

Tehisarupõhine tarkvaraarendus ja agentide juhtimine algajatele

Töövihik

macOS

Pi ja Groq macOS-i Terminalis



Eesmärk / Piirid / Väike samm
Kontroll / Otsus / Jätkamine

2026

Sisukord

1. Töövihiku kasutamine	2
2. Maci töökoht ja tegelikud õigused	2
3. Pi ja Node.js-i eeldused	3
4. Groqi ligipääs ilma võtit jagamata	4
5. Mudel, tasuta kvoot ja arvestus	5
6. Soovist kokkuleppe ja plaanini	6
7. Ühe faili loomine	7
8. Esimese tulemuse nähtav kontroll	8
9. Ettevalmistatud vea leidmine	9
10. Üks piiritletud parandus	10
11. Korduskontroll ja vastuvõtuotsus	11
12. Peatamine ja taastamise proov	12
13. Kontrollitud projektimärke	13
14. Uus seanss, sama projekt	14
15. Oma väike projekt ja kasutuspiir	15
16. Tõrkeabi, lõppkontroll ja allikad	16

1. Töövihiku kasutamine

See on õpiku „**Tehisarupõhine tarkvaraarendus ja agentide juhtimine algajatele**” eraldi praktiline töövihik macOS-ile. Õpik selgitab põhimõtteid; siin rakendad neid Pi ja Groqi mudeliteenusega. Sina määrad vajaduse, piirid ja kontrollid; agent valib tehnilise teostuse. Koodi lugemist, seletamist ega käsitsi muutmist ei eeldata.

Failid kaasa. Salvesta kaasasolev praktiline-toovihik-macos.zip ja paki see Downloads-kausta lahti. Kaustas praktiline-toovihik-macos on juhend, failid ja pildid. PDF-i lugemine ja antud juhtumite arutelu toimivad võrguta. Tegelik Pi/Groqi töö vajab internetti; HTML-näidete avamiseks on vaja varem salvestatud faile. Kui ZIP või näidisfailid puuduvad, küsi neid töövihiku andjalt: tühi failid kaust neid ei loo. See pole agendi töökaust ega kaitstud taastekoopia. PDF üksi ei sisalda käivitavat abivahendit.

Veebilingid: Pi <https://pi.dev>; Groq <https://console.groq.com>.

Õpitee: töökoht ja tasuta ligipääs → kokkulepe → loomine → nähtav kontroll → ettevalmistatud viga → parandus ja korduskontroll → peatamine ja taaste → projektimärke ja uus seanss. Lõpus kavandad oma väikese projekti.

2. Maci töökoht ja tegelikud õigused

Seos õpikuga: peatükid 1–2. Ava Spotlight (Command+Space), kirjuta Terminal ja vajuta Enter. Kasutame macOS-i Terminali, mitte Windowsi käsuviipa ega Linuxi alamkeskkonda. Ära kirjuta käsu ette siin näidatud selgitust ega Terminali viiba kasutajanime.

Määra failide asukoht. Käsk cd vahetab kausta, pwd näitab asukohta ja muutujasse ABIFAILID jääb selle akna jaoks abifailide täielik rada:

```
cd "$HOME/Downloads/praktiline-toovihik-macos/failid" && pwd
```

Kui kaust puudub, otsi lahtipakitud failid Finderist ja lohista see Terminalis cd järel käsureale; rada lisatakse sinu eest. **Jätka alles eduka kaustavahetuse järel**, kui kuvatud asukoht on õige ja Finderis on abifailid olemas:

```
ABIFAILID="$PWD"  
TOO="$HOME/pi-toovihik-macos"  
mkdir "$TOO" && mkdir "$TOO/loomine" "$TOO/parandus" "$TOO/taaste"  
ARVESTUS="$TOO/arvestus.json"
```

Kui mkdir ütleb, et töökaust on juba olemas, **peatu**, vaata sisu üle ja vali uus nimi. Ära tühjenda vana kausta. Muutujad kehtivad ainult selles Terminali aknas; uues aknas määra need uuesti olemasolevate kaustade järgi.

Kaust ei ole liivakast. Tavaline Pi saab kasutada sinu konto õigusi. Siin käivitame Pi-d kaasasoleva piiratud-pi.mjs abil: agendile antakse ainult ühe index.html faili tööriistad; Bash, koodi käivitamine, laiendused ja teiste failide lugemine pole saadaval. Abivahend kasutab Pi SDK-d, mitte teisti agent. See kitsendab mudeli tööriistateid, kuid pole üldine OS-i turvaeraldus ega kaitse pahatahtliku Pi paigalduse eest. Kasuta ainult usaldatud abifaile.

Taastekoopia ja arvestus jäävad valitud muutmiskaustast väljapoole. Sama Maci konto teine kaust üksi pole õigustepiir; siin takistab ligipääsu abivahendi failivalvur. Ära asenda seda harjutuse kestel piiramata pi käsuga. Üks faile muutev vahend korraga; ühised failid pole ühine vestlusmälu.

3. Pi ja Node.js-i eeldused

Vajad lubatud Maci, Terminali, brauserit, Node.js-i vähemalt 22.19 ja Pi-d. Kontrolli Terminalis; need käsud ei saada ülesannet mudelile:

```
node --version
npm --version
pi --version
```

Ootus: Node vähemalt 22.19, npm ja Pi annavad versiooninumbri. 18.10.2026 koostajakatse kasutas macOS 10.15.8 Inteli, Node 22.22.3 ja Pi 1.3.0. See on varasem katsekeskkond, mitte soovitus kasutada vana macOS-i ega Apple Siliconi või uuema Pi abivahendiühilduvuse kinnitus. Kui abivahend annab SDK-törke, peatu ja küsi uuendatud abifaile; ära eemalda valvureid.

Kui Pi puudub ja sul on paigaldusluba, pakub ametlik juhend järgmist teed:

```
curl -fsSL https://pi.dev/install.sh | sh
```

See **laadib ja käivitab võrgust programmi**, mitte ei kontrolli versiooni. Loe enne ametlikku juhendit. Paigaldaja võib lisada Node.js-i ja muuta kasutajakeskkonda. Ära lisa törke tõttu `sudo` käsku. Olemasoleva sobiva Node.js-i puhul on alternatiiv `npm install -g --ignore-scripts @earendil-works/pi-coding-agent`. Vali üks paigaldusviis, mitte mõlemad; ava uus Terminal ja korda versioonikontrolli. Koostaja kasutas olemasolevat Pi-d; uut paigaldust ega konto loomist ei katsetatud.

macOS • eelduste tegelik väljund

Kohalikus brauseris kuvatud päris kontrolliväljund; mitte Terminali foto.

```
$ sw_vers
ProductName:    Mac OS X
ProductVersion: 10.15.8
BuildVersion:   19H2036
$ node --version
v22.22.3
$ npm --version
12.2.0
$ pi --version
1.3.0
$ pi auth check --provider groq --json
{"status":"ready","provider":"groq","authType":"api_key"}
$ pi --list-models groq/openai/gpt-oss-120b
provider model                context max-out  thinking  images
groq      openai/gpt-oss-120b 131.1K  65.5K    yes       no
```

Maci tegelikud kontrolliväljundid, kohalikus brauseris kuvatud; mitte Terminali foto

Pi paigaldus ei tõenda tasuta kvooti ega ligipääsupiiri. Kui käsk puudub või Node on liiga vana, lahenda eeldus enne harjutust; agent ei paigalda tööriistu.

4. Groqi ligipääs ilma võtit jagamata

Pi käitab agenti; **Groq** pakub mudeliteenust. Ära aja teenust segi Groki või teise pakkujaga. Sihtrühm on 18+; kontrolli ise konto sobivust, kehtivaid tingimusi ja andmekasutust. Vajad enda või kooli lubatud **Free Plan** ligipääsu; kooli luba ei asenda teenusetingimusi. Harjutuseks ei osteta krediiti ega lisata maksekaarti. Ligipääsuta tee mõtteülesandeid, mitte agenditööd. API-võti on ligipääsutunnus, mitte ülesande osa. Ära pane seda HTML-i, juhisesse, projektimärkmesse, pilti ega veaaruandesse.

Oma lubatud konto võtmeid haldad ise aadressil <https://console.groq.com/keys>. Kooli võtme puhul järgi kooli korda. Ära kasuta võõrast avalikku näidisevõtit. Koostaja ei läbinud sinu eest konto- ega võtme loomise kasutajaliidest.

Kui Groqi ligipääsu pole Pi-s veel seadistatud, alusta ilma tööriistadeta:

```
pi --no-tools --no-approve --no-extensions --no-mcp --no-context-files --no-session  
--offline
```

Ära saada veel ülesannet. Vali Pi sees autentimine:

```
/login groq
```

Sisesta võti **ise** ainult selleks mõeldud autentimisväljal. Ära kirjuta võtit käsu argumendiks ega palu agendil võtmefaili leida. Välju Pi-st tühjal sisestusreal Control+D-ga ja kontrolli Terminalis:

```
pi auth check --provider groq --json
```

Oodatav mittesalajane vastusekuju:

```
{"status":"ready","provider":"groq","authType":"api_key"}
```

ready näitab autentimisvalmidust, mitte konto tasuta plaani või edukat mudelipäringut. Ära lisa **--credentials** lippu ega kasuta võtit printivaid käske. **--offline** peatab Pi automaatse võrgutöö, mitte sinu antud mudeliülesande. Lekkinud võti tuleb väljastaja juures kehtetuks teha; sõnumi kustutamisest ei piisa.

5. Mudel, tasuta kvoot ja arvestus

Seos õpikuga: peatükk 10. Valitud mudel on Groqi openai/gpt-oss-120b, mitte suvaline sama nimega mudel teisel teenusel.

```
pi --list-models groq/openai/gpt-oss-120b
```

Groqi avalik limiidijuhend loetles 18.10.2026 selle mudeli Free Plan'is. See pole tänase saadavuse kinnitus. Vaata enne päringut üle praegune limiidijuhend, oma konto tasuta plaan, mudeli tööriistatugi ja alles olev kvoot. Kataloogi otsing on ligikaudne: kontrolli pakkujat ja mudeli täpset nime. Mudelil on ka tasuline tokenihind: **kataloogihind pole sinu konto arve**. Ära mine Developer-plaanile ega vali tasulist asendust. Kvooditörke korral peatu.

Abivahendi ülempiir on **6 käivitust ja 12 päringut KOKKU**, iga käivitus kuni **900 sekundit / 6000 väljundtokenit**, kulu 0 €. Token on mudeli töödeldav tekstiühik, mitte tingimata sõna. Andmed on ainult väljamõeldud. Loomisel on lubatud `write`, parandusel `read/edit/write`, uue seansi arutelul tööriistu pole. Väiksema kokkulepitud piiri täitumisel katkestad ise.

Täida enne **iga** loomise, paranduse või uue seansi käivitust kuus rida:

Ülesanne ja kuupäev: _____
Groqi tasuta plaan / kvoot ja kontrolli aeg: _____
Töökaust / lubatud fail, tööriistad ja andmed: _____
Alles olevad käivitused / päringud: _____
Selle sammu aeg / väljundtokenid; kulu: 0 €: _____
Piiride jälgimise ja rakendamise viis: _____

- ☐ Oskan Control+C-ga peatada ja töö lõppu kontrollida (lk 12).
- ☐ Kaitstud toimiva näidisfailiga taasteproov on tehtud (lk 12).

Ava selle kokkuleppe arvestus **üks kord**:

```
node "$BIFAILID/piiratud-pi.mjs" alusta "$ARVESTUS" --tasuta
```

--tasuta tähendab sinu kinnitust, mitte arvelduskonto automaatkontrolli. Abivahend jõustab ülalolevaid ülempiire; ühe vastuse piir on kuni 4000 tokenit. Groqi arutlusrežiim on madalal tasemel, mitte välja lülitatud. Üks käivitus võib teha mitu päringut; automaatsed kordused ja varumudel on keelatud. Märgi ise ka enne valvuri käivitumist nurjunud katsed. Pane taimer käima ja jäta aega brauserikontrolliks. PASS jäljes tähendab abivahendi töö lõppu, mitte nähtava tulemuse vastuvõttu. Piiril või esimesel tõrkel peatu. Uus katse vajab uut luba ja alles oleva eelarve kontrolli; sama tööploki arvestust ei nullita.

Pi + Groq · tegeliku paranduse tööriistajälg

Kohalik brauseris kuvatud päris kontrolliväljund; mitte Terminali foto.

```
Pi 1.3.0 | Groq / openai/gpt-oss-120b | parandus | tööriistad ainult index.html
Mudelipäring 3/12; väljund kuni 4000 tokenit.
Tööriist: read index.html
Tööriistatulemus: OK
Mudelipäring 4/12; väljund kuni 4000 tokenit.
Tööriist: edit index.html
Tööriistatulemus: OK
Mudelipäring 5/12; väljund kuni 4000 tokenit.
```

```
Töö lõppenud. Päringuid selles käivituses 3; väljundtokeneid 366. Kataloogihind
ei ole konto arve.
```

Groqi tegeliku paranduse päringute ja failitööriistade jälg; kohalik kuva, mitte Terminali foto

6. Soovist kokkuleppe ja plaanini

Seos õpikuga: peatükid 3–5. Matkale võib paberloend olla lihtsam. Siin teeme kohaliku rakenduse selleks, et õppida agendi juhtimist, mitte sellepärast, et iga telk vajaks tehisaru heakskiitu.

Järgnev on **näidisdialoog**, mitte juba tehtud mudelikatse:

Sina: Tahan õppimiseks väikest matkavarustuse loendit.
Esialgu täpsustame; faile ära muuda.
Agent: Kas kirjed peavad pärast sulgemist säilima?
Sina: Ei. Kasutan ainult väljamõeldud telki ja lampi.
Agent: Kas märkimine peidab või kustutab rea?
Sina: Ei. Muutub ainult valitud rea märges, ka tagasi.
Agent: Kas vajad kontot, jagamist või pakkimata vaadet?
Sina: Mitte selles sammus. Ainult lisamine ja märkimine.

Kokkulepe: üks `index.html`; lisamine nupuga ja Enteriga; tühi sisend ei lisa rida. Sama nimega esemed on eraldi. `<lamp>` kuvatakse tekstina. Alguses ja taasavamisel on loend tühi. Lehel on mittesäilimise teade. Salvestus, server, konto, välised varad ja avaldamine jäävad välja.

Arendamise ja kasutamise vahe. Rakendus peab toimima kohalikult, kuid Pi saadab juhise ja tööks loetud sünteetilise faili Groqi kaugmudelile. Kohalik fail ei tähenda, et arendamisel võrku ei kasutata või teenus midagi vahel ei säilita. Jaga vaid kokkulepitud sünteetilist materjali.

Nähtav plaan: kontrolli lähteolukorda → üks teostussamm → lubatud kontrollid → üleandmine. Ära küsi mudeli varjatud mõttekäiku. Agent valib tehnilise teostuse; sina otsustad, kas tulemus vastab eesmärgile.

Mõtteülesanne: agent pakub kohe veebis jagamist. Kas see on pisiasi? **Vihje:** kellele tulemus ja andmed nähtavaks muutuvad? **Selgitus:** avaldamine laiendab ulatust ja vajab uut luba; seda ei lisata vaikimisi. Samuti vajavad uut nõusolekut paigaldus, uus teenus ja tasuline tegevus.

7. Ühe faili loomine

Loomiskaust peab olema tühi. Kaasasolev loomise-prompt.txt annab agendile järgmise ülesande; kokkulepe ja eelarve peavad olema enne kinnitatud:

Loo tühja töökausta ainult index.html: eestikeelne matkavarustuse loend.
Lisa esemeid nupuga ja Enteriga. Tühi või tühikutega sisend ei lisa rida.
Märkeruut muudab ainult oma rida ja saab ka märkimata olekusse tagasi minna.
Sama nimega esemed on eraldi read. <lamp> peab ilmuma tekstina, mitte HTML-ina.
Alguses ja pärast faili taasavamist on loend tühi; salvestamist pole.
Näita lehel teadet, et proovisitu ei säili pärast taasavamist.
Kasuta sisendi id=ese, nupu id=lisa, loendi id=esemed, teate id=teade;
iga ese on li, mille nime hoiab span ja märget input type=checkbox.
Tee lihtne loetav kujundus, mis mahub ka kitsasse vaatesse.
Ära lisa kustutamist, filtreid, kontot, serverit, salvestust, väliseid varasid,
rakenduse võrguühendusi, paigaldust ega avaldamist. Kasuta vaid sünteetilist sisu.
Kokkulepitud Groqi mudelipäringud on lubatud, muud andmeedastused mitte.
Ära loe saladusi, keskkonda, kodukausta, Pi seadistust ega kõrvalisi faile.
Alusta write-tööriistaga. Ära käivita genereeritud koodi ega muid protsesse.
Peatu tõrke, piiri või lisaloe vajaduse korral. Lõpus kuni kolm lauset:
mida tegid, mida päriselt kontrollisid ja et brauserikontroll jääb õppijale.

Tehnilised nimed selles mallis aitavad hoida näiteid võrreldavana; sa ei pea nende järgi koodi lugema ega muutma. Juhis ja abivahend jäävad muutmisalast välja. Käivita Terminalis:

```
node "$ABIFAILID/piiratud-pi.mjs" loomine "$T00/loomine" "$ARVESTUS" --tasuta
```

Ootus: jäljes on write index.html, tööriista edukas tulemus ja lõpetamine. Pelgalt vestlusse antud HTML-tekst ei ole tööriistaga faili loomine. Esimene tõrge või lisaloe vajadus lõpetab selle käivituse; ära korda pimesi.

8. Esimese tulemuse nähtav kontroll

Seos õpikuga: peatükid 3 ja 7. Oota agenditöö lõppu. Ava just loodud fail vaikebrauseris, mitte kaasasolev võrdlusnäide:

```
open "$T00/loomine/index.html"
```

Kontroll	Tegevus ja ootus
Lisamine	Lisa <code>telk</code> nupuga ja <code>lamp</code> Enteriga: kaks eraldi rida.
Märkimine	Märgi ainult <code>lamp</code> ; <code>telk</code> jääb märkimata. Võta lambi märgi tagasi.
Tühi sisend	Tühi ja tühikutega sisend ei lisa rida.
Korduv nimi	Lisa teine <code>lamp</code> ; selle märkimine ei muuda esimest.
Tekst	Lisa <code><lamp></code> : tekst säilib tervikuna, mitte HTML-elemendina.
Taasavamine	Sulge lehe vahekaart ja ava fail uuesti: loend on tühi, mittesäilimise teade nähtav.
Kitsas vaade	Tee brauseriaken kitsamaks: nupud ja tekst jäävad kasutatavaks.

Matkavarustuse loend

☐ `telk`☒ `lamp`

Proovisisu ei säili pärast lehe taasavamist.

macOS-is kontrollitud tegelik Groqi loodud fail: märgitud on ainult `lamp`

See on koostaja päris faili pilt, mitte sinu tulemuse tõend. Tee kontrollid oma koopial ja kirjuta nähtud tulemus üles. Töötav nupp ei tõenda täielikku turvalisust, ligipääsetavust ega kõigi võrguühenduste puudumist.

Kui kõik kokkulepitud kontrollid läbisid ja aktiivne töö on lõppenud, säilita muutmisalast väljas kontrollitud koopia:

```
cp "$T00/loomine/index.html" "$T00/toimiv-koopia.html"
```

Puuduva kontrolliga tulemus jääb pooleli. Ära kirjuta „kõik töötab“ üksnes agendi kokkuvõtte järgi. Võrdlusfail `failid/toimiv-index.html` on koostaja kontrollitud loomefail, mitte sinu loodud faili automaatne asendaja.

9. Ettevalmistatud vea leidmine

Ava valmis veaharjutus ilma agenti käivitamata:

```
open "$ABIFAILID/vigane-index.html"
```

1. Lisa tühja loendisse telk ja lamp.
2. Märki ainult lamp.
3. Vaata mõlema rea märget.

Ootus: ainult lamp märgitud. **Veafaili tulemus:** mõlemad märgitud. See viga lisati harjutamiseks ette; seda ei esitata Groqi juhusliku eksimusena.

Matkavarustuse loend

- ☒ telk
- ☒ lamp

Proovisisu ei säili pärast lehe taasavamist.

Ettevalmistatud macOS-i veaharjutus: lambi märkimisel muutusid mõlemad read

Kirjelda enne lahenduse lugemist: lähteolukord → tegevus → ootus → nähtud tulemus. Ühe esemega proov ei näitaks erinevust „muuda valitud rida” ja „muuda kõiki ridu” vahel.

Vihje: ära diagnoosi tehnilist põhjust. Ütle, kumb nähtav tulemus oli vale. **Lahendusnäide:** „Lisasin telgi ja lambi. Märkisin ainult lambi. Ootasin, et telk jääks märkimata, kuid mõlemad märgiti. Paranda ainult valitud rea märkimine; lisamine ja tühja sisendi käsitlemine peavad säilima.”

Paberil: kujutle kahte märkeruutu. Joonista oodatud ja nähtud seis ning kirjuta üks täpne tagasisidelause. See harjutab vea kirjeldamist, mitte ei tõenda brauseri- või agenditöö läbimist.

10. Üks piiritletud parandus

Seos õpikuga: peatükid 5 ja 8. Valmista ette eraldi paranduse töökaust. Originaal jääb alles; abivahend võib muuta ainult koopiat nimega `index.html`:

```
cp "$ABIFAILID/paranduse-lahteseis.html" "$T00/parandus/index.html"
```

Täida lk 5 tööleht uuesti: parandus vajab eraldi luba ja sama arvestuse alles olevat eelarvet, mitte loomisest vaikumisi jätkuvat luba. Tekst on tervikuna failis `parandus-prompt.txt`:

Paranda ainult töökausta `index.html`-i valitud rea märkimine.
Sammud: lisa telk ja lamp ning märgi ainult lamp.
Ootus: lamp märgitud, telk märkimata. Praegu muutuvad mõlemad.
Märkeruudu peab saama ka tagasi muuta; sama nimega read jäävad eraldi.
Säilita lisamine nupuga ja Enteriga, tühja sisendi tõrjumine, `<lamp>` tekstina, mittedäilimise teade, olemasolevad id-d ja alguses tühi loend.
Ära lisa funktsioone, salvestust, serverit, kontot, rakenduse võrguühendusi, paigaldust ega avaldamist. Muuda ainult `index.html`-i. Ära käivita koodi.
Kokkulepitud Groqi mudelipäringud on lubatud, muud andmeedastused mitte.
Ära loe saladusi, keskkonnamuutujaid, Pi seadistust ega kõrvalisi faile.
Peatu esimese tõrke, piiri või lisaloe vajaduse korral.
Lõpus kuni kolm lauset: muudatus, päriselt tehtud kontroll ja kontrollimata osa.

Käivita üks parandustöö:

```
node "$ABIFAILID/piiratud-pi.mjs" parandus "$T00/parandus" "$ARVESTUS" --tasuta
```

Ootus: agent loeb päris faili ja muudab seda `edit`- või `write`-tööriistaga. Tehniline valik on agendi teha. Kui vaja on teist faili või uut funktsiooni, peab ta peatuma. Edukas tööriistatulemus ei tähenda veel nähtava vea parandamist.

Mõtteülesanne: agent pakub pärast tõrget automaatselt „veel üht kiiret katset”. **Selgitus:** peatu ja vaata põhjus, failiseis ning alles olev luba üle. Uut käivitust ei tehta pimesi sama ebaõnnestunud juhiseiga; jätkamine pole kohustuslik.

11. Korduskontroll ja vastuvõtuotsus

Ava just parandatud koopia:

```
open "$T00/parandus/index.html"
```

Korda **kõiki lk 8 kontrolli**, mitte ainult lambi märkimist. See on regressioonikontroll: kas parandus rikkus midagi, mis enne töötas? Vaata Finderis kaust üle: kõrvalisi faile ei pidanud lisanduma.

Matkavarustuse loend

☐ telk☒ lamp

Proovisisu ei säili pärast lehe taasavamist.

Tegelik Groqi parandatud fail macOS-is: telk on märkimata, lamp märgitud

Võrdluseks on failid/parandatud-index.html; selle avamine ei asenda sinu agendi muudetud faili kontrolli. Koostaja katses kasutas agent `read` ja `edit` tööriistu ning eraldi Firefox'i korduskontroll läbis. Agent ei teinud neid brauserikontrolle ise.

Kontrollitud koopia / aeg: _____
Valitud rea märkimine ja tagasi: _____
Lisamine / tühi sisend / korduv nimi / tekst: _____
Taasavamine / kitsas vaade / kõrvalised failid: _____
Puuduva tõendiga nõuded: _____
Otsus: vastu võetud selles ulatuses / pooleli / taaste
Põhjendus: _____

Kvaliteedivärv: mida pidin kontrollima, mis tegelikult juhtus ja kas järgmine samm on põhjendatud? Roheline test toetab seda otsust, mitte ei tee seda sinu eest. Kui tühi sisend lisab nüüd rea, pole parandust tervikuna vastu võetud, kuigi lambi märkimine töötab. Piiri täitumisel peatud.

12. Peatamine ja taastamise proov

Seos õpikuga: peatükid 2 ja 8. Peatu tõrke, kvoodi-/makseteate, saladuse küsimise, lubamatu tegevuse või piiri täitumise korral. Ära anna uut ülesannet.

1. Katkesta selle abivahendi Terminali protsess **Control+C** abil, mitte Command+C-ga.
2. Oota protsessi väljumist. Kontrolli Activity Monitoris just enda käivitatud abiprotsessi lõppu; käsurea naasmine üksi pole piisav tõend.
3. Vaata failiseis üle ja alles siis taasta. Ära alusta paralleelset muutmist.

Siin pole agendil Bash'i ega alamprotsesside käivitamise tööriista. Kui protsess ei lõpe või sa ei suuda seda tuvastada, ära taasta; küsi abi, mitte ära lõpeta tundmatuid protsesse. Kohalik katkestus ei võta tagasi teenuses juba alanud päringut. Varasem koostaja SIGINT-proov läbis, kuid sinu enda käivituse lõppu tuleb ikkagi kontrollida.

Taasteproov enne esimest muutvat agenditööd. Kontrolli kaasasolevat `toimiv-index.html` faili lk 8 tabeliga. See jääb abivahendi muutmisalast välja:

```
TAASTE="$ABIFAILID/toimiv-index.html"
cp "$ABIFAILID/vigane-index.html" "$T00/taaste/index.html"
open "$T00/taaste/index.html"
```

Tuvasta lk 9 viga. Kui muutvat tööd ei käi, taasta proovikoopia:

```
cp "$TAASTE" "$T00/taaste/index.html"
cmp "$TAASTE" "$T00/taaste/index.html"
open "$T00/taaste/index.html"
```

`cmp` vaikib võrdsete failide korral; vea või erinevuse teatega taaste ei läbinud. Korda taastatud failiga **kõiki lk 8 kontrollle** ja märgi tulemus. Ebaõnnestunud prooviga ära alusta agenti. Pärast oma loomefaili kontrollimist korda proovi enda koopiaga, määrates `TAASTE="$T00/toimiv-koopia.html"`. Ära muuda taastealust ennast. Taastealuse kaitse kehtib siin ainult antud abivahendi failitööriistade suhtes, mitte kõigi sinu kontoga töötavate programmide suhtes.

Piir: rakenduse faili taastamine pole kasutajaandmete taastamine. Meie näite loend võib taasavamisel tühjeneda. Taaste ei võta tagasi saadetud infot, kulunud aega ega teenuse kvooti. Märga pole varukoopia.

13. Kontrollitud projektimärge

Seos õpikuga: peatükk 6. Märge kannab edasi kokkulepet, praegust failiseisu, tegelikke kontrollid, teadmata kohti ja järgmist lubatud sammu. See ei ole vestluse täielik mälu ega rakenduse varukoopia.

Kirjuta järgmine mall **tavalise tekstina** faili `projektimarge.txt` väljaspool muutmisala. TextEditis vali vajadusel Format → Make Plain Text (Shift+Command+T); ära salvesta RTF-ina ega lisa võtmeid.

PROJEKT: kohalik sünteetiline matkavarustuse loend
Praegune fail / kontrollitud koopia / aeg: _____
Juba teostatud eesmärk: lisamine ja valitud rea märkimine.
Välistatud: salvestus, server, konto, avaldamine.
Andmed: väljamõeldud; taasavamisel võib loend kaduda.
Tegelikud kontrollid ja tulemused: _____
Viga praegu / varasem viga eraldi: _____
Teadmata või tegemata: _____
Taastekoopia asukoht ja kaitse: _____
Taastamisproovi tegelik tulemus: _____
Aktiivse töö lõpu kontroll: _____
Kulunud käivitused / päringud / aeg / väljund: _____
Praegune luba: ainult märkme arutelu, mitte uus muutmine.
Järgmine võimalik samm ja selle loa vajadus: _____

Fail `failid/projektimarge.txt` on täidetud **näide**, mitte sinu kontrollide tõend. Enda märkme võid salvestada asukohta `$T00/projektimarge.txt`. Ära kirjuta teadmata asja asemel „korras”. Vana õnnestumine ei tõenda hiljem muudetud faili. Märget võib aidata koostada agent, aga sina võrdled väiteid oma vaatlustega.

Mõtteülesanne: „Kõik töötab, varukoopia on märge ja võrku ei kasutata, sest fail on kohalik.” **Vihje:** kolm väidet vajavad eri tõendeid. **Selgitus:** nimeta tehtud kontrollid täpselt, taasteks on vaja päris faili ning Pi/Groqi arendussuhtlus kasutab võrku. Failinimi ei taga automaatset lugemist: oluline tekst peab jõudma uue päringu konteksti.

Õpiku autor: Jaanus Pöial. Töövihiku koostamisel on kasutatud AI abi.

14. Uus seanss, sama projekt

Seos õpikuga: peatükid 1, 6 ja 9. Pi või teenus koostab igale pöördumisele konteksti; mudel ei mäleta vana tööd iseseisvalt. Failid pole ühine vestlusmälu. Ajaloo valimine või säilitamine sõltub vahendist ja teenusest; mittesäilitamist see töövihik ei taga.

Enne jätkamist peavad vana aktiivne töö olema lõppenud, failiseis üle vaadatud, märke ajakohane ning lk 5 tööleht uuesti täidetud. Uus arutelu vajab eraldi luba ja sama arvestuse alles olevat eelarvet: arutelugi kasutab mudelipäringut. Kaasasolev uue-seansi-prompt.txt küsib:

See on UUS seanss, varasemat vestlust ei anta.
Vasta nelja lühikese punktina eesti keeles:
(1) juba teostatud eesmärk;
(2) märkmes tegelikult kontrollitud seis;
(3) teadmata või kontrollimata kohad;
(4) üks järgmine väike muudatusteta kontrollisoovitus.
Ära käsitle olemasolevat funktsiooni uue soovina ega soovita seda uuesti teostada.
Ära loe või muuda faile, kirjuta koodi ega käivita kontrolle.
Ära nõua algajalt koodi mõistmist. Praegu on luba ainult märkme aruteluks;
soovitatud kontroll ei ole tehtud kontroll ja vajab eraldi otsust.

Salvesta **enda** täidetud märke lk 13 asukohta. Käivita:

```
node "$ABIFAILID/piiratud-pi.mjs" uus "$T00/parandus" "$ARVESTUS"  
"$T00/projektimarge.txt" --tasuta
```

Abivahend alustab uue mälusisese seansi ilma vana vestluseta ja ilma failitööriistadeta; kaasa läheb sinu antud märke. Enda arvutisse jäävad arvestus ja sünteetiliste vastuste logid. Ära jaga neid kontrollimata.

Kontrolli ka kokkuvõtet. Koostaja uue seansi vastused ei olnud laitmatud: esimene soovitas olemasolevat lisamist uuesti teostada; täpsem juhis kõrvaldas selle soovituse, kuid vastus lisas kontrollimata „tühjendamise” ning mainis välistatud konto/serveri kontrolli. Failid ei muutunud, kuid sisuline üleandmine jäi **osaliseks**. Õppimiskoht: ära võta ladusat kokkuvõtet tõendina. Paranda märkmes või vastuses vale eeldus; see ei anna uut muutmisluba.

15. Oma väike projekt ja kasutuspiir

Seos õpikuga: peatükid 10–12. Vali väike kohalik sünteetiline vajadus, näiteks väljamõeldud raamatute lugemisloend. Esimene samm võiks olla lisamine ja ainult valitud rea loetuks märkimine; salvestus ja jagamine jäävad välja.

Vajadus / üks väike nähtav tulemus: _____
Välistused / lubatud fail ja andmed: _____
Tavakasutuse tegevus ning ootus: _____
Erandolukorra tegevus ning ootus: _____
Andmed / kõrvalmõjud / õigused / turvaküsimused: _____
Tasuta ligipääs / uus eelarve / peatamine / taaste: _____
Vastuvõtu tingimus ja tõendid: _____

Töotsükkel: täpsusta → kinnita plaan ja luba → lase teostada → kontrolli → kirjelda puudus → luba piiritletud parandus → korda kontrolli → peatu ja vajadusel taasta → anna kontrollitud märgi uude seanssi üle. Kui oma projektis viga ei ilmne, ära väida, et parandamine toimus; vea leidmist harjuta lk 9-ga.

Vihje ja lahendussuund: lisa „Raamat A” nupuga ning „Raamat B” Enteriga. Märki ainult B ja tagasi; oota kahte alles jäävat rida. Katseta tühja sisendit ja korduvat A-d. Leppige enne kokku, et korduvad nimed on eraldi read.

Praegune abivahend loeb valmis matkaloendi ülesannet. Oma projekti kavanda siin paberil. Päril teostuseks küsi töövihiku andjalt eraldi abifailide komplekti, mis kasutab sinu kinnitatud tekstijuhiseid ja samu kontrollitud piire; praegune käsk sinu uut malli automaatselt ei loe. Ära muuda JavaScripti ega turvapiire. Tühjaks saanud arvestust ei asendata loata uuega. Uue ulatuse ja tasuta ligipääsu kontroll on eraldi otsus; siinne koostajakatse ei ole sinu projekti läbimise tõend.

Kasutusotsus: „sobib sünteetiliseks kohalikuks prooviks” ei tähenda „sobib pärisandmeteks või avaldamiseks”. Küsi, kelle materjali kasutad, mis õigusega, mida saadetakse teenusele ja millised kontrollid puuduvad. Ära lisa inimeste andmeid, makseid ega avalikku teenust. Kui kvoot lõpeb, võid oodata, vähendada uut ülesannet või lõpetada; liimitidest ei minda mööda. Teise vahendi või pakkuja proov on valikuline ja vajab oma piiride ülevaatust.

16. Tõrkeabi, lõppkontroll ja allikad

Tunnus	Ohutu järgmine samm
<code>pi</code> või <code>node</code> puudub	Ava uus Terminal; kontrolli ametlikku paigaldusjuhendit, mitte juhuslikke käske.
Kaust või abifail puudub	Vaata ZIP-i sisu Finderis; küsi puuduvad failid töövihiku andjalt. Kaustavahetuse tõrkega ära jätkata.
<code>not_ready</code> või <code>invalid</code>	Vaata ise autentimine üle; ära prindi ega jaga võtit.
HTTP 429, makseteade või tasuta kvoot puudub	Peatu; ei tasulist asendust, uusi kontosid ega automaatset kordust.
SDK-tõrge või valvur ei käivitu	Peatu; küsi sobiva Pi versiooniga kontrollitud abifaile. Ära kasuta piiramata <code>pi</code> käsku asendusena.
Arvestus on suletud või eelmine töö lõpetamata	Ära nulli faili. Kontrolli töö lõppu ja otsusta uus tööplokk eraldi.
Vale brauserivaade või failiseis ebaselge	Kontrolli asukohta, ava õige koopia; ära muuda ega taasta enne selgust.

Märgi ainult tehtud töö; pooleli tulemus on lubatud.

- ☐ Leppisin eesmärgi, failid, tasuta aluse ja eelarve enne kokku.
- ☐ Agent kasutas päris failitööriista; kontrollisin just selle tulemust.
- ☐ Leidsin ettevalmistatud vea ja andsin täpse tagasiside.
- ☐ Kordasin pärast parandust kõiki nähtavaid kontrole.
- ☐ Kinnitasin töö lõpu ja proovisin failikoopiaga taastamist.
- ☐ Uuendasin märget ja kontrollisin uue seansi väiteid.
- ☐ Kirjutasin teadmata kohad ja kasutuspiiri, mitte ainult „kõik töötab”.

Kirjandus ja juhendid

- Pöial, J. (2026). Tehisarupõhine tarkvaraarendus ja agentide juhtimine algajatele.** Peatükid 1–5 aitavad täpsustada eesmärki ja juhtida väikest sammu, peatükid 6–9 õpetavad projektimärget, kontrolli, parandust ja jätkamist; peatükid 10–12 käsitlevad ressursse, kasutuspiire ja oma projekti.
- Pi ametlikud juhendid:** <https://pi.dev>. Vaata siit paigaldamist, autentimist, mudeli valimist ja seansside kasutamist. Tööriistade õiguste ja turvalise käivitamise juhised aitavad mõista, miks töökaust üksi ligipääsu ei piira.
- Groqi mudelid:** <https://console.groq.com/docs/models>. Kontrolli mudeli täpset nime, võimeid ja hinnakirja. Kataloogikirje ei kinnita sinu konto tasuta ligipääsu ega mudeli sobivust igaks ülesandeks.
- Groqi kasutuspiirid:** <https://console.groq.com/docs/rate-limits>. Võrdle Free Plan'i piiranguid oma konto kvoodiga. Päringute ja tokenite limiidid võivad olla erinevad; piirini jõudes peatu, mitte ära vali tasulist asendust.
- Groqi tööriistakutsed:** <https://console.groq.com/docs/tool-use>. Juhend selgitab, kuidas mudel saab paluda tööriista kasutamist ja saada selle tulemuse tagasi. Siinses harjutuses käivitab lubatud failitööriistu Pi.
- Groqi andmekasutus:** <https://console.groq.com/docs/your-data>. Loe, milliseid kasutusandmeid säilitatakse ning millised erandid ja andmekontrollid kehtivad juhiste ja vastuste jaoks. Kontrolli enne teenuse kasutamist ka Groqi kehtivaid konto- ja kasutustingimusi.

Tehisarupõhine tarkvaraarendus ja agentide juhtimine algajatele

Töövihik

macOS

Üks väike kohalik projekt.

Sina määrad eesmärgi ja piirid,

agent valib teostuse, sina kontrollid tulemust.



Pi / Groq / macOS-i Terminal