



ГАРЫН АВЛАГА · 01

Кодын ертөнц

Код бичихгүйгээр ойлгох. Юу юу байдаг вэ, юу нь юуг хийдэг вэ, тэднийг хэрхэн зөв захиалах вэ — зурагтайгаар, эхнээс нь.

Агуулга

Энэ ном код бичиж сургахгүй. Кодын ертөнцөд юу юу байдгийг, юуг нь юу гэж нэрлэдгийг танилцуулна — ингэснээр та хүссэн зүйлээ **зөв үгээр захиалж** чаддаг болно.

1	Код гэж юу вэ	Жор, заавар, хэл
2	Код хэрхэн ажилладаг вэ	Хэн уншиж биелүүлдэг вэ, алдаа гэж юу вэ
3	Ямар ямар зүйлс байдаг вэ	Хэрэгсэл, хэл, давхаргууд — нэр бүрийг нь
4	Claude-аар захиалж сурах нь	Промпт бичих — энэ номын гол бүлэг
5	Сервер гэж юу вэ	Хүсэлт, хариу, интернэтэд гаргах
6	Database гэж юу вэ	Хүснэгт, мөр, багана, холбоо
7	GitHub бол хамгийн чухал нь	Хадгалалтын цэг, түүх, танилцуулга
A	Хавсралт: Промптын загварууд	Хуулж ашиглах долоон загвар
B	Хавсралт: Үгсийн тайлбар	Гучин нэр, нэг мөрийн тайлбартай

Энэ ном хэнд зориулагдсан бэ

Код бичиж үзээгүй, бичих ч бодолгүй хүнд. Та программ бичихгүй — **захиална**. Гэхдээ захиалахын тулд юу захиалж байгаагаа мэдэх хэрэгтэй, тэр мэдлэгийг энэ ном өгнө.

1 Код гэж юу вэ

Хичээл 1 · Хамгийн эхний ойлголт

Код бол компьютерын яг дагаж биелүүлдэг заавар. Өөр юу ч биш. Ид шид ч биш, математик ч биш — дараалсан заавар.

Жор ба заавар

Хоолны жорыг бодоод үз. «Ус буцалга», «давс нэм», «арван минут чана» — дараалсан заавар. Хүн жорыг уншаад дутууг нь өөрөө нөхдөг: хэдэн грамм давс гэж бичээгүй ч чи мэдэж байгаа.

Компьютер нөхөхгүй. Хэдийг нь хэлээгүй бол асуухгүй, таахгүй — зогсоно.

Хүнд хэлэхэд

«Давс амтлаад нэмээрэй»

Хүн: амтлаад, хэрэгтэйг нь нэмнэ

→ Болсон

Компьютерт хэлэхэд

«Давс амтлаад нэмээрэй»

Компьютер: хэдийг нь? амт гэж юу вэ?

→ Зогсоно

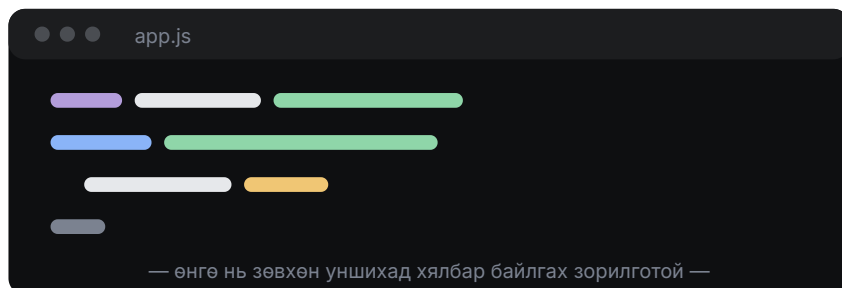
Зураг 1. Компьютер дутууг нөхдөггүй. Тиймээс код бол ухаалаг байхаас илүү **нарийн** байх тухай.

Энэ л чиний гол давуу тал болно

Промпт бичихэд яг ижил дүрэм үйлчилнэ. Хиймэл оюун хүнээс хамаагүй ухаантай таадаг ч, тодорхой хэлсэн зүйлийг үргэлж илүү зөв хийдэг. «Сайхан хуудас хийж өг» гэхээр таана, «цэнхэр өнгийн, гурван хэсэгтэй, дээр нь миний нэр байх хуудас» гэхээр хийнэ.

Код бол зүгээр л текст

Кодыг тусгай хэрэгсэл дээр бичдэг гэж боддог. Үгүй. Код бол **текст файл** — Notepad дээр ч бичиж болно. Ялгаа нь тэр текстийг уншиж, биелүүлдэг програм байдаг л оршино.



Зураг 2. Кодын файл ингэж харагдана. Өнгө нь кодын нэг хэсэг биш — засварлагч програм уншихад хялбар болгож будаж өгдөг.

Хэл гэж юу вэ

JavaScript, Python, Swift — эдгээр нь **хэл**. Хүний хэлтэй адил: үг, дүрэм, утга. Ялгаа нь дүрэм нь маш цөөхөн бөгөөд огт уян хатан биш.

Нэг хэл ойлгосон хүнд дараагийнх нь илүү хялбар болдог. Учир нь ойлголтууд нь бүх хэлэнд ижил — зөвхөн бичлэг нь өөр. Монгол, англи хоёрын аль алинд «өглөө» гэж хэлж болдогтой адил.

Хамгийн түгээмэл эндүүрэл

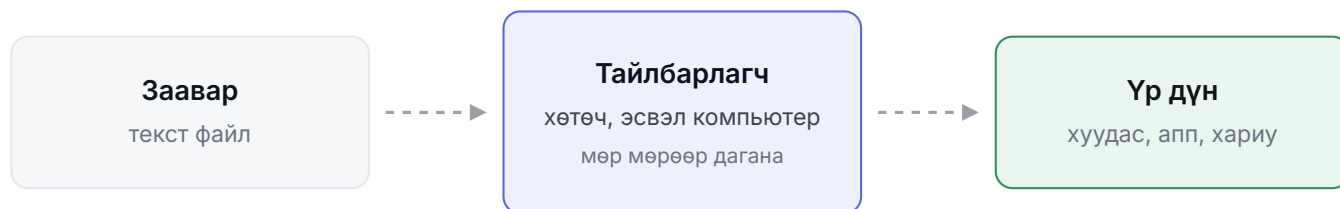
«Кодыг цээжлэх ёстой» гэж боддог.

Үгүй. Хэн ч бүгдийг цээжлээгүй — хэрэгтэй үедээ хайдаг. Цээжлэх ёстой зүйл бол **бодох арга**: том ажлыг жижиг алхам болгон хуваах, алхам бүрийг тодорхой хэлэх. Яг энэ чадвар нь промпт бичихэд чинь хэрэгтэй.

2 Код хэрхэн ажилладаг вэ

Хичээл 2 · Бичсэн зааврыг хэн биелүүлдэг вэ

Хэн нэгэн тэр текстийг уншиж, дагаж биелүүлдэг. Тэр «хэн нэгэн» нь бас л програм — түүнийг тайлбарлагч гэдэг.



Зураг 3. Текст → уншигч → үр дүн. Гурав хоорондоо ийм л холбоотой.

Програмын гурван хэсэг

Ямар ч програм гурван зүйлээс бүрдэнэ. Instagram ч, банкны апп ч ижил — зөвхөн олон давхар давтагдана.



Оролт

Хэрэглэгчийн бичсэн зүйл, дарсан товч, оруулсан зураг.



Боловсруулалт

Тооцоо, шалгалт, эрэмбэлэлт. Дотор нь юу болж байгаа нь.



Гаралт

Дэлгэц дээр гарч ирэх зүйл, хадгалагдсан өгөгдөл, илгээгдсэн мессеж.

Промпт бичихэд шууд хэрэгтэй

Юм захиалахдаа энэ гурвыг нь хэлээрэй: **юу оруулах вэ, юу хийх вэ, юу гарах вэ.** «Хэрэглэгч и-мэйлээ бичнэ (оролт), зөв эсэхийг шалгана (боловсруулалт), буруу бол улаан мессеж гарна (гаралт).» Ийм гурван өгүүлбэр нь хагас хуудас тайлбараас илүү ажилладаг.

Алдаа гэж юу вэ

Улаан мессеж бол бүтэлгүйтэл биш. Тэр бол компьютер чамд **аль зааврыг чинь дагаж чадаагүйгээ** хэлж байгаа хэрэг. Уншиж ойлгох шаардлагагүй — **хуулж аваад Claude-д тавь**. Тэр л хангалттай.



Зөв зан

Алдааны мессежийг бүтнээр нь хуулаад «энэ юу гэсэн үг вэ, яаж засах вэ» гэж асуух.



Буруу зан

Айж бүх зүйлээ устгаад эхнээс нь эхлэх. Алдаа нь хаана байгааг хэлж байхад.

Хоёр газар ажилладаг



Хөтөч дотор

Хэрэглэгчийн харж байгаа зүйл. Товч, өнгө, хөдөлгөөн.



Сервер дээр

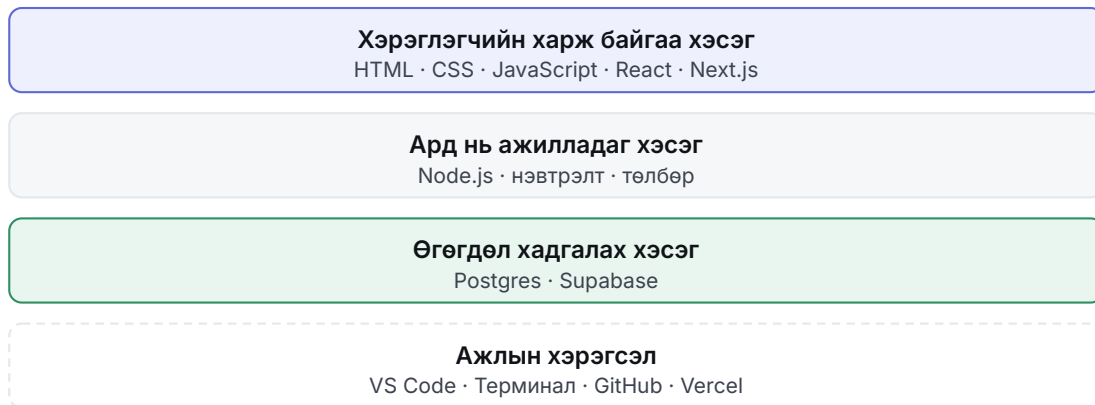
Хэрэглэгчийн харахгүй зүйл. Өгөгдөл, нэвтрэлт, төлбөр.

Энэ хоёрыг ялгаж мэдэх нь чухал: юм захиалахдаа «энэ нь хэрэглэгчид харагдах уу, эсвэл ард нь ажиллах уу» гэдгээ хэлж чадвал хариу нь илүү зөв гарна.

3 Ямар ямар зүйлс байдаг вэ

Хичээл 3 · Нэр бүрийг нь, нэг мөрөөр

Одоо бүгдийг нь ойлгох шаардлагагүй. Нэр нь танил болж, «энэ юуны төлөө байдаг юм бэ» гэдэг нь тодрох л хангалттай. Промпт бичихэд яг эдгээр нэрс хэрэгтэй болно.



Зураг 4. Дөрвөн давхарга. Дээрээс доош: харагддаг, ажилладаг, хадгалагддаг, барьдаг.

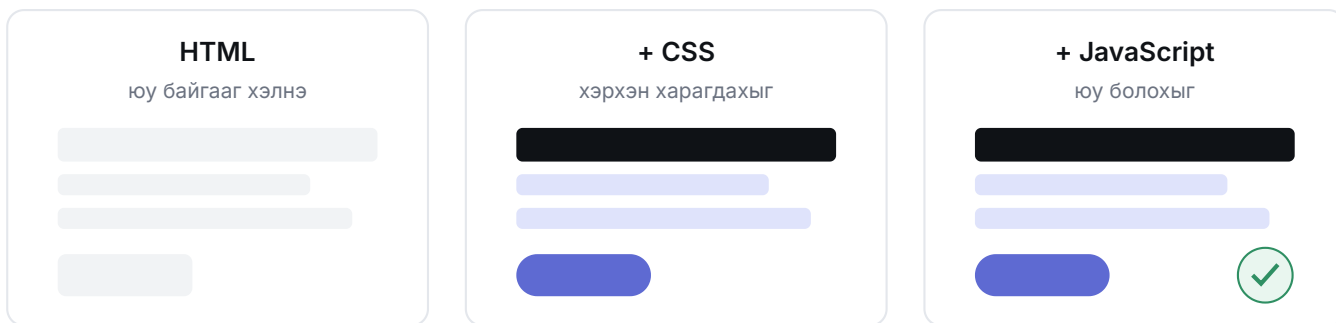
Бичих газар

- VS Code**
Кодын засварлагч. Үнэгүй. Файлуудыг нэг цонхонд барьж, өнгөөр ялгаж, алдааг зааж өгнө. Claude энэ дотор ажиллана.
- Терминал**
Тушаал бичдэг хар цонх. Айх зүйл байхгүй — юу бичихийг нь Claude хэлж өгнө, чи хуулж тавина.
- Хөтөч**
Chrome. Хийсэн зүйлээ энд хараад шалгана. **F12** дарахад ард нь юу болж байгааг харуулна.

Гурван хэл, гурван үүрэг

Вэб хуудас гурван зүйлээс бүрдэнэ. Гурвыг нь ялгаж мэдэх нь захиалахад хамгийн их хэрэгтэй мэдлэг:

- HTML**
Хуудсанд **юу байгааг** хэлнэ. Гарчиг, догол мөр, зураг, товч.
- CSS**
Тэдгээр нь **хэрхэн харагдахыг** хэлнэ. Өнгө, зай, хэмжээ, байрлал.
- JavaScript**
Юу **болохыг** хэлнэ. Товч дархад, өгөгдөл татахад, цонх нээхэд.



Зураг 5. Ижил хуудас, гурван давхарга нэмэгдэхэд. HTML нь агуулга, CSS нь хэлбэр, JavaScript нь амь.

Дээд давхаргууд

-  **React**
Хуудсыг жижиг, дахин ашиглах хэсгүүд болгон бүтээх арга. Нэг товчийг нэг удаа хийгээд хорин газар ашиглана.
-  **Next.js**
React дээр суурилсан бүтэн систем — хуудас, хаяг, серверийн тал бүгд дотроо. Энэ сургуулийн сайт үүгээр хийгдсэн.
-  **Expo**
Ижил мэдлэгээр iOS болон Android апп хийх боломж. Нэг ажлаар хоёр дэлгүүрт гарна.

Доод давхаргууд

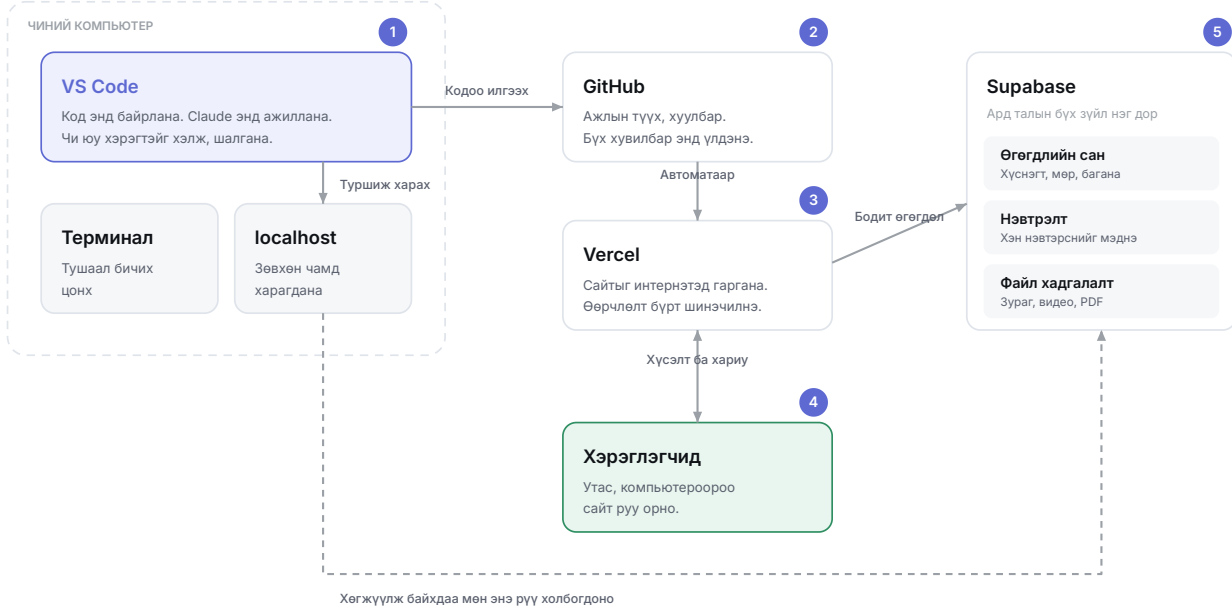
-  **Node.js**
JavaScript-ийг хөтчөөс гадна, компьютер дээр ажиллуулдаг. Серверийн тал үүн дээр тогтоно.
-  **Supabase**
Өгөгдлийн сан, нэвтрэлт, файл хадгалалтыг нэг дор өгдөг үйлчилгээ. Бүлэг 6-г үз.
-  **Git ба GitHub**
Кодын түүх, хуулбар, хамтын ажиллагаа. Бүлэг 7-г үз.
-  **Vercel**
Хийсэн сайтаа интернэтэд гаргах газар. GitHub-тай холбоод нэг товчоор.

Бүгд хэрхэн холбогддог вэ

Дээрх нэрсийг нэг зураг дээр тавибал ингэж харагдана. Нэг өөрчлөлт чиний компьютерээс эхлээд хэрэглэгч хүртэл ингэж аялдаг:

Бүгд хэрхэн холбогддог вэ

Нэг өөрчлөлт чиний компьютерээс хэрэглэгч хүртэл



Зураг 6. **1.** Чи VS Code дотор юу хэрэгтэйг хэлнэ, localhost дээр туршиж харна. **2.** Кодоо GitHub руу илгээнэ. **3.** Vercel үүнийг автоматаар аваад интернэтэд гаргана. **4.** Хэрэглэгчид сайтыг үзнэ. **5.** Өгөгдөл, нэвтрэлт, файл нь Supabase дээр — хөгжүүлж байхад ч, бодит ажиллагаанд ч ижил газар.

Бүгдийг нэг дор бүү оролд

Эдгээрийг зэрэг сурах гэж оролдвол нэг нь ч үлдэхгүй. Чамд хэрэгтэй нь **нэр бүрийг нь таних** чадвар — яг юуны төлөө байдгийг нь мэдэх. Захиалахдаа зөв нэрийг нь хэлж чадвал хангалттай.

4 Claude-аар захиалж сурах нь

Хичээл 4 · Энэ номын гол бүлэг

Чи код бичихгүй. Чи **захиална**. Тиймээс чиний жинхэнэ ур чадвар нь бичих биш, **тодорхой хэлэх** чадвар болно. Энэ бүлэг тэрийг заана.

Хоёр асуулт, хоёр өөр үр дүн

Муу

«Надад сайт хийж өг.»

Ямар сайт? Хэнд зориулсан? Юу байх ёстой?
Хариу нь таамаг болно.

Сайн

«Гоо сайхны салоны нэг хуудас сайт хэрэгтэй. Дээр нь салоны нэр, доор нь үйлчилгээний жагсаалт үнэтэй нь, хамгийн доор утас ба байршил. Утсан дээр зөв харагдах ёстой. Өнгө нь цайвар ягаан.»

Хариу нь эхний оролдлогоор ойрхон гарна.

Сайн промптын дөрвөн хэсэг

Энэ дөрвийг бичвэл хариу нь бараг үргэлж зөв гардаг. Дараалал нь ч чухал.

🕒 1. Юу хийх гэж байна вэ

Юу барих гэж байгаагаа нэг өгүүлбэрээр.

«Хоолны газрын цэсийг харуулах хуудас хийх гэж байна.»

📄 2. Хэнд зориулсан, ямар нөхцөлд

Хэн үзэх вэ, юу чухал вэ.

«Ихэнх хүн утсаараа орно. Интернэт удаан байж магадгүй. Монгол хэл дээр.»

📋 3. Юу байх ёстой — жагсаалтаар

Хүсэж байгаа зүйлээ дугаарлаж бич.

«1. Дээр нь газрын нэр ба зураг. 2. Цэсийг ангилалаар нь, тус бүр үнэтэй. 3. Доор нь утас, хаяг, ажиллах цаг.»

✓ 4. Юуг нь хийх шаардлагагүй

Хязгаараа хэлэх нь нэмэхээс чухал.

«Захиалга авах хэсэг хэрэггүй. Нэвтрэх хэрэггүй. Одоохондоо зөвхөн харуулах хуудас.»

Яагаад дөрөв дэх нь хамгийн чухал вэ

Хязгаар хэлээгүй бол хиймэл оюун «магадгүй хэрэгтэй байх» гэж бодоод илүү зүйл нэмнэ. Тэгээд чи ойлгохгүй зүйл олонтой үлдэнэ. **Хэрэггүй зүйлээ хэлэх нь хэрэгтэйгээ хэлэхээс дутуугүй ач холбогдолтой.**

Алдаа гарвал

Алдааны мессежийг **бүтнээр нь** хуулж тавь — эхний мөрийг нь биш, бүгдийг нь. Дараа нь юу хийж байхад гарснаа хэл.



«Товч дархад энэ алдаа гарлаа: [мессежийг бүтнээр нь]. Юу болоод байна вэ, яаж засах вэ?»

Хариу нь бараг үргэлж зөв гарна.



«Ажиллахгүй байна.»

Юу ажиллахгүй байна вэ? Хаана? Ямар мессеж гарав? Гурван эргэлт үрэгдэнэ.

Дахин дахин сайжруулах

Эхний хариу төгс байх шаардлагагүй. Промпт бичих гэдэг нь **ярилцлага**, нэг удаагийн захиалга биш.



«Энэ хэсгийг арай том бол.»

Жижиг засварыг тусад нь хэл.



«Яагаад ингэж хийсэн бэ?»

Ойлгохгүй зүйлээ асуу. Тайлбарлана.



«Энэ ямар тохиолдолд эвдрэх вэ?»

Дууссаны дараа заавал асуу.



«Анхан шатны хүнд ойлгомжтой тайлбарлаж өгөөч.»

Ойлгохгүй хариу ирвэл.

Хэзээ ч хийж болохгүй гурван зүйл

Юу	Яагаад
Нууц үг, түлхүүр, картын дугаар тавих	Тэдгээр нь ярианд үлдэнэ. Нууц мэдээллийг хэзээ ч хуулж бүү тавь.
Өөр хүний хувийн мэдээлэл тавих	Хэрэглэгчийн жагсаалт, утасны дугаар, и-мэйл — зөвшөөрөлгүйгээр болохгүй.
Шалгалгүй нийтлэх	Ажиллуулж үз. Зөв ажиллаж байна уу? Огт өөр зүйл хийж байна уу?

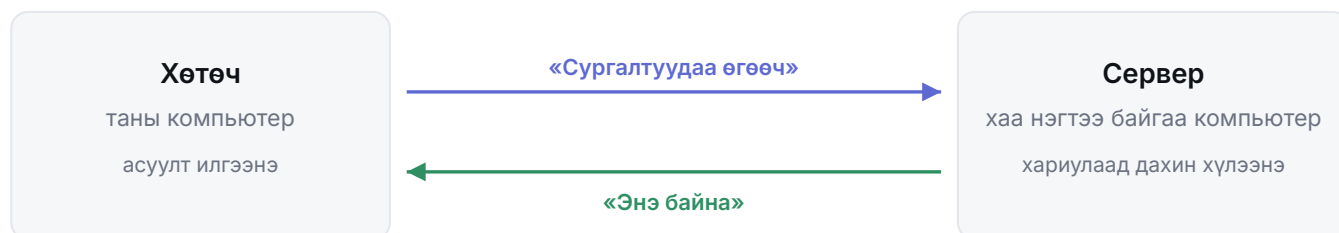
Чиний ажил юу вэ

Код бичих нь биш. **Юу хэрэгтэйг шийдэх, тодорхой хэлэх, гарсан үр дүнг шалгах** — энэ гурав. Хиймэл оюун хурдан бичнэ, гэхдээ юу бичихийг шийдэхгүй. Тэр нь чиний ажил.

5 Сервер гэж юу вэ

Хичээл 5 · Хүсэлт хүлээж суудаг компьютер

Сервер бол тусгай ид шид биш. Асуулт хүлээж суудаг компьютер. Чи асуухад хариулна — тэгээд дахин хүлээнэ.



Зураг 6. Нэг хүсэлт, нэг хариу. Facebook ч, банк ч ижил зарчмаар — зөвхөн олон мянган хүсэлт зэрэг ирдэг.

Домэйн нэр

`jarvis.mn` гэх мэт хүн уншиж чадах хаяг.

Интернэт дотор компьютерууд тоогоор ялгагддаг. Домэйн нэр бол тэр тоог нэрээр нь санахад хялбар болгосон зүйл — утасны номтой адил.

localhost

«Энэ компьютер» гэсэн үг.

Ажиллаж байхдаа чи өөрийнхөө компьютер дээр сайтаа асаадаг. Тэр үеийн хаяг нь `localhost` болно.



localhost холбоосыг найздаа бүү явуул

Тэр хаяг нь **түүний** компьютерыг заана, чинийхийг биш. Юу ч нээгдэхгүй. Бусдад харагдахын тулд интернэтэд байршуулах хэрэгтэй.

Урд тал, ард тал



Урд тал

Хэрэглэгчийн харж байгаа хэсэг. Товч, өнгө, зураг.



Ард тал

Өгөгдөл, нэвтрэлт, төлбөр. Хэрэглэгч хэзээ ч харахгүй.

Яагаад хоёр хэсэг байх ёстой вэ? Учир нь хөтөч дотор байгаа бүх зүйлийг хэрэглэгч харж чадна. Нууц зүйлийг тэнд тавьбал нууц биш болно.

Байршуулах — deploy

Хийсэн сайтаа интернэтэд гаргах.

Vercel зэрэг үйлчилгээ үүнийг нэг товчоор хийдэг: GitHub-тай холбоод, өөрчлөлт хийх бүрт шинэ хувилбар нь өөрөө гарна. Энэ мөчөөс эхлэн дэлхийн хэн ч чиний сайтыг нээж чадна.

6 Database гэж юу вэ

Хичээл 6 · Хаагаад нээхэд үлддэг өгөгдөл

Хуудсаа хаагаад дахин нээхэд бичсэн зүйл чинь алга болдог. Учир нь тэр зөвхөн санах ойд байсан. Үлдээхийн тулд өгөгдлийн сан хэрэгтэй.

Excel-ийн хүснэгт гэж бодоорой



Хүснэгт

Нэг төрлийн зүйлсийн жагсаалт. Жишээ нь бүх хэрэглэгч, эсвэл бүх бүтээгдэхүүн.



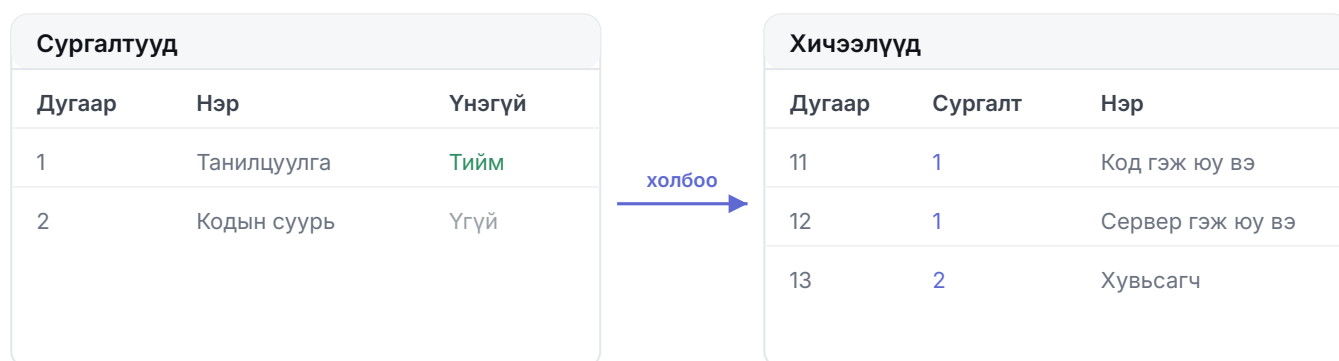
Мөр

Нэг зүйл. Нэг хэрэглэгч, нэг захиалга.



Багана

Тэр зүйлийн нэг шинж. Нэр, утас, бүртгүүлсэн огноо.



Зураг 7. Хичээл бүр өөрийн сургалтын дугаарыг авч явна. Ингэснээр «энэ сургалтын бүх хичээл» гэж асуух боломжтой болно.

Энэ холбоо нь зүгээр нэг тохь тух биш — байхгүй сургалтад хичээл нэмэхийг сан өөрөө татгалзана. Өгөгдөл эвдрэхээс сэргийлж байгаа хэрэг.

Захиалахдаа ингэж хэлээрэй

«Хэрэглэгч, захиалга хоёр хүснэгт хэрэгтэй. Захиалга бүр аль хэрэглэгчийнх болохыг мэддэг байх ёстой.» Ийм хоёр өгүүлбэр нь өгөгдлийн сангийн бүтцийг зөв гаргахад хангалттай.

Хаана байрлах вэ



Postgres

Хамгийн өргөн хэрэглэгддэг өгөгдлийн сан. Үнэгүй, хүчирхэг.



Supabase

Postgres-ийг үүлэн дээр ажиллуулж, нэвтрэлт, файл хадгалалтыг дээр нь нэмж өгнө.

Хоёр зүйлийг үргэлж санаж бай

Нэг: нууц үгийг задгай хадгалж болохгүй. Бэлэн нэвтрэлтийн системийг ашигла — өөрөө бүү зохио.

Хоёр: «Энэ товчийг зөвхөн админ харна» гэж нуух нь хамгаалалт биш. Жинхэнэ шалгалт нь сервер дээр байх ёстой. Захиалахдаа «эрхийн шалгалтыг сервер тал дээр хий» гэж заавал хэлээрэй.

7 GitHub бол хамгийн чухал нь

Хичээл 7 · Ажлын түүх, хамгаалалт, танилцуулга

Код бичихгүй ч гэсэн үүнийг мэдэх ёстой. Учир нь чиний хийсэн бүхэн энд хадгалагдаж, эндээс л бусдад харагдана.

Git

Кодын цагийн машин.

Ажиллаж байсан үеийн хувилбарыг хадгалж, хүссэн үедээ буцаж очих боломж өгдөг. Хоёр цаг ажиллаад бүхнээ эвдчихвэл — өмнөх цэг рүүгээ буцна.

GitHub

Тэр түүхийг интернэтэд байрлуулах газар.

Компьютер чинь эвдэрсэн ч ажил чинь тэнд байна. Бусад хүн хараад тусалж чадна. Ажил олгогч чиний түүхийг үзнэ.

✓ Commit — хадгалалтын цэг

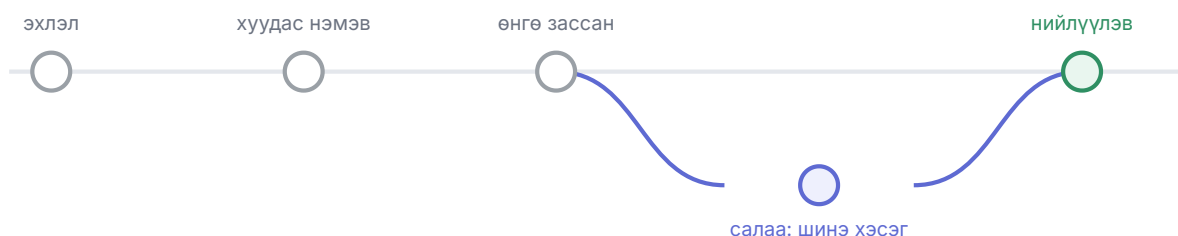
«Энэ хүртэл ажиллаж байсан» гэж тэмдэглэсэн мөч.

Цэг бүр нь тайлбартай: хэзээ, юуг, хэн өөрчилсөн. Гурван сарын дараа «энэ яагаад ингэж болсон юм бэ» гэхэд хариулт нь тэнд байна.

Салаа — branch

Ажиллаж байгаа зүйлээ эвдэлгүйгээр шинэ зүйл туршиж үзэх зам.

Туршилт нь бүтвэл үндсэн зам руугаа нийлүүлнэ. Бүтэхгүй бол салаагаа хаяад л болоо — үндсэн ажилд юу ч болоогүй.



Зураг 8. Дугуй бүр нэг хадгалалтын цэг. Салаа нь тусдаа явж, бэлэн болмогц буцаж нийлнэ.

Сайн тайлбар бичих

Муу

«засвар», «шинэчлэл», «asdf»

Сайн

«Нэвтрэх хуудсанд алдааны мессеж
нэмэв»

Дүрэм нь энгийн: «Энэ юу хийсэн бэ?» гэсэн асуултад бүтэн өгүүлбэрээр хариул. Гурван сарын дараа уншиж байгаа хүн бол чи өөрөө.

Нууц түлхүүрийг хэзээ ч GitHub дээр бүү тавь

Нэг удаа тавигдсан түлхүүр нь дараа нь устгасан ч **түүхэнд үлдэнэ**. Тэр түлхүүрийг солихоос өөр аргагүй болно. Claude-д «нууц утгуудыг тусад нь, GitHub руу оруулахгүй байхаар тохируулаад өгөөч» гэж хэлээрэй — тэр хийж өгнө.

GitHub бол чиний CV

Ажил олгогч ярианаас илүү түүхийг харна: чи юу хийдэг вэ, тогтмол ажилладаг уу. Эхний өдрөөсөө хадгалж эхэл — «эхэлсэн» гэсэн цэг ч түүхийн эхлэл болно.

А Хавсралт: Промптын загварууд

Хуулж аваад өөрийнхөөрөө сольж ашигла

Долоон загвар. Хаалт доторх зүйлийг л өөрчилнө.

1. Шинэ зүйл барих

«[юу] хийх гэж байна. Хэрэглэгч нь [хэн], ихэвчлэн [утас/компьютер]-аар орно. Дараах зүйлс байх ёстой: 1) [...] 2) [...] 3) [...]. Одоохондоо [юу] хэрэггүй. Эхлээд юу хийхээ хэлээд, дараа нь хийгээрэй.»

2. Алдаа засах

«[юу хийж байхад] дараах алдаа гарлаа: [мессежийг бүтнээр нь]. Юу болоод байна вэ, яаж засах вэ? Эхлээд шалтгааныг тайлбарлаад дараа нь зас.»

3. Ойлгох

«Энэ хэсэг юу хийж байгааг код мэдэхгүй хүнд ойлгомжтой, богинохон тайлбарлаж өгөөч.»

4. Өөрчлөх

«[аль хэсгийг] [яаж] болгож өгөөч. Бусдыг нь бүү хөндөөрэй.»

5. Шалгуулах

«Энийг хараад юуг нь сайжруулж болох вэ? Ямар тохиолдолд эвдрэх вэ? Хамгийн чухал гурвыг нь эрэмбэлээд хэлээрэй.»

6. Сонголт хийх

«[А] ба [Б] хоёрын аль нь энэ тохиолдолд тохиромжтой вэ? Давуу, сул талыг нь хэлээд нэгийг нь зөвлөөрэй.»

7. Аюулгүй байдал

«Нууц утгуудыг тусад нь тохируулаад, GitHub руу орохгүй байхаар хийгээрэй. Эрхийн шалгалтыг сервер тал дээр хийгээрэй.»

Нэг зөвлөгөө

«Эхлээд юу хийхээ хэлээд, дараа нь хийгээрэй» гэсэн өгүүлбэрийг нэмэх нь хамгийн үр дүнтэй заль. Ингэснээр буруу зүйл рүү явахаас нь өмнө зогсоож чадна.

В Хавсралт: Үгсийн тайлбар

Сонсоод ойлгодог болоход л хангалттай

Үг	Нэг мөрөөр
Код	Компьютерын дагаж биелүүлдэг заавар.
Хэл	Заавар бичих дүрмийн цогц. JavaScript, Python гэх мэт.
HTML	Хуудсанд юу байгааг хэлдэг хэсэг.
CSS	Хэрхэн харагдахыг хэлдэг хэсэг.
JavaScript	Юу болохыг хэлдэг хэсэг. Хөтөч дотор ажиллана.
React	Хуудсыг дахин ашиглах хэсгүүд болгон бүтээх арга.
Next.js	React дээр суурилсан бүтэн вэб систем.
Node.js	JavaScript-ийг сервер дээр ажиллуулдаг зүйл.
Сервер	Хүсэлт хүлээж суудаг компьютер.
Клиент	Хүсэлт илгээдэг тал — хөтөч, эсвэл апп.
localhost	«Энэ компьютер». Зөвхөн чамд харагдана.
Домэйн	jarvis.mn гэх мэт хүн уншдаг хаяг.
Deploy	Хийсэн зүйлээ интернэтэд гаргах.
Database	Хаагаад нээхэд үлддэг өгөгдөл хадгалах газар.
Хүснэгт	Нэг төрлийн зүйлсийн жагсаалт өгөгдлийн сан дотор.
Postgres	Хамгийн өргөн хэрэглэгддэг өгөгдлийн сан.
Supabase	Өгөгдлийн сан, нэвтрэлт, файлыг нэг дор өгдөг үйлчилгээ.
API	Хоёр програм хоорондоо ярилцах гарц.
Git	Ажлын түүх хөтөлдөг цагийн машин.
GitHub	Тэр түүхийг интернэтэд байрлуулах газар.
Commit	Хадгалалтын цэг, тайлбартай.
Branch	Үндсэн ажлаа эвдэлгүй туршиж үзэх салаа.
Vercel	Сайтаа интернэтэд гаргадаг үйлчилгээ.
VS Code	Код бичдэг, засдаг програм.
Терминал	Тушаал бичдэг цонх.
Промпт	Хиймэл оюунд өгч байгаа заавар. Чиний гол хэрэгсэл.
Алдааны мессеж	Компьютер аль зааврыг дагаж чадаагүйгээ хэлж байгаа нь.

Frontend	Хэрэглэгчийн харж байгаа хэсэг.
Backend	Ард нь ажиллаж байгаа хэсэг.
Responsive	Дэлгэцийн хэмжээнд тааруулж өөрчлөгддөг гэсэн үг.

Гацвал дараалан шалга

1. Алдааны мессежийг бүтнээр нь хуулж Claude-д тавьсан уу?
2. Хадгалсан уу? Хуудсаа дахин ачаалсан уу?
3. Хамгийн сүүлд ажиллаж байсан үеэс хойш яг юуг өөрчилсөн бэ?
4. Юу хүлээж байснаа, оронд нь юу болсныг тодорхой хэлж чадаж байна уу?

Асуух газар

Аль ч алхам дээр гацвал бидэнтэй холбогдоорой: info@jarvis.mn · [+976 7000 0000](tel:+97670000000). Хичээлийн нэр, гарсан мессежээ хавсаргавал хариу хурдан ирнэ.