

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Даваадорж Энхманлай

Зайнаас удирдах Веб хөтөч
(Remote access and remote control Web browser)

Мэдээллийн технологи(D061303)
Бакалаврын судалгааны ажил

Улаанбаатар

2024 оны 5 сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
МЭДЭЭЛЛИЙН ТЕХНОЛОГИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХААНЫ ТЭНХИМ

Зайнаас удирдах Веб хөтөч
(Remote access and remote control Web browser)

Мэдээллийн технологи(D061303)
Бакалаврын судалгааны ажил

Удирдагч: _____ Г.Гантулга

Гүйцэтгэсэн: _____ Д.Энхманлай (20B1NUM0690)

Улаанбаатар

2024 оны 5 сар

Зохиогчийн баталгаа

Миний бие Даваадорж Энхманлай ”Зайнаас удирдах Веб хөтөч” сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг Бакалаврын судалгааны ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

УДИРТГАЛ	1
БҮЛГҮҮД	3
1. СУДАЛГАА	3
1.1 Программууд судлах	3
1.2 Шийдлүүдийг боловсруулах	11
2. ШИНЖИЛГЭЭ ЗОХИОМЖ	14
2.1 Шаардлага боловсруулах	14
2.2 Диаграммууд зурах	15
3. ХЭРЭГЖҮҮЛЭЛТ	16
3.1 Сервер	16
3.2 Клиент	18
4. ҮР ДҮН	20
4.1 Ашигласан технологиуд, программчлалын хэл	20
4.2 UI	32
4.3 Сервер	36
4.4 Клиент	42
5. ДҮГНЭЛТ	48
5.1 Үр дүнгийн тайлан	48
НОМ ЗҮЙ	49

ЗУРГИЙН ЖАГСААЛТ

2.1	Server Diagram	15
2.2	Connection Diagram	15
4.1	UI phase 1.1	35
4.2	UI phase 1.2	36
4.3	UI phase 2.1	42
4.4	UI phase 2.2	42
4.5	UI phase 2.3	47
4.6	UI phase 2.4	47

ХҮСНЭГТИЙН ЖАГСААЛТ

1	7 хоногийн үечилсэн төлөвлөгөө	2
---	--------------------------------------	---

УДИРТГАЛ

Миний бие Д.Энхманлай нь Бакалаврын судалгааны ажлын хугацаанд Network, Linux, File server, Remote Control, Web browser гэсэн технологиуд дээр голчлон ажилласан ба уг технологиуд ямар шалтгаанаар үүссэн, цаана нь технологийн ямар дэвшил, хөгжүүлэлтийн арга барил ашигладаг, компаниуд хэрхэн үүн дээр хөгжүүлэлт хийж эцсийн бүтээгдэхүүнийг гаргадаг, Мөн Монголын №1 оператор компани болох "Mobicom corporation llc" дэмжлэгтэйгээр Бакалаврын судалгааны ажлыг хийж гүйцэтгэлээ.

Зорилго Сүлжээний үндсэн бүтэц, загварчлал болон файл сервер, remote access судлан хэрэгжүүлэх

Зорилт Удирдагчийн зааварчилгааны дагуу алхам алхмаар судалгаа хийж өгсөн шаардлагын хүрээнд хэрэгжүүлэлт хийх

Table 1: 7 хоногийн үечилсэн төлөвлөгөө

№	Гүйцэтгэх ажил	Хугацаа	Биелэлт	Удирдагчийн үнэлгээ
1	Программууд судлах	02/13 - 02/20	100	100
2	Шийдлүүдийг боловсруулах	02/20 - 02/27	100	100
3	Шаардлага боловсруулах	02/27 - 03/05	100	100
4	Диаграммууд зурах	03/05 - 03/12	100	100
5	Сервер хэрэгжүүлэлт	03/12 - 05/23	100	100
6	Клиент хэрэгжүүлэлт	04/09 - 05/23	100	100
7	Тайлан бичих	02/13 - 05/23	100	100

1. СУДАЛГАА

1.1 Програмууд судлах

1.1.1 RemotePC

¹ RemotePC нь хэрэглэгчдэд интернэтэд холбогдсон дурын төхөөрөмж ашиглан хаанаас ч хамаагүй компьютер эсвэл сервертээ хандах, удирдах боломжийг олгодог алсаас хандалтын программ хангамжийн шийдэл юм. RemotePC-ийн талаарх зарим мэдээлэл энд байна:

1. Онцлогууд: RemotePC нь дараах функцуудыг санал болгодог.

- Алсын компьютерт хандах: Хэрэглэгчид компьютер дээрээ бодитоор сууж байгаа мэт алсаас хандах боломжтой.

- Платформ хоорондын дэмжлэг: RemotePC нь Windows, Mac, Linux, iOS, Android зэрэг янз бүрийн үйлдлийн системүүдтэй нийцдэг.

- Файл дамжуулах: Хэрэглэгчид алсаас компьютер болон түүнд хандаж байгаа төхөөрөмж хооронд файл дамжуулах боломжтой.

- Алсын зайнаас хэвлэх: Хэрэглэгчид бичиг баримтыг алсаас дотоодын хэвлэгчээр хэвлэх боломжтой.

- Аюулгүй холболт: RemotePC нь TLS v 1.2/AES-256 шифрлэлт болон Хувийн түлхүүрийн баталгаажуулалтаар дамжуулан аюулгүй холболтыг баталгаажуулдаг.

2. Төлөвлөлт ба үнэ: RemotePC нь хувь хүн, жижиг бизнес, аж ахуйн нэгжүүдийн хэрэгцээнд тохирсон өөр өөрөөр санал болгодог. Үнэ нь ихэвчлэн хандахыг хүсэж буй компьютерын тоо болон шаардлагатай функцуудаас хамаарч өөр өөр байдаг.

3. Суулгах ба тохируулах: Хэрэглэгчид ихэвчлэн алсаас хандахыг хүссэн компьютер болон тэдгээрт хандахын тулд ашиглах төхөөрөмждөө RemotePC программыг суулгах шаардлагатай

¹RemotePC official site <https://www.remotedesktop.com>

болдог. Тохируулах процесс нь ихэвчлэн бүртгэл үүсгэх, RemotePC программ хангамжийг суулгах, алсаас хандах тохиргоог хийх зэрэг орно.

4. Хэрэглээ :

- Алсын техникийн дэмжлэг: Мэдээллийн технологийн мэргэжилтнүүд алслагдсан компьютер дээрх асуудлыг биечлэн байхгүй ч олж засварлаж, засах боломжтой.

- Алсаас ажиллах: RemotePC нь ажилчдад гэрээсээ эсвэл аялж байхдаа ажлынхаа компьютерт хандах боломжийг олгож, алсаас ажиллах боломжийг олгодог.

- Хувийн хэрэглээ: Хувь хүмүүс RemotePC-ийг гэрийн компьютертоо хандах эсвэл гэр бүлийн гишүүд, найз нөхдөдөө техникийн туслалцаа үзүүлэх зорилгоор ашиглаж болно.

5. Аюулгүй байдал: RemotePC нь алсаас найдвартай холболтыг баталгаажуулахын тулд шифрлэлт болон баталгаажуулалтын протоколуудыг хэрэгжүүлснээр аюулгүй байдлыг хангадаг. Энэ нь нууц мэдээллийг хамгаалах, алсын компьютерт зөвшөөрөлгүй нэвтрэхээс сэргийлэхэд тусална.

Ерөнхийдөө RemotePC нь компьютерт алсаас хандах, удирдах найдвартай, найдвартай арга хайж буй хувь хүмүүс, бизнес эрхлэгчид болон мэдээллийн технологийн мэргэжилтнүүдэд зориулсан алсаас хандалтын олон талт шийдэл юм.

1.1.2 TeamViewer

² TeamViewer нь хэрэглэгчдэд дэлхийн хаанаас ч алсаас төхөөрөмжид хандах, удирдах боломжийг олгодог алдартай алсаас хандах болон алсын удирдлагатай программ хангамжийн шийдэл юм.

1. Онцлогууд:

- Зайнаас хандалт: Хэрэглэгчид компьютер, сервер эсвэл хөдөлгөөнт төхөөрөмжүүдэд бие махбодтой байгаа мэт алсаас хандаж, удирдах боломж.
 - Платформ хоорондын дэмжлэг: TeamViewer нь Windows, macOS, Linux, iOS, Android зэрэг янз бүрийн үйлдлийн системүүдтэй нийцдэг.
 - Файл дамжуулах: Хэрэглэгчид дотоод болон алсын төхөөрөмжүүдийн хооронд файлуудыг аюулгүйгээр дамжуулах боломжтой.
 - Хамтран ажиллах хэрэгслүүд: TeamViewer нь дэлгэц хуваалцах, самбар зэрэг онлайн уулзалт, танилцуулга, хамтран ажиллах боломжуудыг агуулдаг.
 - Аюулгүй холболт: TeamViewer нь төгсгөл хоорондын шифрлэлт, олон хүчин зүйлийн баталгаажуулалт, хандалтын хяналтаар дамжуулан аюулгүй холболтыг баталгаажуулдаг.
2. Төлөвлөлт ба үнэ: TeamViewer нь хувийн, бизнесийн болон байгууллагын хэрэглэгчдэд зориулсан өөр өөр санал болгодог. Үнэ нь ихэвчлэн хэрэглэгчид болон төхөөрөмжүүдийн тоо, мөн төлөвлөгөөнд тусгагдсан онцлогоос хамаарч өөр өөр байдаг. Захиалгат болон нэг удаагийн худалдан авалтын сонголтууд байдаг.
3. Суулгах ба тохируулах: Хэрэглэгчид TeamViewer-ийг ашиглахын тулд алсаас хандахыг хүссэн төхөөрөмж болон түүнд хандахдаа ашиглах төхөөрөмждөө хоёуланд нь TeamViewer программ хангамжийг татаж аваад суулгах шаардлагатай. Тохируулах процесс нь TeamViewer бүртгэл үүсгэх, программ хангамжийг суулгах, алсын зайнаас хандах тохиргоог тохируулах явдал юм.
4. Хэрэглэх тохиолдлууд: TeamViewer-ийг янз бүрийн зорилгоор ашигладаг, үүнд:

²TeamViewer official site <https://www.teamviewer.com/en/>

- Мэдээллийн технологийн алсаас дэмжлэг үзүүлэх: Мэдээллийн технологийн мэргэжилтнүүд алслагдсан компьютер, сервер эсвэл хөдөлгөөнт төхөөрөмж дээрх асуудлыг засах боломжтой.

- Алсын зайнаас ажиллах: TeamViewer нь ажилчдад ажлын компьютертоо хандах эсвэл хамтран ажиллагсадтайгаа алсаас хамтран ажиллах боломжийг олгож, алсаас ажиллах, виртуал уулзалт хийх боломжийг олгодог.

- Хувийн хэрэглээ: Хувь хүмүүс TeamViewer-ийг ашиглан гэрийн компьютертоо хандах, гэр бүл, найз нөхдөдөө техникийн асуудлаа шийдвэрлэхэд нь туслах эсвэл аялж байхдаа төхөөрөмждөө хандах боломжтой.

5. Аюулгүй байдал: TeamViewer нь нууц мэдээллийг хамгаалах, зөвшөөрөлгүй хандалтаас сэргийлэхийн тулд салбарын стандартын шифрлэлтийн протокол, аюулгүй байдлын арга хэмжээг хэрэгжүүлснээр аюулгүй байдлыг эрхэмлэдэг. Үүнд төгсгөл хоорондын шифрлэлт, холболтын нууц үг, хоёр хүчин зүйлийн баталгаажуулалт орно.

Ерөнхийдөө TeamViewer нь төхөөрөмжид алсаас хандах, удирдах, онлайнаар хамтран ажиллах, алсаас дэмжлэг үзүүлэх найдвартай арга хайж буй хувь хүмүүс, бизнес эрхлэгчид болон мэдээллийн технологийн мэргэжилтнүүдэд тохиромжтой алсаас нэвтрэх олон талын шийдэл юм.

1.1.3 Zoho Assist

³ Zoho Assist нь Zoho корпорацийн хөгжүүлсэн алсын удирдлага, алсаас хандах программ хангамж юм. Энэ нь мэдээллийн технологийн мэргэжилтнүүд болон туслах багуудад компьютер, сервер, хөдөлгөөнт төхөөрөмжүүдэд алсаас хандах, алдааг олж засварлах хэрэгслүүдээр хангадаг.

1. Онцлогууд:

- Алсаас хандах: Техникчид интернэт холболттой хаанаас ч алсын компьютер, сервер, хөдөлгөөнт төхөөрөмжид хандаж, удирдах боломжтой.

- Дэлгэц хуваалцах: Хэрэглэгчид үзүүлэн үзүүлэх, танилцуулга хийх эсвэл хамтран ажиллах зорилгоор дэлгэцээ алсын хэрэглэгчидтэй хуваалцах боломжтой.

- Файл дамжуулах: Zoho Assist нь техникч болон алсын хэрэглэгчийн төхөөрөмжүүдийн хооронд аюулгүй файл дамжуулах боломжийг олгодог.

- Хяналтгүй хандалт: Техникчид байнгын засвар үйлчилгээ эсвэл алдааг олж засварлах зорилгоор алсын компьютерт хяналтгүй хандалтыг тохируулах боломжтой.

- Олон платформын дэмжлэг: Zoho Assist нь Windows, macOS, Linux, iOS, Android зэрэг янз бүрийн үйлдлийн системүүдтэй нийцдэг.

- Хадгалах: Хэрэглэгчид баримтжуулалт эсвэл сургалтын зорилгоор алсаас дэмжлэг үзүүлэх холболтыг хадгалах боломжтой.

- Захиалга: Zoho Assist нь компанийн брэнд, лого бүхий дэмжлэгийн порталыг өөрчлөх боломжийг олгодог.

2. Төлөвлөлт ба үнэ: Zoho Assist нь хувь хүмүүс, жижиг бизнесүүд, аж ахуйн нэгжүүдийн хэрэгцээнд тохирсон өөр өөр үнийг санал болгодог. Үнэ нь ихэвчлэн шаардлагатай техникийн ажилтны тоо болон зэрэгцээ холболт зэргээс хамаарч өөр өөр байдаг.

3. Суулгах ба тохируулах: Zoho Assist-ийг ашиглахын тулд техникч болон алсын хэрэглэгч хоёулаа Zoho Assist клиент программ хангамжийг тус тусын төхөөрөмж дээрээ суулгах шаардлагатай. Тохируулах процесс нь Zoho данс үүсгэх, программ хангамжийг татаж авах, алсаас хандах

³Zoho Assist official site <https://www.zoho.com/assist/remote-access-control.html>

тохиргоог хийх зэрэг орно.

4. Хэрэглээ: Zoho Assist-ийг янз бүрийн зорилгоор ашигладаг, үүнд:

- Мэдээллийн технологийн алсаас дэмжлэг үзүүлэх: Мэдээллийн технологийн мэргэжилтнүүд хэрэглэгчийн компьютер, сервер, гар утасны төхөөрөмж дээрх техникийн асуудлыг алсаас олж засварлаж, шийдвэрлэх боломжтой.

- Тусламжийн ширээний дэмжлэг: Тусламжийн баг нь техникийн асуудалтай тулгарсан үйлчлүүлэгч эсвэл ажилчдад алсаас тусламж үзүүлэх боломжтой.

- Сургалт ба хамтын ажиллагаа: Zoho Assist-ийг оролцогчдод дэлгэцээ хуваалцах, алсаас харилцах боломжийг олгох замаар онлайн сургалт, танилцуулга, хамтын ажилд ашиглах боломжтой.

5. Аюулгүй байдал: Zoho Assist нь алсын холболт болон өгөгдлийг хамгаалахын тулд шифрлэлтийн протокол, аюулгүй байдлын арга хэмжээг хэрэгжүүлснээр аюулгүй байдлыг нэн тэргүүнд тавьдаг. Үүнд өгөгдөл дамжуулахад зориулсан AES 256 битийн шифрлэлт, аюулгүй харилцааны TLS 1.2 протокол орно.

Ерөнхийдөө Zoho Assist нь алсаас тусламж үзүүлэх, техникийн асуудлыг шийдвэрлэхэд хялбар, аюулгүй аргыг хайж буй мэдээллийн технологийн мэргэжилтнүүд, туслах баг, бизнесүүдэд тохиромжтой алсаас дэмжлэг үзүүлэх, алсаас нэвтрэх цогц шийдэл юм.

1.1.4 TSplus

⁴ TSplus нь бизнес эрхлэгчдэд хаанаас ч, ямар ч төхөөрөмжөөс өөрийн аппликейшн болон ширээний компьютерт аюулгүй хандах, удирдах боломжийг олгодог алсын ширээний компьютер болон программ хангамжийн шийдэл юм.

1. Онцлогууд:

- Алсаас хандах: Хэрэглэгчид интернэт холболттой дурын төхөөрөмжөөс өөрийн ширээний компьютер эсвэл виртуалчлагдсан програмдаа алсаас хандаж, удирдах боломжтой.

- Аппликейшн нийтлэх: TSplus нь бизнесүүдэд алсын хэрэглэгчдэд программ нийтлэх боломжийг олгож, ширээний компьютерт бүрэн хандалт өгөхгүйгээр тодорхой программ хангамжид хандах боломжийг олгодог.

- HTML5 клиент: TSplus нь HTML5 веб клиентээр хангадаг бөгөөд энэ нь хэрэглэгчдэд нэмэлт программ хангамж суулгах шаардлагагүйгээр веб хөтчөөс шууд программууд болон ширээний компьютерт хандах боломжийг олгодог.

- Ачаалал тэнцвэржүүлэх: TSplus нь хэрэглэгчийн холболтыг олон серверт түгээх ачааллыг тэнцвэржүүлэх чадварыг багтаасан бөгөөд оновчтой гүйцэтгэл, найдвартай байдлыг хангадаг.

- Алсын зайнаас хэвлэх: Хэрэглэгчид өөрийн алсаас компьютер эсвэл программын сессиноос локал принтерт бичиг баримтыг алсаас хэвлэх боломжтой.

- Аюулгүй байдал: TSplus нь алсын сессиноос болон өгөгдлийг хамгаалахын тулд шифрлэлтийн протокол, хандалтын хяналт, баталгаажуулалтын механизмыг хэрэгжүүлснээр аюулгүй байдлыг нэн тэргүүнд тавьдаг.

2. Лицензүүд: TSplus нь үндсэн, гар утас, аж ахуйн нэгжийн хувилбар зэрэг бүх төрлийн бизнесийн хэрэгцээнд нийцсэн өөр хувилбаруудыг санал болгодог. Лиценз нь хэрэглэгчийн тоо эсвэл шаардлагатай зэрэгцээ холболтоос хамааран өөр өөр байдаг.

3. Суулгах ба тохируулах: TSplus-ийг тохируулах нь хэрэглэгчдийн алсаас хандах шаардлагатай программууд эсвэл компьютер байршуулсан сервер дээр TSplus программ хангамжийг суулгах.

⁴TSplus official site <https://tsplus.net/>

Тохируулах процесс нь хэрэглэгчийн зөвшөөрөл, программыг нийтлэх, аюулгүй байдлын тохиргоог багтаана.

4. Хэрэглээ:

- Алсын ажил: Бизнесүүд TSplus-ийг ашиглан ажилчдаа ажлын ширээний компьютер болон программдаа алсаас хандах боломжийг олгож, алсаас ажиллах болон зайнаас ажиллах боломжийг хөнгөвчлөх боломжтой.

- Хүртээмж: TSplus нь бизнесүүдэд программуудыг өөр өөр байршилд төвлөрсөн байдлаар удирдах, хэрэглэгчдэд хүргэх боломжийг олгож, мэдээллийн технологийн ачаалал, нарийн төвөгтэй байдлыг багасгадаг.

- Алсын дэмжлэг: Мэдээллийн технологийн мэргэжилтнүүд TSplus-ийг ашиглан алсаас тусламж үзүүлж, хэрэглэгчийн ширээний компьютер эсвэл программ дээрх техникийн асуудлыг шийдвэрлэх боломжтой.

5. Дэмжлэг ба шинэчлэлт: TSplus нь хэрэглэгчид хамгийн сүүлийн үеийн функцүүд болон аюулгүй байдлын хангамжийг ашиглах боломжийг хангах үүднээс техникийн дэмжлэг, программ хангамжийн шинэчлэлтүүдийг санал болгодог.

Ерөнхийдөө TSplus нь алсаас нэвтрэх орчинд бүтээмж, уян хатан байдал, аюулгүй байдлыг сайжруулахыг эрэлхийлж буй бизнесүүдэд тохиромжтой, алсын ширээний компьютер болон программ хангамжийн найдвартай шийдэл юм.

1.2 Шийдлүүдийг боловсруулах

1.2.1 Харьцуулалт

RemotePC, TeamViewer, Zoho Assist гэсэн гурван программын давуу болон сул талуудыг харьцуулж, аль нь их сургууль болон ажлын орчинд илүү тохиромжтой болохыг үнэлье.

1. RemotePC:

Давуу тал:

- Хувь хүн, аж ахуйн нэгжүүдэд боломжийн үнийн санал.
- Энгийн бөгөөд хэрэглэгчдэд хялбар интерфейс.
- Файл дамжуулах чадвар сайтай.
- Шифрлэлт, баталгаажуулалт бүхий аюулгүй холболт.
- Хувийн хэрэглээ болон жижиг бизнест тохиромжтой.

Сул тал:

- Бусад шийдлүүдтэй харьцуулахад зарим дэвшилтэт боломжууд дутмаг.
- Хязгаарлагдмал хамтын ажиллагааны хэрэгсэл.
- Томоохон аж ахуйн нэгжүүдийн хувьд тийм ч хангалттай биш.

2. TeamViewer:

Давуу тал:

- Алсаас компьютерт хандах, дэлгэц хуваалцах, файл дамжуулах зэрэг өргөн боломжууд.
- Олон платформын дэмжлэг.
- Онлайн уулзалт, танилцуулга хийхэд зориулсан маш сайн хэрэгсэл.
- Шифрлэлт, баталгаажуулалт бүхий аюулгүй байдлын хүчтэй шийдэл.
- Хувийн болон бизнесийн хэрэглээ, тэр дундаа томоохон аж ахуйн нэгжүүдэд тохиромжтой.

Сул тал:

- Үнийн хувьд бусад шийдлүүдтэй харьцуулахад харьцангуй өндөр.
- Хэрэглэгчид хааяа холболтын асуудлын гомдол хэлдэг.

3. Zoho Assist:

Давуу тал:

- Алсаас хандах, дэлгэц хуваалцах, хараа хяналтгүй хандалт зэрэг алсаас туслах иж бүрэн боломжууд.

- Бизнесийн хувьд боломжийн үнийн санал.
- Бүтээмжийг нэмэгдүүлэхийн тулд бусад Zoho бүтээгдэхүүнтэй нэгтгэж болдог.
- Шифрлэлт, хандалтын хяналт бүхий аюулгүй байдлын сайн арга зохицуулалт.
- Бүх төрлийн бизнест тохиромжтой.

Сул тал:

- Бусад шийдлүүд шиг хамтын ажилд зориулсан өргөн боломжууд байхгүй.
- Бусад алсаас хандах хэрэгслүүдтэй харьцуулахад хязгаарлагдмал тохируулгатай.

4. TSplus

Давуу тал:

- Хэрэглэгчид өөрсдийн ширээний компьютер эсвэл апп-д алсаас хандах боломжийг олгож, уян хатан байдлыг нэмэгдүүлнэ.

- Аппликейшнд сонгон хандахыг зөвшөөрч, аюулгүй байдлыг сайжруулна.
- Өргөн хүртээмжтэй байх үүднээс янз бүрийн үйлдлийн системүүд дээр ажилладаг.
- Илүү сайн ажиллахын тулд хэрэглэгчийн холболтыг түгээдэг.
- Тохиромжтой болгох үүднээс алсаас хэвлэх боломжийг олгоно.
- Өгөгдлийн хамгаалалтад зориулсан шифрлэлт болон хандалтын хяналтыг багтаасан.

Сул тал:

- Том хэмжээний тохиргоонд анхны хөрөнгө оруулалт өндөр байж магадгүй.
- Тохиргоо нь техникийн болон төвөгтэй байж магадгүй.
- Шинэчлэлт болон хэрэглэгчтэй ажиллахад байнгын удирдлага шаардлагатай.
- Өргөтгөхөд нэмэлт төлөвлөлт шаардлагатай байж магадгүй.
- Тогтвортой сүлжээний холболтод тулгуурлан саадгүй нэвтрэх боломжтой.

1.2.2 Шийдэл

Их сургууль болон ажлын орчинд аль нь илүү дээр вэ?

- Зардлын нөлөө чухал бол RemotePC нь боломжийн үнэтэй учир тохиромжтой байж болно.

- Хэрэв хамтын ажиллагааны дэвшилтэт хэрэгсэл, өргөн боломжууд, ялангуяа томоохон аж ахуйн нэгжүүдэд шаардлагатай бол TeamViewer илүү тохиромжтой.

- Өөр программуудтай нэгтгэх чухал бол Zoho Assist илүү тохиромжтой.

- Миний бодлоор ажлын болон их сургуулийн орчинд аль алинд нь энэхүү программ хангамж шаардлага хэрэгцээ байгаа гэж үзэж байна.

-Жишээ 1. Хааяа хувийн компьютергүй их сургууль дээрээ очих өдөр элбэг байдаг. Их сургуулийн лабораторийн компьютер дээр ажилласан даалгавраа өөрийн хувийн компьютер лүүгээ авах гэж заавал лабораторийн компьютер дээр Teams, Facebook гэх мэт программ ашиглан хувийн хаягаараа дамжуулж авдаг. Энэ нь маш эрсдэлтэй юм. Хувийн хаягаа өөр хүнд алдахаас эхлээд нэр төрдөө халдуулах ч үр дагавар гарж болзошгүй.

-Жишээ 2. Ажлын төрлөөс хамаараад өөр өөрийн гэсэн онцлогууд байж болох ч одоо үед ихэнх ажлын газрууд байгууллагын нууцлалын бодлогын дагуу мэдээллийн аюулгүй байдлыг ханган ажлын компьютерыг ашиглах нь элбэг. Хэрвээ би 24/7 хараа хяналттай байх ёстой ажлын байранд ажилладаг гэж үзье(NOC engineer etc). Оффисоос гадуур явж байх үед гэнэт гэмтэл саатал гарах үед асуудал үүснэ. Ажлын газрын суурин компьютер дээр надад шаардлагатай файл байж болзошгүй.

2. ШИНЖИЛГЭЭ ЗОХИОМЖ

Peer-to-Peer загвараар хийхийг зорьсон.

2.1 Шаардлага боловсруулах

2.1.1 Хэрэглэгчид

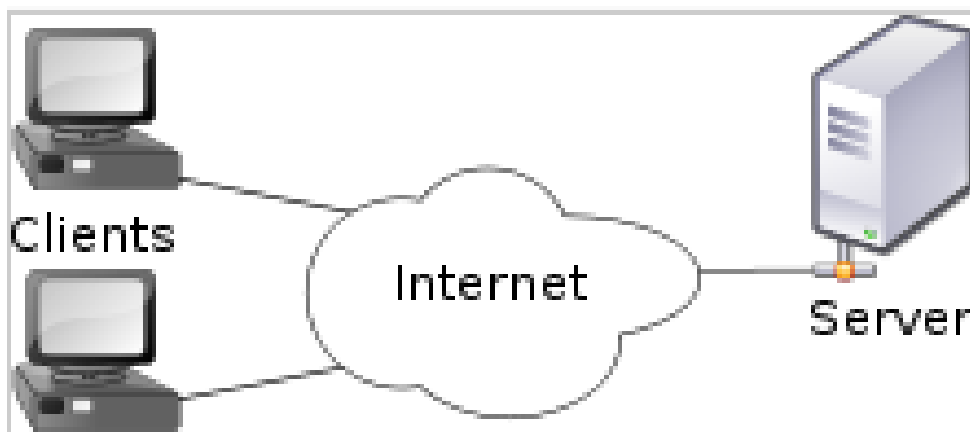
- Хэрэглэгч болон Админаас бүрдэнэ.

2.1.2 Функционал хэрэглэгчийн шаардлагууд

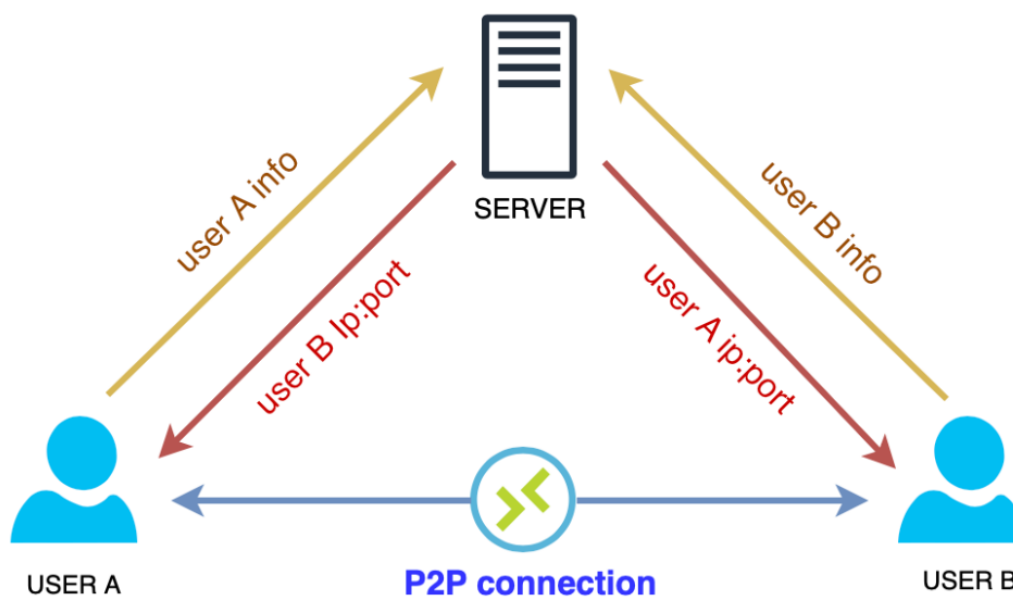
- Хэрэглэгч IP холбоосоор нэвтэрдэг байх
- Хэрэглэгч өөрийн хаяг дээрээ олон төхөөрөмжийг бүртгэх боломжтой байх
- Зөвхөн тухайн төхөөрөмжийг хянах эрхтэй нэг л хэрэглэгч байх
- Холбогдсон төхөөрөмжүүдийн ангилалтай байх
- Хэрэглэгчийн холболтыг хязгаарладаг байх
- Ямар ч төрлийн платформ дээр ашиглаж болохоор байх

2.1.3 Функционал бус шаардлага

- Хурц өнгөгүй байх
- Цэгцтэй, ойлгомжтой байх
- Responsive дизайнтай байх
- Гацалт үүсдэггүй байх
- Ямар ч үед хэвийн ажиллагаатай байх
- Ашиглах заавар авах хэсэгтэй байх



Зураг 2.1: Server Diagram



Зураг 2.2: Connection Diagram

2.2 Диаграммууд зурах

2.2.1 Серверийн topology диаграмм

2.2.2 Хэрэглэгч болон Серверийн topology диаграмм

3. ХЭРЭГЖҮҮЛЭЛТ

3.1 Сервер

Сервер гэдэг нь сүлжээн дэх бусад компьютер эсвэл төхөөрөмжүүдэд үйлчилгээ, нөөцөөр хангадаг компьютер эсвэл систем юм. Энэ нь үндсэндээ хүсэлтийн дагуу бусад компьютер, төхөөрөмж, хэрэглэгчид байж болох мэдээлэл, файл, программ эсвэл нөөцийг үйлчлүүлэгчдэд дамжуулдаг. Серверүүд нь веб сайт байршуулах, файл хадгалах, удирдах, сүлжээний урсгалыг удирдах, программуудыг ажиллуулах, мэдээллийн санд хандах хандалтыг хангах зэрэг төрөл бүрийн ажлуудыг гүйцэтгэхэд зориулагдсан.

Серверүүд нь тодорхой функцээс хамааран өөр өөр төрлөөр ирдэг. Зарим нийтлэг төрлийн серверүүд нь:

1. Веб серверүүд: Эдгээр серверүүд нь веб хуудаснуудыг байршуулж, веб хөтчөөр дамжуулан хүссэн хэрэглэгчдэд веб хуудаснуудаар үйлчилдэг.
2. Файл серверүүд: Файл серверүүд нь сүлжээнд байгаа олон хэрэглэгч эсвэл төхөөрөмжид хандаж, хуваалцаж болох файлуудыг хадгалж, удирддаг.
3. Өгөгдлийн сангийн серверүүд: Өгөгдлийн сангийн серверүүд нь өгөгдлийн санг хадгалж, удирдаж, үйлчлүүлэгчдэд тэдгээрт хадгалагдсан өгөгдлийг сэргээх, удирдах боломжийг олгодог.
4. Мэйл серверүүд: Мэйл серверүүд нь байгууллага эсвэл сүлжээн дэх хэрэглэгчдэд мэйл мессеж илгээх, хүлээн авах, хадгалах үйл ажиллагааг зохицуулдаг.
5. Хэрэглээний серверүүд: Хэрэглээний серверүүд нь ихэвчлэн тархсан тооцооллын орчинд программуудыг ажиллуулах, удирдахад шаардлагатай дэд бүтэц, үйлчилгээгээр хангадаг.
6. DNS серверүүд: DNS (Домэйн Нэрийн Систем) серверүүд нь домэйн нэрийг IP хаяг руу хөрвүүлж, хүний унших боломжтой хаягийг ашиглан веб сайт болон бусад эх сурвалжид хандах боломжийг хэрэглэгчдэд олгодог.
7. Прокси серверүүд: Прокси серверүүд нь үйлчлүүлэгч болон бусад серверүүдийн хооронд

зуучлагчийн үүрэг гүйцэтгэж, гүйцэтгэл, аюулгүй байдал эсвэл нууцлалыг сайжруулахын тулд хүсэлт, хариултыг дамжуулдаг.

Эдгээр нь сүлжээний орчинд тус бүр нь тодорхой зорилгоор үйлчилдэг олон төрлийн серверүүдийн цөөн хэдэн жишээ юм. Серверүүд нь ихэвчлэн зориулалтын функцээ үр дүнтэй, найдвартай гүйцэтгэх зориулалттай тусгай программ хангамжийг ажиллуулдаг.

Миний хувьд "Mobicom Corporation LLC" Software Developer-оор ажиллаж байгаа учир ажлын газраас зөвшөөрөлтэй, түр ашиглах нөхцөлтэйгөөр test server ашиглах эрхтэй болсон.

Энэхүү тест серверийг файл сервер болгон хөгжүүлсэн бөгөөд байгууллагын нууцлалын бодлогоос хамааран тайлагнах зөвшөөрөл хараахан байхгүй болно.

3.2 Клиент

Компьютерын сүлжээний хүрээнд үйлчлүүлэгч гэдэг нь серверээс үйлчилгээ, эх сурвалжийг хүсэх компьютерын программ эсвэл төхөөрөмжийг хэлнэ. Үйлчлүүлэгчид серверээс өгсөн өгөгдөл, үйлчилгээ эсвэл программд хандахын тулд серверүүдтэй харилцах харилцааг эхлүүлдэг. Үйлчлүүлэгчид нь компьютер, ухаалаг гар утас, таблет эсвэл бусад сүлжээнд холбогдсон төхөөрөмж дээр ажилладаг программ хангамжийн программууд байж болно.

Үйлчлүүлэгч болон серверийн хооронд харилцаа холбоо, өгөгдөл дамжуулахад туслах хэд хэдэн бүрэлдэхүүн хэсэг байдаг. Үндсэн бүрэлдэхүүн хэсэг нь үйлчлүүлэгч болон серверийн хооронд өгөгдөл дамжуулах боломжийг олгодог чиглүүлэгч, унтраалга, кабель зэрэг янз бүрийн сүлжээний төхөөрөмжүүдийг багтаасан сүлжээ өөрөө юм.

1. Сүлжээний дэд бүтэц: Үүнд өгөгдөл дамжуулах физик сүлжээг бүрдүүлдэг чиглүүлэгч, унтраалга, кабель зэрэг техник хангамжийн төхөөрөмжүүд орно.

2. Протоколууд: Үйлчлүүлэгч болон сервер хоорондын харилцаа холбоо нь ихэвчлэн TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol) болон бусад сүлжээний протоколд тулгуурладаг. Эдгээр протоколууд нь өгөгдөл дамжуулах дүрэм, конвенцуудыг тодорхойлж, өөр өөр системүүдийн харилцан үйлчлэлийг хангадаг.

3. Галт хана ба хамгаалалтын арга: Сүлжээ болон түүний хөрөнгийг зөвшөөрөлгүй хандалт, хортой халдлага, өгөгдөл зөрчлөөс хамгаалахын тулд галт хана болон бусад хамгаалалтын арга хэмжээг хэрэгжүүлж болно.

4. Middleware: Зарим тохиолдолд үйлчлүүлэгч болон сервер хоорондын харилцааг хөнгөвчлөх зорилгоор дунд программыг ашиглаж болно. Middleware нь мессежний дараалал, гүйлгээний боловсруулалт, алсын зайн процедурын дуудлага (RPCs) зэрэг үйлчилгээгээр хангадаг программ хангамж бөгөөд үйлчлүүлэгч болон серверүүдтэй үр дүнтэй харилцах боломжийг олгодог.

5. Интернет: Интернетээр харилцахын тулд үйлчлүүлэгчид болон серверүүд хоорондоо холбогдсон төхөөрөмжүүдийн өргөн сүлжээгээр өгөгдөл дамжуулахын тулд чиглүүлэгч, унтраалга,

интернэт үйлчилгээ үзүүлэгч (ISP) зэрэг интернэтийн дэд бүтцийн янз бүрийн бүрэлдэхүүн хэсгүүдэд тулгуурладаг.

Ерөнхийдөө эдгээр бүрэлдэхүүн хэсгүүд нь үйлчлүүлэгчид болон серверүүдийн хооронд харилцааны сувгийг бий болгохын тулд хамтран ажиллаж, компьютерын сүлжээгээр өгөгдөл, үйлчилгээ солилцох боломжийг олгодог.

4. ҮР ДҮН

4.1 Ашигласан технологиуд, программчлалын хэл

4.1.1 WebRTC

¹ WebRTC (Web Real-Time Communication) нь энгийн хэрэглээний программчлалын интерфейс (API) ашиглан веб хөтчүүд, гар утасны программууд болон IoT төхөөрөмжүүдийн хооронд шууд бодит цагийн харилцаа холбоог бий болгох нээлттэй төсөл юм. Энэ нь веб хуудаснууд дотор дуудлага хийх, видео чат хийх, PC хооронд файл хуваалцах боломжийг олгодог. WebRTC-ийн онцлогууд:

1. Peer-to-Peer Communication: WebRTC нь хөтчүүд болон бусад программуудын хооронд зуучлагч сервер ашиглахгүйгээр бодит цагийн харилцааг идэвхжүүлдэг.
2. Media Streams: PC хооронд аудио болон видео дамжуулах боломжийг олгодог.
3. Өгөгдлийн суваг: PC хооронд дур зоргоороо өгөгдөл дамжуулах боломжийг олгодог.
4. Дохио өгөх: WebRTC нь сигнал тодорхойлж өгөөгүй байдаг. Энэ нь сессийн хяналтын мессежийг солилцоход дохио өгөх сервер шаарддаг боловч сонгох эсэх нь хөгжүүлэгчээс хамаарна. Websockets, HTTP, XMPP (Extensible Messaging and Presence Protocol) ашиглаж болно.
5. NAT дамжуулалт ба галт ханын зохицуулалт: WebRTC нь NAT дамжуулалтын механизмуудыг агуулдаг бөгөөд энэ нь хоёулаа галт хана эсвэл чиглүүлэгчийн ард байх үед интернэт дэх үйлчлүүлэгчдийн хооронд өгөгдөл солилцох боломжийг олгодог.
6. Аюулгүй байдал: WebRTC нь нууцлал, аюулгүй байдлын үүднээс Datagram Transport Layer Security (DTLS) болон Secure Real-time Transport Protocol (SRTP) ашиглан медиа урсгалыг шифрлэдэг.

¹WebRTC official site <https://webrtc.org/>

4.1. АШИГЛАСАН ТЕХНОЛОГИУД, ПРОГРАММЧЛАЛЫН ХЭЛ БҮЛЭГ 4. ҮР ДҮН

7. Платформ хоорондын нийцтэй байдал: Энэ нь Chrome, Firefox, Safari, Edge, Opera зэрэг томоохон веб хөгчүүдтэй нийцдэг.

8. Нээлттэй эх сурвалж: Нээлттэй төсөл учраас WebRTC нь чөлөөтэй ашиглах боломжтой бөгөөд хөгжүүлэгчдийн өргөн хүрээний дэмжлэгтэй.

WebRTC-ийн программууд нь видео хурал, онлайн тоглоом, файл дамжуулах гэх мэт. Энэ нь веб хуудаснууд болон программууд руу шууд бодит цагийн харилцаа холбоог бий болгох хүчирхэг хэрэгсэл юм.

4.1.2 Go language

² Golang гэж нэрлэдэг Go нь Google-ийн боловсруулсан нээлттэй программчлалын хэл юм. Үүнийг Роберт Гриземер, Роб Пайк, Кен Томпсон нар бүтээсэн бөгөөд 2009 онд анх худалдаанд гарсан. Олон цөмт, сүлжээнд холбогдсон машинууд болон том кодын баазуудын эрин үед программчлалын бүтээмжийг сайжруулах зорилгоор уг хэлийг бүтээжээ. Go хэлний гол онцлогууд:

1. Concurrent and Concurrent: Go нь олон цөмт болон сүлжээнд холбогдсон машинуудаас хамгийн ашигтай программ бичихэд хялбар болгох зорилготой юм. Энэ нь статик хэлбэрээр бичигдсэн, эмхэтгэсэн хэл бөгөөд C хэлнээс салангид синтакстай боловч санах ойн аюулгүй байдал, санах ойн хог цуглуулах, бүтцийн онцлог, CSP маягийн юм.
2. Энгийн: Энгийн, товч синтакстай тул унших, бичихэд хялбар болгодог. Энэ нь шаардлагагүй синтакс арилгаж, хөгжүүлэгчдэд программынхаа логик дээр анхаарлаа төвлөрүүлэх боломжийг олгодог.
3. Зэрэгцээ: Goroutines нь Go-ийн ажиллах хугацаанд удирддаг хөнгөн утаснууд бөгөөд хялбар параллелизм хийх боломжийг олгодог. Сувгууд нь Goroutines хоорондын харилцаа холбоо, синхрончлохыг хөнгөвчилдөг.
4. Баялаг стандарт сан: Сүлжээ, криптограф, файлын оролт гаралт гэх мэт төрөл бүрийн ажлуудад иж бүрэн дэмжлэг үзүүлэх баялаг стандарт сантай.
5. Статик бичилт: Статик хэлбэрээр бичигдсэн, өөрөөр хэлбэл хувьсагчийн төрлүүд эмхэтгэх үед тодорхойлогддог бөгөөд энэ нь хөгжүүлэлтийн явцад алдаа гаргахад хялбар засна.
6. Санах ойн хог: Автоматаар санах ойн хог цуглуулдаг бөгөөд энэ нь санах ойг автоматаар удирдаж, санах ойн алдагдал гэх мэт программчлалын нийтлэг алдаанаас сэргийлдэг.
7. Compiled хэл: Машин кодыг хөрвүүлдэг. Энэ нь Go программыг interpreted хэлээс илүү хурдан болгодог.
8. Cross-Platform: Windows, macOS, Linux зэрэг олон платформ дээр эмхэтгэлийг дэмждэг

²Go language official site <https://go.dev/>

4.1. АШИГЛАСАН ТЕХНОЛОГИУД, ПРОГРАММЧЛАЛЫН ХЭЛ БҮЛЭГ 4. ҮР ДҮН

тул платформ хоорондын программуудыг хөгжүүлэхэд хялбар болгодог.

9. Нээлттэй эх: Нээлттэй эх хэл бөгөөд түүний хөгжлийг хөгжүүлэгчид болон хувь нэмэр оруулагчдын томоохон нөлөөтэй.

10. Google-ээр дэмжигдсэн: Go-г Google хөгжүүлсэн бөгөөд үүнийг Google-ийн олон төсөлд ашиглаж байна. Энэ нь найдвартай байдлыг нэмэгдүүлж, байнгын хөгжил, дэмжлэгийг баталгаажуулдаг.

Go нь веб сервер, сүлжээний хэрэгсэл, тархсан систем болон бусад олон зүйлийг хөгжүүлэхэд түгээмэл байдаг. Энгийн байдал, гүйцэтгэл, зэрэгцээ ажиллахад зориулсан суурилуулсан дэмжлэг нь үүнийг олон хөгжүүлэгчдэд, ялангуяа үүлэн тооцоолол болон микро үйлчилгээний архитектурт сонирхолтой сонголт болгодог.

4.1.3 TypeScript

³ TypeScript нь Microsoft-оос хөгжүүлж, засвар үйлчилгээ хийдэг нээлттэй программчлалын хэл юм. Энэ нь JavaScript-ийн багц бөгөөд ямар ч JavaScript код нь мөн TypeScript кодыг дэмждэг гэсэн үг юм. TypeScript нь хөрвүүлэх явцад программчлалын алдааг олж, урьдчилан сэргийлэхэд ашиглаж болох статик хэл. TypeScript-ийн гол онцлогууд:

1. Статик бичилт: Хөгжүүлэгчдэд хувьсагчийн төрлүүд, функцийн параметрууд болон буцаах утгуудыг тодорхойлох боломжийг олгодог. Энэ нь алдааг боловсруулах явцад эрт илрүүлж, кодыг урьдчилан таамаглах боломжтой, өөрөө алдаагаа олоход тусална.
2. Type Inference: Төрлүүдийг тодорхой заагаагүй тохиолдолд төрлүүдийн тодорхойлолтыг ашигладаг. TypeScript нь хувьсагчийн төрлийг хэрэглээнээс нь тодорхойлох боломжтой байдаг тул төрлийг үргэлж тодорхой тэмдэглэх шаардлагагүй гэсэн үг.
3. Объект хандалтат программчлал: TypeScript нь анги, интерфейс, удамшил, хандалтын хувиргагч (нийтийн, хувийн, хамгаалагдсан), ерөнхий шинж чанарууд зэрэг объект хандалтат программчлалын функцийг дэмждэг.
4. ES6/ES7-ийн онцлог: TypeScript нь ECMAScript 6 (ES6) болон ECMAScript 7 (ES7)-ийн олон функцийг дэмждэг бөгөөд энэ нь орчин үеийн, онцлог шинж чанартай хэл юм. Эдгээр функцүүд нь arrow функцүүд, ангиуд, асинхрончлол/хүлээлт гэх мэт.
5. Tool support: TypeScript нь Visual Studio Code, Sublime Text, WebStorm зэрэг алдартай код засварлагчтай нэгтгэх зэрэг багаж хэрэгслийн хүчтэй дэмжлэгтэй. Редакторууд код бөглөх, дахин боловсруулах, шугаман баримтжуулалт гэх мэт функцийг хангаж чадна.
6. JavaScript сантай нийцтэй: JavaScript санг ашиглах боломжтой. TypeScript тодорхойлолтын файлууд ('.d.ts' өргөтгөлтэй) нь одоо байгаа JavaScript кодын хэлбэрийг дүрсэлнэ.
7. Код засвар: Статик бичих замаар TypeScript нь кодын баазыг ялангуяа том төслүүдэд илүү засвартай болгодог. Энэ нь кодыг баримтжуулахад тусалж, шинэ хөгжүүлэгчдэд ойлгох, хувь нэмэр оруулахад хялбар болгодог.

³TypeScript official site <https://www.typescriptlang.org/>

8. Эмхэтгэх: TypeScript кодыг JavaScript код руу хөрвүүлэх ба дараа нь ямар ч JavaScript орчинд ажиллуулах боломжтой. TypeScript нь төрөл бүрийн зорилтот орчин болон ECMAScript хувилбаруудыг дэмждэг бөгөөд хөгжүүлэгчид хуучин платформуудыг дэмжихийн зэрэгцээ орчин үеийн JavaScript бичих боломжийг олгодог.

9. Support: TypeScript нь өргөн цар хүрээтэй, идэвхтэй нийгэмлэгтэй бөгөөд Microsoft-оос байнгын хөгжүүлэлт, дэмжлэг үзүүлдэг.

10. Нээлттэй эх сурвалж: TypeScript нь нээлттэй эхийн төсөл бөгөөд үнэ төлбөргүй ашиглах боломжтой бөгөөд хэн ч үүнийг хөгжүүлэхэд хувь нэмрээ оруулах боломжтой.

TypeScript нь том хэмжээний веб хөгжүүлэлт, ялангуяа JavaScript-ийг удаан хугацаанд хадгалахаар төлөвлөж буй төслүүдэд ихэвчлэн ашиглагддаг. JavaScript-ийн орчин үеийн шинж чанаруудын хамт статик бичих чадвар нь үүнийг өргөтгөх боломжтой, засвар үйлчилгээ хийх боломжтой программуудыг бүтээх хүчирхэг хэл болгодог.

4.1.4 Docker

⁴ Docker бол контейнер ашиглан программуудыг үүсгэх, байршуулах, ажиллуулахад хялбар болгох зорилготой платформ юм. Контейнер нь программыг сан болон бусад хамаарал гэх мэт шаардлагатай бүх хэсгүүдийн хамт багцалж, бүгдийг нь нэг багц болгон хүргэх боломжийг хөгжүүлэгчид олгодог. Энэ нь ихэвчлэн нэг компьютерын орчноос нөгөөд шилжихэд программууд жигд ажиллахад ашиглагддаг.

Докерын бүрэлдэхүүн хэсгүүд:

1. Docker Engine: Docker Engine нь Docker-ийн үндсэн хэсэг юм. Энэ нь Docker контейнеруудыг бүтээж, ажиллуулдаг хөнгөн жинтэй ажиллах хугацаа, багаж хэрэгсэл юм. Энэ нь Docker даэмон (dockerd), REST API болон CLI докерыг агуулдаг.

2. Docker Images: Энэ нь контейнерийн үндэс суурь болдог. Зураг нь код, ажиллах хугацаа, сан, орчны хувьсагч, тохиргооны файл зэрэг программ хангамжийг ажиллуулахад шаардлагатай бүх зүйлийг багтаасан хөнгөн жинтэй, бие даасан, гүйцэтгэх боломжтой програм хангамжийн багц юм. Зургууд нь ихэвчлэн бусад зураг дээр тулгуурладаг.

3. Docker containers: Докер контейнерууд нь Docker зургийн жишээнүүд юм. Тэд бодит хэрэглүүрийг ажиллуулж, тухайн программ ажиллаж байгаа орчныг бүрдүүлдэг. Докерын CLI ашиглан контейнерыг эхлүүлэх, зогсоох, зөөх, устгах боломжтой. Контейнер бүр нь тусгаарлагдсан, аюулгүй хэрэглээний платформ бөгөөд программ нь ямар ч орчинд тогтвортой ажиллах боломжийг олгодог.

4. Docker Hub: Docker Hub нь Docker-ийн зургийг хуваалцах, удирдах зориулалттай cloud-д суурилсан репозитор юм. Энэ нь контейнерын дүрсийг илрүүлэх, түгээх, өөрчлөх менежментийн төвлөрсөн эх сурвалж болдог.

Гол ойлголтууд:

1. Савлах: Докер нь программ болон түүний хамаарлыг контейнерт багтаахын тулд контейнержилтийг ашигладаг. Контейнержуулалт нь хөгжүүлэгчдэд программыг бүх хамааралтайгаар нь хялбархан

⁴Docker official site <https://www.docker.com/get-started/>

4.1. АШИГЛАСАН ТЕХНОЛОГИУД, ПРОГРАММЧЛАЛЫН ХЭЛ БҮЛЭГ 4. ҮР ДҮН

багцалж, ямар ч дэд бүтцэд тогтвортой ажиллах боломжийг олгодог.

2. Зөөх чадвар: Docker контейнер нь зөөврийн бөгөөд үйлдлийн систем эсвэл үүлэн үйлчилгээ үзүүлэгчээс үл хамааран Docker суулгасан ямар ч машин дээр ажиллах боломжтой. Энэхүү зөөврийн чадвар нь программыг хөгжүүлэлтээс туршилт, үйлдвэрлэл хүртэл өөр өөр орчинд шилжүүлэхэд хялбар болгодог.

3. Тусгаарлах: Docker контейнер бүр өөр өөрийн гэсэн тусгаарлагдсан орчинд ажилладаг бөгөөд энэ нь программын гүйцэтгэлийг өөр өөр орчинд тогтвортой байлгах, программ хоорондын зөрчилдөөнөөс сэргийлдэг.

4. Microservices Architecture: Докерийг ихэвчлэн микро үйлчилгээний архитектурт ашигладаг бөгөөд программууд нь жижиг бие даасан үйлчилгээнүүдэд хуваагддаг. Үйлчилгээ бүр өөрийн гэсэн саванд ажилладаг бөгөөд энэ нь илүү хялбар болгох, байршуулах, удирдах боломжийг олгодог.

5. Docker Compose: Docker Compose нь олон контейнерт Docker программуудыг тодорхойлох, ажиллуулах хэрэгсэл юм. Энэ нь программын үйлчилгээ болон хамаарлыг тохируулахын тулд YAML файлыг ашигладаг бөгөөд нэг үйлчилгээ болгон олон контейнер үүсгэх, удирдах боломжийг олгодог.

6. Docker Swarm and Kubernetes: Кластер орчинд олон Docker контейнерыг удирдах боломжийг олгодог контейнер зохион байгуулах хэрэгсэл юм. Эдгээр хэрэгслүүд нь контейнерт агуулагдсан программуудыг байрлуулах, масштаблах, удирдах ажлыг автоматжуулдаг.

Docker нь орчин үеийн программ хангамжийг хөгжүүлэх, байршуулах үйл явцын салшгүй хэсэг болж, хөгжүүлэгчдэд программуудыг хурдан бөгөөд үр дүнтэй бүтээх, тээвэрлэх, ажиллуулах боломжийг олгосон. Энэ нь хөгжүүлэлт болон үйл ажиллагааны багуудад тогтвортой орчин бүрдүүлж, өөр өөр орчинд программуудыг хамтран ашиглах, байрлуулахад хялбар болгодог.

4.1.5 WebSocket

⁵ WebSocket нь үйлчлүүлэгч болон серверийн хооронд нэг урт хугацааны холболтоор бүрэн дуплекс холбооны сувгуудыг хангадаг харилцааны протокол юм. Энэ нь веб хөтчүүд болон веб серверүүдэд хэрэгжихээр бүтээгдсэн бөгөөд бодит цагийн, хоцрогдол багатай харилцах боломжийг олгодог. WebSocket нь чат программ, онлайн тоглоом, санхүүгийн арилжааны платформ, спортын шууд шинэчлэлт зэрэг бодит цагийн өгөгдөл дамжуулах шаардлагатай программуудад ихэвчлэн ашиглагддаг. WebSocket-ийн зарим гол санааг энд оруулав.

Гол онцлог:

1. Бүрэн давхар харилцаа холбоо: WebSocket нь бүтэн дуплекс холбоог хангадаг. Энэ функц нь бодит цагийн өгөгдөл дамжуулах боломжийг олгодог.
2. Байнгын холболт: Хүсэлт-хариу загварыг дагаж, хүсэлт бүрд шинэ холболт нээдэг HTTP-ээс ялгаатай нь WebSocket нь үйлчлүүлэгч болон серверийн хооронд байнгын холболтыг бий болгодог. Холболт хийгдсэний дараа үйлчлүүлэгч эсвэл сервер үүнийг хаахаар шийдтэл нээлттэй хэвээр байна.
3. Бага хоцролт: WebSocket нь хоцролтыг багасгах зорилготой юм. Анхны гар барих ажиллагаа дууссаны дараа HTTP толгойн нэмэлт ачаалалгүйгээр клиент болон серверийн хооронд өгөгдөл дамжуулах боломжтой бөгөөд ингэснээр хоцролт багасаж, харилцаа холбоо илүү хурдан болно.
4. Зурвасын өргөнийг үр ашигтай ашиглах: Нэг TCP холболтыг ашигладаг бөгөөд олон холболт үүсгэхтэй холбоотой нэмэлт зардлыг бууруулдаг. Энэ үр ашиг нь зурвасын өргөнийг илүү сайн ашиглах, гүйцэтгэлийг сайжруулахад хүргэдэг.
5. Домэйн хоорондын харилцаа: WebSocket нь домэйн хоорондын харилцаа холбоог зөвшөөрдөг бөгөөд энэ нь WebSocket клиент нь өөр домэйн дээрх WebSocket сервертэй холболт үүсгэх боломжтой гэсэн үг юм. Энэ функц нь илүү уян хатан, өргөтгөх боломжтой веб программуудыг хөгжүүлэх боломжийг олгодог.

⁵WebSocket lesson site https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

6. Аюулгүй харилцаа: WebSocket нь HTTP-той ижил аюулгүй залгуурын давхарга (SSL) шифрлэлтийг ашиглаж, үйлчлүүлэгч болон серверийн хооронд аюулгүй холбооны сувгийг хангах боломжтой.

4.1.6 Zscaler

⁶ Zscaler бол аюулгүй веб гарц, cloud-д суурилсан галт ханын үйлчилгээ, cloud-д суурилсан сүлжээний аюулгүй байдлын шийдлүүдийг хангадаг үүлэн хамгаалалтын компани юм. Энэхүү платформ нь хэрэглэгчид болон төхөөрөмжүүдийн аль алинд нь cloud-д суурилсан интернэтийн аюулгүй байдлыг хангадаг үйлчилгээ хэлбэрээр аюулгүй байдлыг санал болгодог.

Гол бүрэлдэхүүн хэсгүүд:

1. Zscaler интернэт хандалт (ZIA): Zscaler Internet Access нь интернэт болон гадаад программуудад аюулгүй нэвтрэх боломжийг олгодог үүлэн аюулгүй байдлын платформ юм. ZIA нь хэрэглэгчдийг кибер аюулаас хамгаалж, компанийн бодлогыг хэрэгжүүлж, интернэтийн урсгалыг бодит цаг хугацаанд харуулах, хянах боломжийг олгодог. ZIA-ийн гол онцлогууд нь:

- Secure Web Gateway (SWG): ZIA нь кибер аюулыг илрүүлэх, урьдчилан сэргийлэх зорилгоор интернэтэд холбогдсон бүх урсгалыг шалгаж, аюулгүй веб гарц болж үйлчилдэг. - URL шүүлтүүр: ZIA нь интернэт ашиглалтын бодлогыг хэрэгжүүлэх, хортой эсвэл зохисгүй веб сайт руу нэвтрэхээс урьдчилан сэргийлэхийн тулд URL шүүлтүүрийг ашигладаг. - Cloud Firewall: ZIA нь дотогшоо болон гадагш чиглэсэн интернэтийн урсгалыг нарийн хянах боломжийг олгодог үүлэнд суурилсан галт ханыг агуулдаг. - Дэвшилтэт аюулын хамгаалалт: ZIA нь хамгаалагдсан хамгаалалт, вирусийн эсрэг хамгаалалт, аюулын тагнуул зэрэг дэвшилтэт аюулаас хамгаалах боломжуудыг санал болгодог.

2. Zscaler хувийн хандалт (ZPA): Zscaler Private Access нь дата төв эсвэл нийтийн үүлэнд байршуулсан дотоод аппликешнүүдэд найдвартай үүлэн үйлчилгээ юм. ZPA нь хэрэглэгчид программуудыг интернэтэд оруулахгүйгээр аюулгүйгээр хандах боломжийг олгодог. ZPA-

⁶Zscaler official site <https://www.zscaler.com/>

4.1. АШИГЛАСАН ТЕХНОЛОГИУД, ПРОГРАММЧЛАЛЫН ХЭЛ БҮЛЭГ 4. ҮР ДҮН

ийн гол онцлогууд нь:

- Архитектур: ZPA нь дотоод аппликейшнд нэвтрэхээс өмнө хэрэглэгчид болон төхөөрөмжүүдийг баталгаажуулж, зөвшөөрөл авсан эсэхийг баталгаажуулж, аюулгүй байдлыг баримталдаг. - Программ хангамжаар тодорхойлогдсон периметр (SDP): ZPA нь дотоод хэрэглээнд аюулгүй, таних тэмдэгт суурилсан хандалтын хяналтыг бий болгохын тулд программ хангамжаар тодорхойлсон периметрийг ашигладаг. - Программын сегментчилэл: ZPA аппликейшнүүдэд хандах хандалтыг сегментчилснээр хэрэглэгчид зөвхөн ашиглах эрхтэй программууддаа хандах боломжтой. - Хандалтын бодлого: ZPA нь хэрэглэгчийн таниулбал, төхөөрөмжийн байрлал, хэрэглээний мэдрэмж дээр тулгуурлан хандалтын бодлогыг хэрэгжүүлдэг.

3. Zscaler Digital Experience (ZDX): Zscaler Digital Experience нь дижитал туршлагыг харагдуулах, хянах боломжийг олгодог үүлд суурилсан платформ юм. ZDX нь байгууллагуудад интернэт болон үүлэн программынхаа гүйцэтгэлийг хянах, сайжруулахад тусалдаг. ZDX-ийн гол онцлогууд нь:

- Программын гүйцэтгэлийн хяналт (APM): ZDX нь аппликейшны гүйцэтгэлийн бодит цагийн хяналтыг бий болгож, байгууллагуудад гүйцэтгэлийн асуудлыг тодорхойлж шийдвэрлэхэд тусалдаг. - Эцсийн хэрэглэгчийн туршлагын хяналт (EUEM): ZDX нь эцсийн хэрэглэгчийн туршлагыг хянаж, программын ашиглалт, гүйцэтгэл, хэрэглэгчийн сэтгэл ханамжийн талаарх ойлголтыг өгдөг. - Сүлжээний ойлголт: ZDX нь сүлжээний урсгалыг харагдуулах боломжийг олгож, байгууллагуудад программуудыг хэрхэн ашиглаж байгааг ойлгох, сүлжээний гүйцэтгэлийг оновчтой болгох боломжийг олгодог.

Гол онцлог:

1. Cloud-Native Architecture: Zscaler нь үүлд суурилсан архитектур дээр бүтээгдсэн бөгөөд байгууллагуудад дотоод дэд бүтэц ашиглахгүйгээр хэрэглэгчид болон төхөөрөмжүүдийг аппликейшнууд найдвартай холбох боломжийг олгодог.

2. Аюулгүй байдал: Zscaler нь бүх хэрэглэгчид болон төхөөрөмжүүд нь дотоод аппликейшнд нэвтрэхээс өмнө баталгаажуулж, зөвшөөрөл авсан эсэхийг баталгаажуулах Zero Trust аюулгүй

4.1. АШИГЛАСАН ТЕХНОЛОГИУД, ПРОГРАММЧЛАЛЫН ХЭЛ БҮЛЭГ 4. ҮР ДҮН

байдлын загварыг баримталдаг.

3. Дата төвүүдийн дэлхийн сүлжээ: Zscaler нь дэлхийн хаана ч байсан программ, үйлчилгээнд хоцрогдол багатай хандах боломжийг хангадаг дэлхийн дата төвүүдийн сүлжээг ажиллуулдаг.

4. Хамгаалалтын нэгдсэн стек: Zscaler нь веб хамгаалалт, cloud, галт хана, хамгаалагдсан хязгаарлагдмал орчинд ашиглах, аюулаас хамгаалах дэвшилтэт хамгаалалт, мэдээллийн алдагдлаас урьдчилан сэргийлэх зэрэг аюулгүй байдлын нэгдсэн стекээр хангадаг.

5. Өргөтгөх, уян хатан байдал: Zscaler нь өндөр цар хүрээтэй, уян хатан тул байгууллагуудад өөрчлөгдөж буй бизнесийн шаардлагад дасан зохицож, шаардлагатай бол аюулгүй байдлын дэд бүтцээ өргөжүүлэх боломжийг олгодог.

6. Төвлөрсөн удирдлага, тайлагнах: Zscaler нь төвлөрсөн удирдлага, тайлагнах боломжийг олгодог бөгөөд байгууллагууд өөрсдийн аюулгүй байдлын төлөв байдлыг нэг консолоос хянах, удирдах боломжийг олгодог.

7. Зохицуулалт ба зохицуулалтын дэмжлэг: Zscaler нь аюулгүй байдлын шаардлагатай хяналт, тайлагнах чадавхыг хангаснаар байгууллагуудад дагаж мөрдөх болон зохицуулалтын шаардлагыг хангахад тусалдаг.

Zscaler-ийг байгууллагууд интернэт болон үүлэнд суурилсан хэрэглээнийхээ аюулгүй байдлыг хангах, кибер аюулаас хамгаалах, зохицуулалтын шаардлагыг хангах зорилгоор өргөнөөр ашигладаг. Энэ нь ямар ч төхөөрөмж, байршлаас программ, өгөгдөл, үйлчилгээнд аюулгүй нэвтрэх боломжийг олгодог үүлэнд суурилсан аюулгүй байдлын цогц платформыг санал болгодог.

4.2 UI

Гол шаардлага нь интерфейс байсан. Сурах ур чадвар:

- Web Browser бүтэц ойлгох
- Тест сервер ажлын байран дээр бий болгох
- Админ эрхээр ажиллах
- Сүлжээний аюулгүй байдал
- Cryptography
- Stress test, Unit test etc ...

Web Browser-ын шаардлага:

- Хэрэглэхэд маш хялбар байх
- багадаа 2, ихдээ 4 өнгөнөөс бүрдэх
- Script бэлдэх

4.2.1 Code-workspace

```
1  {
2    "folders": [
3      {
4        "path": "server"
5      },
6      {
7        "path": "client"
8      },
9      {
10       "path": "."
11     }
12   ],
13   "settings": {
14     "editor.tabSize": 2,
15     "editor.insertSpaces": true,
16     "editor.detectIndentation": false,
17     "files.encoding": "utf8",
18     "files.eol": "\n",
19     "typescript.tsdk": "./client/node_modules/typescript/lib",
20     "todo-tree.filtering.excludeGlobs": ["**/node_modules/**"],
21     "eslint.validate": [
22       "vue",
23       "javascript",
24       "javascriptreact",
25       "typescript",
26       "typescriptreact",
```

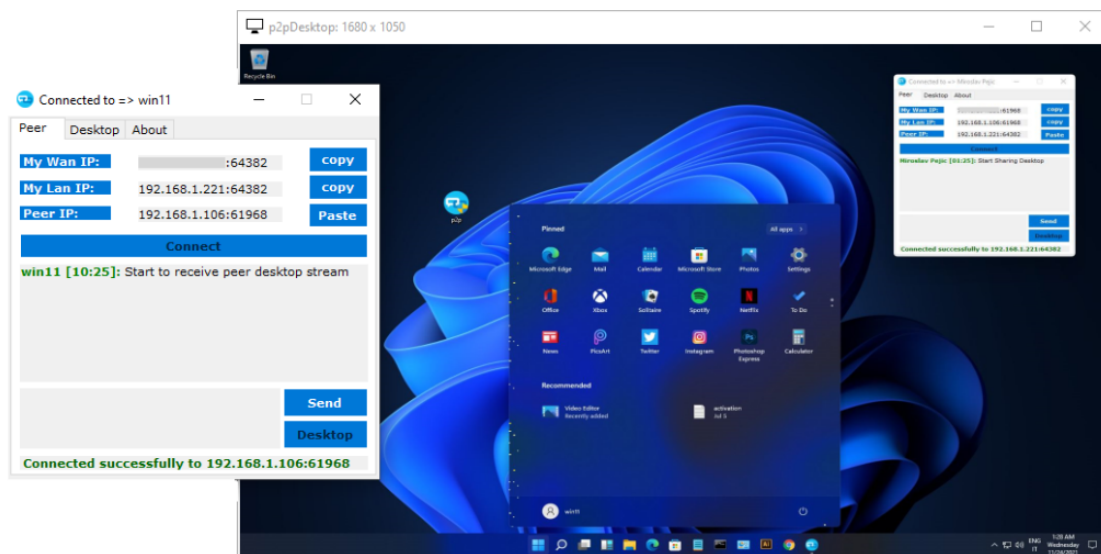
```
27     ],
28     "vetur.validation.template": true,
29     "vetur.useWorkspaceDependencies": true,
30     "remote.extensionKind": {
31       "ms-azuretools.vscode-docker": "ui"
32     },
33     "remote.containers.defaultExtensions": [
34       "ms-vscode.go",
35       "octref.vetur",
36       "ms-vscode.cpptools",
37       "esbenp.prettier-vscode",
38       "dbaeumer.vscode-eslint",
39       "ms-vscode-remote.vscode-remote-extensionpack",
40       "ms-vscode-remote.remote-containers",
41       "ms-azuretools.vscode-docker",
42       "editorconfig.editorconfig",
43       "gruntfuggly.todo-tree",
44       "eamodio.gitlens",
45       "swyphcosmo.spellchecker"
46     ]
47   }
48 }
```


The image shows a web application interface with the following elements:

- Tabs:** A row of three tabs: "Peer" (active), "Desktop", and "About".
- IP Fields:** Three rows of input fields. The first row contains "My Wan IP:" and ":58806". The second row contains "My Lan IP:" and "192.168.1.221:58806". The third row contains "Peer IP:" and an empty field.
- Action Buttons:** To the right of each IP field is a blue button: "copy" for the first two rows, and "Paste" for the third row.
- Connect Button:** A large blue button labeled "Connect" is positioned below the IP fields.
- Message Area:** A large, empty light gray rectangular area occupies the center of the interface.
- Send and Desktop Buttons:** At the bottom, there is a light gray rectangular area with a vertical cursor on the left. To its right are two stacked blue buttons: "Send" (top) and "Desktop" (bottom).

Зураг 4.1: UI phase 1.1

4.2.2 Харагдах байдал



Зураг 4.2: UI phase 1.2

4.3 Сервер

Remote Web Browser хийх үйл явцыг эхний алхамаас нь сурахыг зорьсон бөгөөд файл болон чат Сервер хөгжүүлж үзсэн.

Сурах ур чадвар:

- Админ эрхээр хэрхэн ажиллах
- CLI ажиллах
- Олон технологиудыг хооронд нь уяалдуулан ашиглах

Апп-ын шаардлага:

- Visual Studio ашиглах
- Тестын серверээс өгөгдөл татах
- Тестын сервер лүү өгөгдөл илгээх

4.3.1 *index.ts*

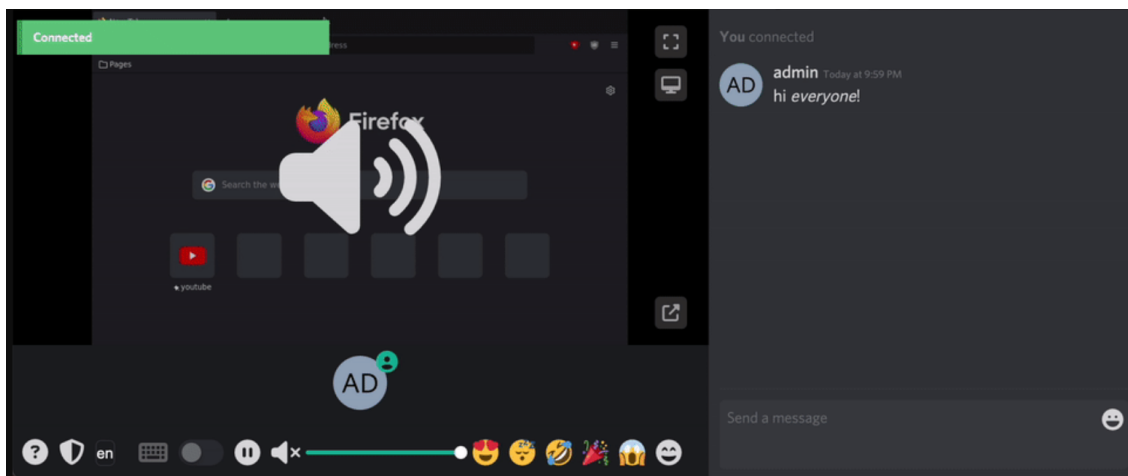
```
1  import Vue from 'vue'
2  import Vuex from 'vuex'
3  import { useAccessor, mutationTree, getterTree, actionTree } from '
    typed-vuex'
4  import { EVENT } from '~/neko/events'
5  import { AdminLockResource } from '~/neko/messages'
6  import { get, set } from '~/utils/localstorage'
7
8  import * as video from './video'
9  import * as chat from './chat'
10 import * as files from './files'
11 import * as remote from './remote'
12 import * as user from './user'
13 import * as settings from './settings'
14 import * as client from './client'
15 import * as emoji from './emoji'
16
17 export const state = () => ({
18   displayname: get<string>('displayname', ''),
19   password: get<string>('password', ''),
20   active: false,
21   connecting: false,
22   connected: false,
23   locked: {} as Record<string, boolean>,
24 })
25
```

```
26 export const mutations = mutationTree(state, {
27   setActive(state) {
28     state.active = true
29   },
30
31   setLogin(state, { displayname, password }: { displayname: string;
32     password: string }) {
33     state.displayname = displayname
34     state.password = password
35   },
36
37   setLocked(state, resource: string) {
38     Vue.set(state.locked, resource, true)
39   },
40
41   setUnlocked(state, resource: string) {
42     Vue.set(state.locked, resource, false)
43   },
44
45   setConnecting(state) {
46     state.connected = false
47     state.connecting = true
48   },
49
50   setConnected(state, connected: boolean) {
51     state.connected = connected
52     state.connecting = false
53     if (connected) {
```

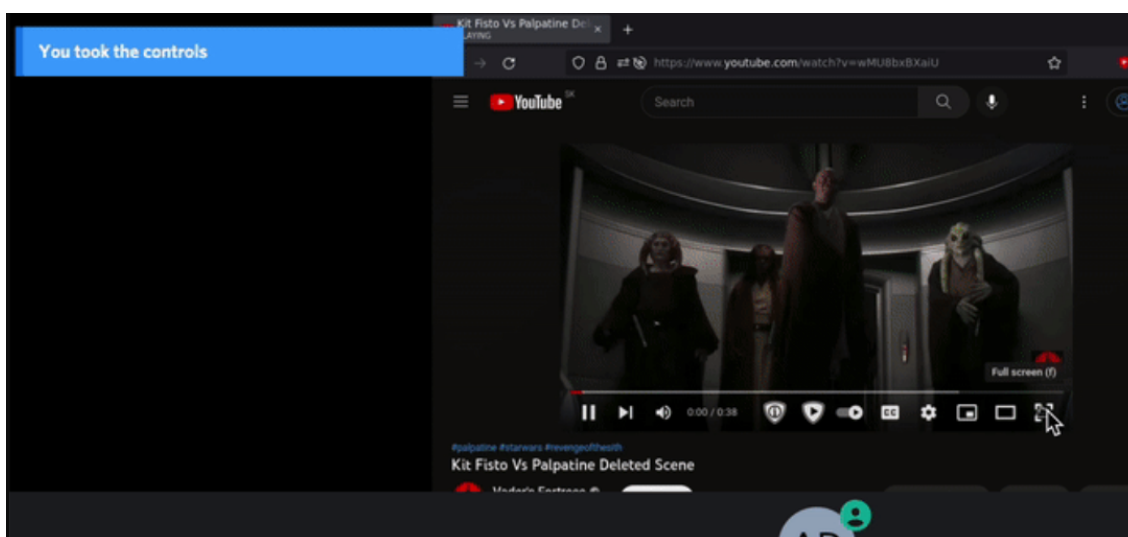
```
53     set('displayname', state.displayname)
54     set('password', state.password)
55   }
56 },
57 })
58
59 export const getters = getterTree(state, {
60   isLocked: (state) => (resource: AdminLockResource) => resource in
61     state.locked && state.locked[resource],
62 })
63
64 export const actions = actionTree(
65   { state, getters, mutations },
66   {
67     initialise() {
68       accessor.emoji.initialise()
69       accessor.settings.initialise()
70     },
71
72     lock(_, resource: AdminLockResource) {
73       if (!accessor.connected || !accessor.user.admin) {
74         return
75       }
76
77       $client.sendMessage(EVENT.ADMIN.LOCK, { resource })
78     },
79
80     unlock(_, resource: AdminLockResource) {
```

```
80     if (!accessor.connected || !accessor.user.admin) {
81         return
82     }
83
84     $client.sendMessage(EVENT.ADMIN.UNLOCK, { resource })
85 },
86
87 toggleLock(_, resource: AdminLockResource) {
88     if (accessor.isLocked(resource)) {
89         accessor.unlock(resource)
90     } else {
91         accessor.lock(resource)
92     }
93 },
94
95 login(store, { displayname, password }: { displayname: string;
96     password: string }) {
97     accessor.setLogin({ displayname, password })
98     $client.login(password, displayname)
99 },
100
101 logout() {
102     accessor.setLogin({ displayname: '', password: '' })
103     set('displayname', '')
104     set('password', '')
105     $client.logout()
106 },
107 },
```

```
107   )
108
109   export const storePattern = {
110     state,
111     mutations,
112     actions,
113     getters,
114     modules: { video, chat, files, user, remote, settings, client,
115               emoji },
116   }
117
118   Vue.use(Vuex)
119
120   const store = new Vuex.Store(storePattern)
121   export const accessor = useAccessor(store, storePattern)
122
123   Vue.prototype.$accessor = accessor
124
125   declare module 'vue/types/vue' {
126     interface Vue {
127       $accessor: typeof accessor
128     }
129   }
130
131   export default store
```



Зураг 4.3: UI phase 2.1



Зураг 4.4: UI phase 2.2

4.4 Клиент

Веб хөтөч хийх үйл явцыг эхний алхмаас нь сурахыг зорьсон. Сурах ур чадвар:

- Алхам алхмаар нарийн сайн төлөвлөх
- Шинэ программыг хурдан сурах
- Minimalist design сурах

Апп-ын шаардлага:

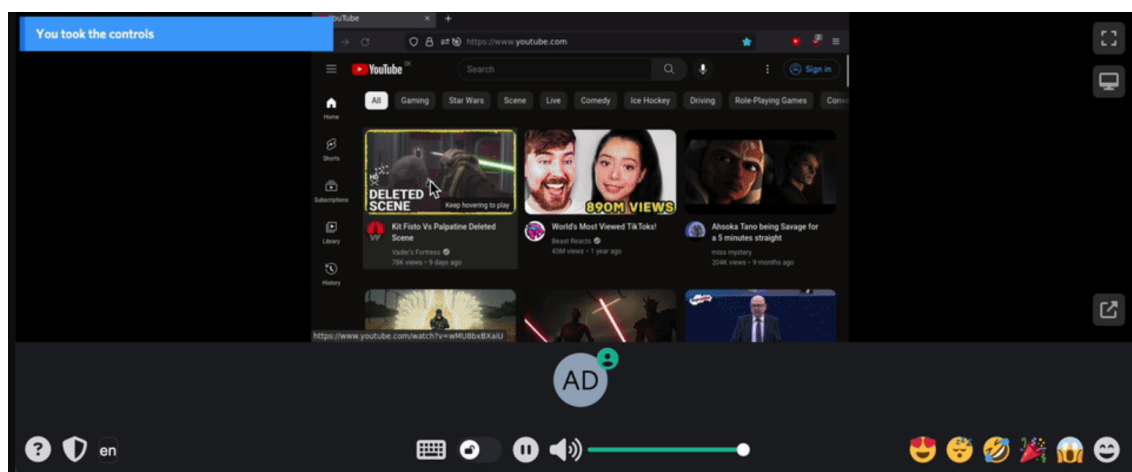
- Visual Studio ашиглах
- Ямар ч ажлын байран дээр тохиромжтой байх
- Бүх насныхан ашиглахад тохиромжтой байх

4.4.1 Гол дэлгэцлэх хэсэг

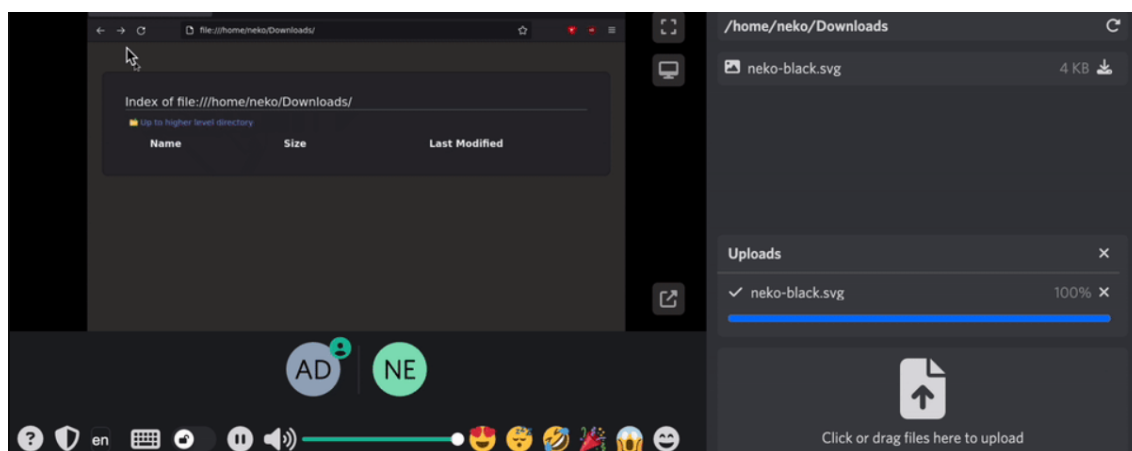
```
1  export function makeid(length: number) {
2      let result = ''
3      const characters = '
          ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 '
4      const charactersLength = characters.length
5      for (let i = 0; i < length; i++) {
6          result += characters.charAt(Math.floor(Math.random() *
              charactersLength))
7      }
8      return result
9  }
10
11 export function lockKeyboard() {
12     if (navigator && navigator.keyboard) {
13         navigator.keyboard.lock()
14     }
15 }
16
17 export function unlockKeyboard() {
18     if (navigator && navigator.keyboard) {
19         navigator.keyboard.unlock()
20     }
21 }
22
23 export function elementRequestFullscreen(el: HTMLElement) {
24     if (typeof el.requestFullscreen === 'function') {
```

```
25     el.requestFullscreen()
26     //@ts-ignore
27   } else if (typeof el.webkitRequestFullscreen === 'function') {
28     //@ts-ignore
29     el.webkitRequestFullscreen()
30     //@ts-ignore
31   } else if (typeof el.webkitEnterFullscreen === 'function') {
32     //@ts-ignore
33     el.webkitEnterFullscreen()
34     //@ts-ignore
35   } else if (typeof el.mozRequestFullScreen === 'function') {
36     //@ts-ignore
37     el.mozRequestFullScreen()
38     //@ts-ignore
39   } else if (typeof el.msRequestFullScreen === 'function') {
40     //@ts-ignore
41     el.msRequestFullScreen()
42   } else {
43     return false
44   }
45   return true
46 }
47
48 export function isFullscreen(): boolean {
49   return (
50     document.fullscreenElement ||
51     //@ts-ignore
52     document.msFullscreenElement ||
```

```
53     //@ts-ignore
54     document.mozFullScreenElement ||
55     //@ts-ignore
56     document.webkitFullscreenElement
57 )
58 }
59
60 export function onFullscreenChange(el: HTMLElement, fn: () => void) {
61     if (el.onfullscreenchange === null) {
62         el.onfullscreenchange = fn
63         //@ts-ignore
64     } else if (el.onmsfullscreenchange === null) {
65         //@ts-ignore
66         el.onmsfullscreenchange = fn
67         //@ts-ignore
68     } else if (el.onmozfullscreenchange === null) {
69         //@ts-ignore
70         el.onmozfullscreenchange = fn
71         //@ts-ignore
72     } else if (el.onwebkitfullscreenchange === null) {
73         //@ts-ignore
74         el.onwebkitfullscreenchange = fn
75     }
76 }
```



Зураг 4.5: UI phase 2.3



Зураг 4.6: UI phase 2.4

5. ДҮГНЭЛТ

5.1 Үр дүнгийн тайлан

Уг хугацаанд хичээлийн хүрээнд үзсэн онолын ойлголтуудыг практик дээр туршиж, хэрэгжүүлсэн ба Network голчилсон технологийн компанийн ерөнхий үйл ажиллагаа, баг хооронд зохицон ажиллах чадвар, хөгжүүлэлтийн шинэ арга барилуудыг амжилттай суралцсан гэж дүгнэж байна.

Том асуудлыг олон болгон хувааж ажиллах, алхам дэс дараатай асуудлыг шийдвэрлэх мөн ашиглаж буй сан, технологийнхоо гарын авлага буюу documentation-тай илүү сайн танилцаж уг технологийнхоо цаана нь буй гол агуулгыг хялбараар ойлгох гэх мэт чадваруудыг эзэмшсэн. Үүнийгээ цаашид илүү хөгжүүлж мэргэшсэн IT engineer болох болно. Үүнээс гадна сүлжээний аюулгүй байдал гэдэг ухагдахууныг маш чухал гэдгийг сайн ойлгосон.

Өөртөө тавьсан зорилгодоо хүрсэндээ баяртай байгаа боловч цаашид сурах мэдэх ёстой зүйл маш их байгааг ойлгосон.

НОМ ЗҮЙ

- [1] Visual Studio ерөнхий мэдлэг
url<https://visualstudio.microsoft.com/vs/getting-started/>
- [2] Web browser Заавар
url<https://www.wikihow.com/Make-a-Web-Browser>
- [3] Заавар хичээл
url<https://www.youtube.com/watch?v=HJN8uvtn850>