք **այսօր**, քո համակարգչի վրա
- **Webhook**՝ Telegram-ը յուրաքանչյուր հաղորդագրություն հրում է քո սերվերի URL-ին
  → այն, ինչ օգտագործում ենք **Օր 4**, production-ում

Բոտի կոդը նույնն է երկու դեպքում էլ: Փոխվում է միայն առաքման եղանակը:

Տեխնոլոգիական կույտը

Telegram → Flask (Python) → Cerebras → reply

- **Telegram Bot API**՝ հաղորդագրությունների ինտերֆեյսը
- **Flask**՝ ստանում է հաղորդագրությունները
- **Cerebras**՝ գործարկում է LLM-ը, որ գրում է պատասխանները *(անվճար փաթեթ)*
- **SQLite**՝ հիշողություն *(Օր 4)*
- **PythonAnywhere**՝ հոստինգ *(Օր 4)*
- **GitHub**՝ քո կոդը, քո թեստերը, քո տեղակայումները

Սկզբից՝ կարգավորիր քո գործիքակազմը

*Այս լաբորատորիան աշխատում է **Windows**-ով, ուստի մենք աշխատում ենք **PowerShell**-ով՝ **Scoop**-ով — փաթեթների կառավարիչ, որ տեղադրում է մեր գործիքները **առանց ադմինի իրավունքների**:*

Բացիր **PowerShell**-ը (Start menu → գրիր *PowerShell*), հետո գործարկիր **մեկ անգամ**՝

```
Set-ExecutionPolicy -Scope CurrentUser RemoteSigned
irm get.scoop.sh | iex
scoop install git python gh
```

> *Քո սեփական Mac-ի կամ Linux-ի նոութբուքն է: Արդեն պատրաստ ես — անցիր Setup 1-ին:*

Ստուգիր քո գործիքակազմը

Ամեն տող պետք է տպի տարբերակի համարը — եթե որևէ մեկը բացակայում է, ասա մեզ:

```
git --version
python --version
gh --version
```

Հետո անցիր քո **home** թղթապանակ, որտեղ կլինեն քո նախագծերը:

```
cd ~        # your C:\Users\<you> folder
```

Խրվե՞լ ես: `setup/WINDOWS.md`-ն ունի լուծումները — «scripts disabled», Python-ը PATH-ում չէ, PowerShell 7:

Setup 1 of 3: GitHub

1. Ստեղծիր GitHub հաշիվ *(եթե չունես)*
2. **Fork** արա `EdikSimonian/telegram-pythonanywhere-bot`-ը — քո սեփական պատճենը, որ պետք է Օր 4-ի ավտո-տեղակայման համար
3. Clone արա քո fork-ը և տեղադրիր.

```
git clone https://github.com/<your-username>/telegram-pythonanywhere-bot.git
cd telegram-pythonanywhere-bot
.\make.ps1 install
```

Setup 2 of 3: Telegram բոտի token

1. Telegram-ում փնտրիր **@BotFather**
2. Ուղարկիր `/newbot`
3. Ընտրիր անուն, հետո՝ `bot`-ով վերջացող օգտանուն
4. Պատճենիր **token**-ը (նման է `7123456789:AAF...`-ի)

Setup 3 of 3: AI API բանալի

1. Գրանցվիր **cloud.cerebras.ai**-ում *(անվճար, առանց վարկային քարտի)*
2. Profile-ի պատկերակ → **API Keys** → **Create new API key**
3. Պատճենիր բանալին (նման է `csk-...`-ի)

Միացրու. `.env`

```
copy .env.example .env
```

Կարգավորիր երկու տող՝

```
TELEGRAM_BOT_TOKEN=<your BotFather token>
AI_API_KEY=<your Cerebras key>
```

Մնացածը թող այնպես, ինչպես կա:

> *Գաղտնիքներն ապրում են `.env`-ում: Երբեք կոդում, երբեք git-ում:*

Առաջին շփումը

```
.\make.ps1 run
```

```
Bot @your_bot_username starting in polling mode.
Send your bot a message on Telegram to try it out.
```

Գրիր քո բոտին: Դիտիր, թե ինչպես է փոխանակումը հայտնվում քո տերմինալում:

«Stateless mode»-ն սպասելի է: Հիշողությունը կգա Օր 4-ին:

Ապացուցիր, որ աշխատում է՝ ինժեների պես

```
.\make.ps1 test
```

- Ամբողջ հավաքածուն աշխատում է **օֆլայն**՝ ոչ API բանալիներ, ոչ ցանց
- **Նույն հավաքածուն** աշխատում է GitHub Actions-ում քո fork-ին ուղարկվող ամեն push-ի ժամանակ
- Կանաչ նշանը GitHub-ում = քո բոտի տրամաբանությունը դեռ աշխատում է

Քո առաջին կոդի խմբագրումը

`bot/handlers.py`:

```
@bot.message_handler(commands=["start"], func=is_allowed)
def cmd_start(message):
    bot.send_message(
        message.chat.id,
        "Hello! I'm your AI assistant...",
    )
```

Փոխիր ողջույնը → `Ctrl+C` → `.\make.ps1 run` → ուղարկիր `/start`:

Դու հենց նոր թողարկեցիր քո առաջին փոփոխությունը:

Հետագա սլայդներում՝ քո PowerShell փոխարկումը

Ամեն օր ունի համապատասխան **Windows սլայդաշար** — բայց եթե երբևէ Mac/Linux հրաման հայտնվի, երեք փոխարկում այն դարձնում են քոնը.

Սլայդը ցույց է տալիս	Դու գրում ես
`make <thing>`	`.\make.ps1 <thing>`
`./connect-claude-code.sh KEY`	`.\make.ps1 claude KEY`
`--persist` դրոշ	`-Persist`

Մնացած ամեն ինչը — `git`, `gh`, ֆայլերի խմբագրումը — **ճիշտ նույնն է**:

> *Օր 4-ի `.\make.ps1 deploy-pa`-ն պահանջում է PowerShell 7՝ `scoop install pwsh`:*

Այսօր → Վաղը

Այսօր բոտն աշխատում է քո համակարգչի վրա և պատասխանում է որպես ընդհանուր օգնական:

Վաղը. նախագծի ամենահզոր մեկ տողը՝ **համակարգային հուշումը**: Քո բոտը բնավորություն է ստանում: