

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

Баянжаргалын Энх-Амгалан

**Бизнесийн байгууллагын ажилтны гүйцэтгэлийг
үнэлэх систем**
**(Employee performance evaluation system for business
organization)**

Мэдээллийн технологи(D061303)
Бакалаврын судалгааны ажил

Улаанбаатар

2025 оны 5 сар

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

Бизнесийн байгууллагын ажилтны гүйцэтгэлийг үнэлэх
систем

(Employee performance evaluation system for business
organization)

Мэдээллийн технологи(D061303)
Бакалаврын судалгааны ажил

Удирдагч: _____ Б. Энхтуул

Гүйцэтгэсэн: _____ Б. Энх-Амгалан (21B1NUM0344)

Улаанбаатар

2025 оны 5 сар

Зохлогчийн баталгаа

Миний бие Баянжаргалын Энх-Амгалан ”Бизнесийн байгууллагын ажилтны гүйцэтгэлийг үнэлэх систем” 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 дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

УДИРТГАЛ	1
БҮЛГҮҮД	2
1. СИСТЕМИЙН ТАНИЛЦУУЛГА	2
1.1 Системийн зорилго	2
1.2 Системийн зорилт	2
2. СИСТЕМИЙН СУДАЛГАА	4
2.1 Ижил төстэй системүүдийг судлах	4
2.2 Технологийн судалгаа	8
2.3 Бүлгийн дүгнэлт	13
3. СИСТЕМИЙН ШИНЖИЛГЭЭ	14
3.1 Системийн хэрэглэгчид	14
3.2 Функциональ шаардлага	14
3.3 Технологийн шаардлага	17
3.4 Функциональ загвар	18
3.5 Функциональ загварын тодорхойлолт	19
3.6 Бүлгийн дүгнэлт	19
НОМ ЗҮЙ	21

ЗУРГИЙН ЖАГСААЛТ

3.1 Системийн Функциональ Загварын Диаграмм	19
---	----

ХҮСНЭГТИЙН ЖАГСААЛТ

2.1	Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт	6
2.2	Ижил төстэй системүүдийн ”Тийм/Үгүй” харьцуулалт	7

Кодын жагсаалт

УДИРТГАЛ

Бизнесийн байгууллагуудын өрсөлдөх чадвар, амжилт нь ажилтнуудын гүйцэтгэлээс ихээхэн хамаардаг. Ажилтны гүйцэтгэлийг үнэлэх нь байгуулла зорилгодоо хүрэх, бүтээмжийг нэмэгдүүлэхэд чухал үүрэгтэй. Энэхүү судалгааны ажлын зорилго нь бизнесийн байгууллагад зориулсан ажилтны гүйцэтгэлийн үнэлгээний системийг вебд суурилан бүтээхэд оршино.

Энэ хүрээнд Next.js болон Golang хэл дээр суурилсан веб апп-ийг хөгжүүлсэн бөгөөд уг систем нь төслийн удирдлага, даалгаврын менежмент, ажилтны гүйцэтгэлийн үнэлгээний систем зэргийг нэгтгэсэн болно. Тус систем нь удирдлага болон ажилтнуудын хамтын ажиллагааг дэмжиж, даалгаврын хуваарилалт, гүйцэтгэлийн хяналт, үнэлгээний процессыг автоматжуулан, илүү үр дүнтэй, шударга системийг бий болгохыг зорьдог. Судалгаагаар энэхүү системийн онолын загвар, хэрэгжилт, удирдлагын арга барилд үзүүлэх нөлөөг авч үзнэ.

Энэхүү ажлын үр дүнд бизнесийн байгууллагын удирдах албан тушаалтан болон хүний нөөцийн мэргэжилтнүүдэд ажилтны чадавхийг нээн илрүүлэх, гүйцэтгэлийг дээшлүүлэхэд чиглэсэн шийдвэр гаргалтанд дэмжлэг үзүүлэхэд технологийн дэвшилтийг ашиглахад оршино.

1. СИСТЕМИЙН ТАНИЛЦУУЛГА

1.1 Системийн зорилго

Энэхүү дипломын ажлаар бизнесийн байгууллагын ажилтны гүйцэтгэлийн үнэлгээний системийг хөгжүүлж, байгууллагын ажилтны гүйцэтгэлийг автоматжуулан, ил тод, үр ашигтайгаар хэмжих боломжтой веб-д суурилсан систем хөгжүүлэх зорилготой. Энэхүү систем нь менежерүүд болон ажилтнуудын ажлыг хөнгөвчилж, гүйцэтгэлийн хяналтыг хурдан бөгөөд найдвартай болгох, улмаар байгууллагын бүтээмжийг нэмэгдүүлэхэд чиглэгдэнэ. Зорилгын хүрээнд дараах зүйлс багтана:

- **Автоматжуулсан гүйцэтгэлийн үнэлгээ хийх:** Систем нь KPI-д суурилсан үнэлгээг бодит цагийн мэдээлэл ашиглан автоматжуулж, гараар хийх ачааллыг бууруулна.
- **Автоматжуулсан гүйцэтгэлийн үнэлгээ хийх:** Систем нь менежерүүдийн ажилчдаа ү
- **Даалгавар хуваарилах, хянах:** Менежерүүд даалгавар хуваарилах, түүний гүйцэтгэлийг хянах, тайлан боловсруулалтыг хялбарчилах боломжийг олгоно.
- **Хэрэглэгчийн эрх удирдах, мэдээлэл хамгаалах:** Администраторууд хэрэглэгчийн эрхийг удирдаж, мэдээллийн аюулгүй байдлыг хангах боломжтой болно.

1.2 Системийн зорилт

Эдгээр зорилтууд нь ажилтны гүйцэтгэлийн үнэлгээний системийг хэрэгжүүлэх программ хангамжийн үе шаттай үйлдлүүдээс бүрдэнэ:

1. Сэдэвтэй холбоотой судалгаа хийх: Ижил төстэй системүүд (жишээ нь, Interactive.mn, Asana) болон технологийн судалгааг гүйцэтгэх.

2. Системийг хэрэглэх боломжит хэрэглэгчдийн шаардлага тогтоох: Администратор, Менежер, Ажилтан гэсэн хэрэглэгчдийн хэрэгцээ, шаардлагыг тодорхойлох.
3. Шаардлагатай уялдуулан системийн зохиомж гаргах: Функциональ болон технологийн шаардлагад нийцүүлэн системийн архитектур, загварыг боловсруулах.
4. Зохиомжийн дагуу системийг хөгжүүлэх: Next.js, Go (Gin, GORM), PostgreSQL, Docker зэрэг технологийг ашиглан системийг хэрэгжүүлэх.
5. Хөгжүүлэлтийг туршиж, алдааг засаж, сайжруулах: Системийн гүйцэтгэлийг туршиж, хэрэглэгчийн саналд үндэслэн сайжруулалт хийх.

Эдгээр зорилго, зорилтууд нь байгууллага доторх ажилтнуудын гүйцэтгэлийн хяналтыг сайжруулах, менежментийн шийдвэр гаргалтыг дэмжихэд чиглэгдэнэ.

2. СИСТЕМИЙН СУДАЛГАА

Энэ бүлэгт бакалаврын судалгааны ажлын хүрээнд тавьсан зорилго, зорилтын хүрээнд ижил төстэй системийн судалгаа мөн уг системийг хөгжүүлэхэд ашиглагдах технологиудын судалгаа хийв.

2.1 Ижил төстэй системүүдийг судлах

Эндээс бид дипломын ажлын хүрээнд хөгжүүлж буй системтэй ижил төстэй үйл ажиллагаа явуулдаг хоёр системийг судалж, тэдгээрийг харьцуулна. Судалгаанд Interactive.mn-ийн бүтээгдэхүүн болон Asana-ийн онцлогуудыг авч үзэх бөгөөд эдгээр системүүдийн байгууллагын танилцуулга, ажилтны гүйцэтгэлийг үнэлэх үндсэн модулиудыг тодорхойлно. Дараа нь эдгээр системүүдийг хөгжүүлж буй системтэй харьцуулж, давуу тал, онцлогийг шинжилнэ.

2.1.1 Interactive.mn-ийн Бүтээгдэхүүн

- **Байгууллагын товч танилцуулга:** Interactive.mn нь Монгол Улсад байрладаг технологийн компани бөгөөд бизнесийн байгууллагуудад зориулсан программ хангамжийн шийдэл санал болгодог. Тус компанийн гол бүтээгдэхүүн нь байгууллагын дотоод үйл ажиллагааг автоматжуулах, ажилтнуудын гүйцэтгэлийг хянахад чиглэсэн системүүдийг хамардаг. Вэбсайтаас (<https://interactive.mn/product>) харахад тэдний шийдэл нь хэрэглэгчдэд ээлтэй интерфэйс, өөрчлөн тохируулах боломжоороо онцлогтой.
- **Ажилтны гүйцэтгэлийг үнэлэх үндсэн модуль:** Interactive.mn-ийн систем нь ажилтнуудын гүйцэтгэлийг хянахад чиглэсэн модультай бөгөөд энэ нь ажлын төлөвлөгөө, гүйцэтгэлийн үзүүлэлтүүдийг (KPI) тодорхойлох, тогтмол тайлагнах боломжийг олгодог. Модуль нь менежерүүдэд ажилтны ажлын ахиц, үр дүнг хянах боломж олгодог бөгөөд энгийн тайлангуудыг үүсгэх боломжтой. Гэсэн хэдий ч тодорхой мэдээлэл хязгаарлагдмал тул уг модуль нь голчлон энгийн KPI-д суурилсан үнэлгээнд чиглэдэг гэж дүгнэж болно.

2.1. ИЖИЛ ТӨСТЭЙ СИСТЕМҮҮДИЙГ СУДЛАХ ТЭГ 2. СИСТЕМ ИЙН СУДАЛГАА

2.1.2 Asana-ийн Онцлогууд

- **Байгууллагын товч танилцуулга:** Asana нь АНУ-ын Сан Франциско хотод байрладаг программ хангамжийн компани бөгөөд 2008 онд Дустин Москович болон Жастин Розенштейн нар үүсгэн байгуулжээ (<https://asana.com/features>). Тус компани нь багуудын ажлыг зохион байгуулах, хянах, удирдахад зориулсан SaaS платформыг санал болгодог. 2025 оны байдлаар Asana нь дэлхий даяар 131,000 гаруй хэрэглэгчтэй бөгөөд томоохон байгууллагууд (Uber, IBM гэх мэт)-д өргөн хэрэглэгддэг.
- **Ажилтны гүйцэтгэлийг үнэлэх үндсэн модуль:** Asana нь шууд гүйцэтгэлийн үнэлгээний модульгүй боловч "Goals" (Зорилго) онцлог болон тасгийн хяналтын системээр дамжуулан ажилтны гүйцэтгэлийг хянах боломжтой. Хэрэглэгчид тодорхой ажлуудыг хуваарилж, хугацаа тогтоож, ахицыг хянах боломжтой бөгөөд энэ нь ажилтны гүйцэтгэлийг шууд бусаар үнэлэхэд ашиглагддаг. Нэмж дурдахад, Asana-ийн тайлагнах хэрэгслүүд нь баг болон хувь хүний гүйцэтгэлийн статистикийг гаргахад тусалдаг.

2.1.3 Харьцуулалт

Эндээс бид Interactive.mn болон Asana-ийн системүүдийг хөгжүүлж буй системтэй харьцуулна. Энэхүү систем нь Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), PostgreSQL, Docker зэрэг технологийг ашиглан хөгжүүлэгдэж байгаа бөгөөд ажилтны гүйцэтгэлийг бодит цагийн мэдээлэлд суурилан үнэлэхэд чиглэсэн модультай.

2.1. ИЖИЛ ТӨСТЭЙ СИСТЕМҮҮДИЙГ СУДЛАХЛАГ 2. СИСТЕМИЙН СУДАЛГАА

Дэлгэрэнгүй харьцуулалтын хүснэгт

Онцлог	Interactive.mn	Asana	EPES
Танилцуулга	Монголын технологийн компани, бизнесийн шийдэл	АНУ-ын SaaS платформ, баг хамтын ажиллагаа	Дипломын ажлын хүрээнд хөгжүүлэгдсэн вэб систем
Үнэлгээний модуль	KPI-д суурилсан энгийн үнэлгээ	Таскийн хяналт, зорилго тогтоох	Бодит цагийн мэдээлэлд суурилсан үнэлгээ
Технологи	Тодорхой бус (вэбд суурилсан)	JavaScript, React гэх мэт	Golang, Next.js, PostgreSQL
Интерфэйс	Хэрэглэгчдэд ээлтэй	Маш сайн UI/UX	Tailwind CSS-ээр хариу үйлдэлтэй
Бодит цагийн хяналт	Хязгаарлагдмал	Хэсэгчлэн дэмждэг	Бүрэн дэмждэг
Тохируулга	Дунд зэрэг	Өндөр	Өндөр (нээлттэй эх)
Аюулгүй байдал	Тодорхой бус	JWT болон HTTPS	JWT ашигласан
Хэрэглээний хүрээ	Жижиг, дунд бизнес	Том байгууллага	Төрөл бүрийн байгууллага

Хүснэгт 2.1: Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт

Системүүдийн харьцуулалтын хүснэгт

Системүүдийн онцлогуудыг илүү тодорхой харуулахын тулд "Тийм" эсвэл "Үгүй" хариултад суурилсан харьцуулалтыг доорх хүснэгтэд үзүүлэв:

2.1. ИЖИЛ ТӨСТЭЙ СИСТЕМҮҮДИЙГ СУДЛАХ ТЭГ 2. СИСТЕМ ИЙН СУДАЛГАА

Онцлог	Interactive.mn	Asana	EPES
Ажилтны гүйцэтгэлийг шууд үнэлэх модультай юу?	Тийм	Үгүй	Тийм
Бодит цагийн мэдээлэлд суурилсан уу?	Үгүй	Үгүй	Тийм
Нээлттэй эхийн технологи ашигласан уу?	Үгүй	Үгүй	Тийм
KPI-д суурилсан үнэлгээ хийдэг үү?	Тийм	Хэсэгчлэн	Тийм
Таскийн хяналтын системтэй юу?	Тийм	Тийм	Тийм
Хэрэглэгчийн интерфэйс хариу үйлдэлтэй юу?	Тийм	Тийм	Тийм
JWT аюулгүй байдлын стандарт ашигладаг уу?	Тодорхой бус	Тийм	Тийм
Docker-ээр контейнержсэн уу?	Үгүй	□ Үгүй	Тийм

Хүснэгт 2.2: Ижил төстэй системүүдийн ”Тийм/Үгүй” харьцуулалт

Шинжилгээ:

- **Interactive.mn:** Энгийн KPI-д суурилсан үнэлгээний модультай боловч бодит цагийн хяналт, нээлттэй эхийн технологи дутагдалтай. Жижиг, дунд бизнесүүдэд тохиромжтой ч том хэмжээний өргөтгөлд хязгаарлалттай.
- **Asana:** Таскийн хяналт, зорилго тогтоох онцлог нь гүйцэтгэлийг шууд бусаар үнэлэх

боломжтой ч шууд модульгүй. Том байгууллагуудад тохиромжтой, гэхдээ бодит цагийн шинэчлэлт хязгаарлагдмал.

- **EPES:** Бодит цагийн мэдээлэлд суурилсан үнэлгээний модуль, нээлттэй эхийн технологи, Docker-ийн дэмжлэгээрээ онцлог. Энэ нь илүү уян хатан, тохируулгатай боловч хөгжүүлэлтийн эхний шатандаа тул том хэмжээний туршилт шаардлагатай.

2.2 Технологийн судалгаа

Энэ хэсэгт системд ашиглагдах технологийн талаар дэлгэрэнгүй судалгаа хийж, тэдгээрийн онолын үндэс, практик хэрэглээ, давуу тал, сул тал шинжилнэ. Судалгаа нь Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), Docker, Postman, DBeaver, PostgreSQL зэрэг технологийг хамарна. Эдгээр хэрэгслүүд нь орчин үеийн вэб хөгжүүлэлтийн шаардлагыг хангахад чиглэсэн бөгөөд дипломын ажлын зорилгод нийцүүлэн сонгосон болно.

Энэхүү судалгааны гол зорилго нь дипломын ажлын хүрээнд сонгосон технологийн гүйцэтгэл, өргөтгөх чадвар, хөгжүүлэлтийн үр ашгийг үнэлэх явдал юм. Технологи тус бүр ямар онолын үндэслэлээр ажилладаг, яагаад энэ системд тохиромжтой вэ гэдгийг тодорхойлж, практик хэрэглээний жишээнүүдийг судална.

2.2.1 Golang

Golang (Go) нь Google-ийн 2009 онд танилцуулсан статик төрөлтэй, хөрвүүлэгддэг програмчлалын хэл бөгөөд өндөр гүйцэтгэл, тогтвортой байдлыг хангахад чиглэдэг.

Gin

- Gin нь Go-д зориулсан хөнгөн, өндөр гүйцэтгэлтэй HTTP вэб фреймворк юм. HTTP чиглүүлэлтийг Radix модны алгоритмаар хурдан гүйцэтгэдэг бөгөөд энэ нь хүсэлт боловсруулалтын хугацааг багасгадаг. Ингэснээр RESTful API-уудыг хялбархан бий болгох боломжийг олгодог бөгөөд route, middleware дэмжлэг, хүсэлтийн боловсруулалтыг хангадаг.

- **Давуу тал:**
 - Хамгийн бага зардалтай, хурдан ажиллагаатай.
 - Микро үйлчилгээ болон API-д суурилсан програмуудад тохиромжтой.
 - Middleware-ийн дэмжлэгээр аюулгүй байдал, лог хөтлөлтийг хялбаршуулна.
- **Сул тал:** Нарийн тохиргоо шаардлагагүй тул жижиг төслүүдэд хэт хүнд байж болно.
- **Хэрэглээ:** RESTful API бүтээхэд тохиромжтой бөгөөд системийн backend хэсгийг хөгжүүлэхэд ашиглагдана.

GORM

- GORM нь Go-д зориулсан ORM (Object-Relational Mapping) сан бөгөөд PostgreSQL өгөгдлийн сан хоорондын харилцаа болон хөрвүүлэлтийг автоматжуулна.
- **Давуу тал:**
 - SQL бичих хугацааг хэмнэж, хөгжүүлэлтийг хурдасгана. Migrations, queries зэрэг үйлдлийг хялбаршуулна.
 - Transactions, preloading, hooks зэрэг дэвшилтэт технологи ашигладаг.
 - Автомат миграци нь мэдээллийн сангийн схемийн өөрчлөлтийг хялбаршуулна.
- **Сул тал:** Нарийн SQL асуулгад сул талтай тул зарим тохиолдолд SQL бичих шаардлага гарна.
- **Хэрэглээ:** PostgreSQL-тэй хослуулан мэдээллийн сангийн удирдлагыг хэрэгжүүлнэ.

JWT

- JWT нь криптографийн HMAC алгоритмаар токены агуулгыг баталгаажуулдаг бөгөөд төлөвгүй (stateless) баталгаажуулалтыг хангадаг.

- **Давуу тал:**
 - Сервер дээр сесси хадгалах шаардлагагүй тул өргөтгөхөд хялбар.
 - REST API-д нийцтэй.
- **Сул тал:** Токеныг хулгайлах эрсдэл бий тул HTTPS-ийн хамт ашиглах ёстой.
- **Хэрэглээ:** Хэрэглэгчийн нэвтрэлтийг баталгаажуулахад ашиглагдана.

2.2.2 *Next.js ба Tailwind CSS*

Next.js нь React-д суурилсан фреймворк бөгөөд сервер талын рендерингийг дэмждэг бол Tailwind CSS нь хурдан UI хөгжүүлэлтэд чиглэнэ.

- Next.js нь виртуал DOM болон SSR-ийн хослолоор ажилладаг бөгөөд энэ нь хуудасны ачааллыг хурдасгадаг.
- Tailwind нь utility-first зарчмаар CSS-ийг урьдчилан тодорхойлж, загварчлалын хугацааг багасгадаг.

Давуу тал

- Next.js: SEO-д ээлтэй, хэрэглэгчийн туршлагыг сайжруулна.
- Tailwind: Кодын давхцыг багасгаж, загварыг хялбар өөрчлөх боломжтой.

Сул тал

- Next.js: SSR нь серверын ачааллыг нэмэгдүүлнэ.
- Tailwind: Том төслүүдэд классын удирдлага төвөгтэй болж болно.

Хэрэглээ

Дипломын ажлын хүрээнд фронтенд хөгжүүлэлтэд ашиглаж, хэрэглэгчийн интерфэйсийг хариу үйлдэлтэй болгоно.

2.2.3 Docker

- Контейнержуулалтын технологи бөгөөд OS-ийн виртуалчлалын зарчмаар ажилладаг.
- **Давуу тал:**
 - Орчны тогтвортой байдлыг хангана.
 - Хувилбарын хяналтыг хялбаршуулна.
- **Сул тал:** Нөөцийн хэрэглээ ихтэй тул жижиг төслүүдэд тохиромжгүй байж болно.
- **Хэрэглээ:** PostgreSQL болон програмын бүрэлдэхүүнийг контейнерт ажиллуулж, хөгжүүлэлтийн орчныг стандартчилна.

2.2.4 Postman

- API-ийн туршилтын автоматжуулалт ба REST архитектурын стандартыг дагадаг.
- **Давуу тал:**
 - API-ийн хариуг хурдан шалгана.
 - Хамтын ажиллагааг дэмжинэ.
- **Сул тал:** Том төслүүдэд автоматжуулалтын скриптүүд нарийн болно.
- **Хэрэглээ:** Gin-ийн API-ийн гүйцэтгэлийг туршихад ашиглана.

2.2.5 DBeaver

- Мэдээллийн сангийн GUI удирдлага бөгөөд SQL стандартыг дэмждэг.
- Давуу тал:
 - Мэдээллийн сангийн схемийг графикаар харах боломжтой.
 - Асуулгын дибаг хийхэд хялбар.
- Сул тал: Том хэмжээний өгөгдөлтэй ажиллахад удаан байж болно.
- Хэрэглээ: PostgreSQL-ийн удирдлага, GORM-ийн үр дүнг шалгахад ашиглана.

2.2.6 PostgreSQL

- Харилцааны мэдээллийн сангийн онол дээр суурилсан бөгөөд ACID зарчмыг хангана.
- Давуу тал:
 - JSON дэмжлэгтэй тул нийлмэл өгөгдөл хадгална.
 - Индексжүүлэлтээр гүйцэтгэл сайн.
- Сул тал: Том хэмжээний ачаалалд тохиргоо шаардлагатай.
- Хэрэглээ: Дипломын ажлын мэдээллийн сангийн үндэс болно.

2.2.7 Судалгааны Дүгнэлт

Эдгээр технологиуд нь дипломын ажлын хүрээнд өндөр гүйцэтгэлтэй, өргөтгөх боломжтой веб програм хөгжүүлэхэд тохиромжтой гэдэг нь тодорхойлогдлоо. Golang-ийн хурд, Next.js-ийн хэрэглэгчийн туршлага, Docker-ийн тогтвортой байдал нь төслийн амжилтад хувь нэмэр оруулна. Гэсэн хэдий ч эдгээрийг хэрэгжүүлэхэд сургалт, нөөцийн удирдлага шаардлагатай бөгөөд энэ нь төслийн төлөвлөлтөд анхаарах ёстой хүчин зүйл болно.

2.3 Бүлгийн дүгнэлт

Эндээс Interactive.mn болон Asana-ийн системүүдийг дипломын ажлын хүрээнд хөгжүүлж буй системтэй харьцуулан судалсны үр дүнг нэгтгэн дүгнэв. Interactive.mn-ийн бүтээгдэхүүн нь Монголын жижиг болон дунд байгууллагуудад чиглэсэн, локал хэрэглээнд тохирсон шийдэл болохыг тогтоосон боловч технологийн хувьд хязгаарлагдмал бөгөөд том хэмжээний байгууллагуудын шаардлагыг хангахад хангалтгүй байж болно. Харин Asana нь олон улсын хэмжээнд том байгууллагуудыг дэмжих, өргөн хүрээний интеграцитай, ажлын удирдлагад чиглэсэн хүчирхэг систем болох нь тодорхойлогдсон ч жижиг багуудад хэт нарийн, өндөр өртөгтэй байж болзошгүй юм.

Хөгжүүлж буй систем нь эдгээр хоёр системийн давуу талыг хослуулсан хэлбэртэй бөгөөд жижиг, дунд байгууллагуудад хямд, хялбаршуулсан шийдэл санал болгохын зэрэгцээ Golang, Next.js, Docker зэрэг орчин үеийн технологиудыг ашигласнаар гүйцэтгэл, өргөтгөх чадварыг хангана. Interactive.mn-тэй харьцуулахад миний систем илүү модульчлагдсан, PostgreSQL болон GORM-ийн тусламжтайгаар мэдээллийн сангийн бат бөх удирдлагатай бол Asana-тай харьцуулахад илүү хямд, локал хэрэглээнд тохирсон онцлогтой. Гэсэн хэдий ч хөгжүүлж буй системд Asana-ийн "Workload" эсвэл "Goals" гэх мэт нарийн функцууд дутагдаж байгаа нь том байгууллагуудад хэрэглэхэд хязгаарлалт болж болзошгүй.

Эцэст нь, энэхүү судалгаа нь хөгжүүлж буй системийн давуу тал болох хямд байдал, хялбаршуулсан интерфэйс, технологийн бат бөх байдлыг онцолж, зах зээл дээрх ижил төстэй системүүдээс ялгарах боломжийг харууллаа. Цаашид системийн хөгжүүлэлтэд Interactive.mn-ийн локал тохируулга, Asana-ийн аналитик хэрэгслүүдийг тусгах нь илүү өрсөлдөх чадвартай шийдэл болоход тусална гэж дүгнэж байна.

3. СИСТЕМИЙН ШИНЖИЛГЭЭ

Энэхүү бүлэгт дипломын ажлын хүрээнд хөгжүүлж буй системийн шинжилгээг дэлгэрэнгүй хийж, түүний хэрэглэгчид, шаардлагууд, загварыг тодорхойлно. Системийн шинжилгээ нь хөгжүүлэлтийн үндсэн суурь болж, систем хэрхэн ажиллах, ямар зорилгоор ашиглагдахыг тодорхойлоход чиглэнэ. Энэ бүлэг нь дараах хэсгүүдээс бүрдэнэ: Системийн хэрэглэгч, Функциональ шаардлага, Технологийн шаардлага, Функциональ загвар, Функциональ загварын тодорхойлолт, Бүлгийн дүгнэлт.

3.1 Системийн хэрэглэгчид

- **Админ:** Системийн тохиргоо, хэрэглэгчийн удирдлага, мэдээллийн сангийн засвар үйлчилгээг хариуцна. Тэдэнд системийн бүрэн хандалт шаардлагатай.
- **Менежер:** Ажилтнуудын гүйцэтгэлийг хянах, тайлан гаргах, даалгавар хуваарилах үүрэгтэй. Тэдэнд аналитик хэрэгсэл, хяналтын самбар хэрэгтэй.
- **Хүнийн нөөцийн мэргэжилтэн**
- **Ажилтан:** Даалгаврыг гүйцэтгэх, хувийн гүйцэтгэлийн мэдээлэл харах боломжтой хэрэглэгчид. Тэдэнд хялбар интерфэйс, тодорхой заавар хэрэгтэй.

Эдгээр хэрэглэгчид нь системийн үндсэн үйл ажиллагааг тодорхойлох бөгөөд тэдний хэрэгцээ нь функциональ болон технологийн шаардлагыг хангахад чиглэнэ.

3.2 Функциональ шаардлага

3.2.1 Функциональ Шаардлага

Функциональ шаардлага нь системийн гол үйл ажиллагааг тодорхойлж, хэрэглэгчдэд ямар боломж олгохыг заана. Эдгээр шаардлага нь системийн зорилгыг хангаж, хэрэглэгчийн туршлагыг сайжруулан, гүйцэтгэлийн үнэлгээний процессыг автоматжуулна. Шаардлагуудыг ерөнхий

болон хэрэглэгчийн төрлүүдээр (Администратор, Менежер, Ажилтан) ангилж, доорх хүснэгтэд дэлгэрэнгүй харуулав. Шаардлага бүрт код оноож, тодорхойлолтыг нарийвчлан өгснөөр хөгжүүлэлтийн явцад тодорхой бус байдлыг багасгана.

Ерөнхий Шаардлага

Ерөнхий шаардлагууд нь системийн суурь үйл ажиллагааг хамардаг бөгөөд бүх хэрэглэгчидтэй холбоотой үндсэн функцуудыг тодорхойлно. Эдгээр нь аюулгүй байдал, хэрэглэгчийн бүртгэл, мэдээллийн хадгалалтад чиглэнэ.

|p2cm|p8.5cm|

Код Функциональ Шаардлага

ФШ100 Хэрэглэгч бүртгэх: Систем шинэ хэрэглэгчдийг бүртгэнэ (имэйл, нэр, дүр). Давхардлыг шалгана.

ФШ101 Нэвтрэлтийн баталгаажуулалт: Хэрэглэгчид имэйл, нууц үгээр нэвтэрнэ. JWT ашиглана.

ФШ102 Нууц үг сэргээх: Хэрэглэгч нууц үгээ мартвал имэйлээр сэргээх холбоос илгээнэ (хугацаатай токен).

ФШ103 Системийн лог: Хэрэглэгчийн үйлдэл, алдааг PostgreSQL-д тэмдэглэнэ.

ФШ104 Мэдээлэл хадгалах: Хэрэглэгчийн профайл (нэр, имэйл, дүр) PostgreSQL-д хадгалагдаж, засварлагдана.

ФШ105 Сессийн удирдлага: JWT токеноор сессийг хянаж, 30 минут идэвхгүй бол дуусгана.

Администраторын Шаардлага

Администраторын шаардлагууд нь системийн удирдлага, аюулгүй байдал, засвар үйлчилгээтэй холбоотой бөгөөд тогтвортой байдлыг хангадаг.

|p2cm|p8.5cm|

Код Функциональ Шаардлага

ФШ200 Эрхийн удирдлага: Администратор хэрэглэгчдэд дүр (Admin, Manager, Employee) оноож, хандалтыг тохируулна.

ФШ201 Системийн тохиргоо: Параметрууд (KPI-ийн жин, мэдэгдлийн давтамж) Next.js интерфэйсээр засварлагдана.

ФШ202 Нөөцлөлт, сэргээлт: PostgreSQL-ийн мэдээллийг өдөр бүр нөөцөлж, Docker-д сэргээнэ.

ФШ203 Үйлдлийн түүх: Нэвтрэлт, үнэлгээний оруулалтыг хугацаа, ID-тай хянаж, тайлан гаргана.

ФШ204 Алдааны тайлан: API тасалдал, PostgreSQL холболтын алдааг тэмдэглэн, мэдэгдэнэ.

ФШ205 Хэрэглэгч идэвхгүй болгох: Хэрэглэгчийг хасах эсвэл түр идэвхгүй болгоно.

Менежерын Шаардлага

Менежерын шаардлагууд нь ажилтны гүйцэтгэлийг удирдах, хянах, тайлагнахад чиглэж, системийн гол зорилгыг хэрэгжүүлнэ.

|p2cm|p8.5cm|

Код Функциональ Шаардлага

ФШ300 Даалгавар хуваарилах: Менежер даалгавар үүсгэж, хугацаа, тодорхойлолттой хуваарилна.

ФШ301 Гүйцэтгэлийн хяналт: Даалгаврын статус (эхэлсэн, дууссан) бодит цагт хянагдана (жишээ: 75%).

ФШ302 KPI-д суурилсан үнэлгээ: Менежер KPI (хурд, чанар) оноо, санал оруулна.

ФШ303 Тайлан гаргах: Гүйцэтгэлийн түүх, KPI оноог график хэлбэрээр харуулна.

ФШ304 Тайлан экспорт: Тайланг PDF (график) эсвэл CSV (өгөгдөл) хэлбэрээр татна.

ФШ305 Харьцуулалт: Багийн гишүүдийн гүйцэтгэлийг KPI оноогоор харьцуулж, шилдэгийг тодорхойлно.

Ажилтны Шаардлага

Ажилтны шаардлагууд нь хувь хүний гүйцэтгэлийг хянах, даалгаврыг удирдахад чиглэж, идэвхийг дэмжинэ.

|p2cm|p8.5cm|

Код Функциональ Шаардлага

ФШ400 Даалгавар харах: Ажилтан хуваарилагдсан даалгавруудын жагсаалтыг (хугацаа, тодорхойлолт) харна.

ФШ401 Статус шинэчлэх: Даалгаврын явцыг (хийгдэж байна, дууссан) тэмдэглэнэ.

ФШ402 Гүйцэтгэл хянах: Ажилтан KPI оноо, саналыг график хэлбэрээр харна.

ФШ403 Мэдэгдэл: Даалгаврын хугацаа ойртох (24 цагийн өмнө) эсвэл хэтэрвэл мэдэгдэнэ.

ФШ404 Профайл засах: Ажилтан мэдээлэл (имэйл, зураг), нууц үгээ өөрчилнө.

ФШ405 Өөрийгөө үнэлэх: Ажилтан гүйцэтгэлээ үнэлж, санал оруулна (менежерт хянагдана).

3.3 Технологийн шаардлага

Технологийн шаардлага нь системийн хөгжүүлэлт, ажиллагааг хангахад шаардлагатай техник хангамж, програм хангамжийг тодорхойлно:

- **Серверын Орчин:** Docker контейнерт суурилсан PostgreSQL мэдээллийн сан болон Golang-д суурилсан API сервер.

- **Клиент Тал:** Next.js-д суурилсан вэб интерфэйс, Tailwind CSS ашиглан хариу үйлдэлтэй дизайн.
- **Холболт:** RESTful API ашиглан клиент-сервер хоорондын харилцааг хангана.
- **Туршилтын Хэрэгсэл:** Postman ашиглан API-ийн туршилт, DBeaver ашиглан мэдээллийн сангийн удирдлага.
- **Гүйцэтгэлийн Шаардлага:** Систем нь 100 хүртэлх нэгэн зэрэг хэрэглэгчийг 1 секундээс бага хариу өгөх хугацаатай дэмжих ёстой.

Эдгээр шаардлагууд нь системийн тогтвортой байдал, хурдыг хангахад чиглэнэ.

3.4 Функциональ загвар

Функциональ загвар нь системийн гол бүрэлдэхүүн хэсгүүд болон тэдгээрийн харилцан үйлчлэлийг тодорхойлж, систем хэрхэн ажиллахыг графикаар харуулна. Энэ загвар нь хөгжүүлэлтийн үндсэн суурь болж, системийн архитектурыг ойлгоход тусална. Загварт дараах бүрэлдэхүүн хэсгүүд багтана:

- **Хэрэглэгчийн Интерфэйс:** Next.js болон Tailwind CSS ашиглан хэрэглэгчийн харагдац, харилцан үйлдэл хийх хэсэг.
- **API Сервер:** Gin фреймворк ашиглан RESTful API endpoint-уудыг хангаж, клиент болон мэдээллийн сан хоорондын харилцааг зохицуулна.
- **Мэдээллийн Сан:** PostgreSQL дээр GORM ашиглан мэдээллийн хадгалалт, удирдлагыг гүйцэтгэнэ.
- **Контейнержуулалт:** Docker ашиглан сервер болон мэдээллийн сангийн орчныг стандартжуулж, тогтвортой байдлыг хангана.

Эдгээр бүрэлдэхүүн хэсгүүд хоорондоо HTTP хүсэлт (REST API) болон SQL асуулгаар харилцана.

Доорх диаграммд энэ харилцан үйлчлэлийг графикаар харуулав.

3.5. ФУНКЦИОНАЛЬ ЗАГВАРЫН ТОДОРХОЙЛОЛТ

Функциональ Загварын Диаграмм

Зураг 3.1: Системийн Функциональ Загварын Диаграмм

Эндээс хэрэглэгчээс эхэлсэн хүсэлт хэрэглэгчийн интерфэйсээр дамжин API сервер руу очиж, улмаар мэдээллийн сантай харилцан үйлчлэл үйлдэгдэнэ. Docker нь API сервер болон мэдээллийн сангийн орчныг стандартжуулж, тогтвортой байдлыг хангана. Хариу нь эсрэг чиглэлд буцаж, хэрэглэгчид харагдана.

3.5 Функциональ загварын тодорхойлолт

Функциональ загварыг илүү тодорхой болгохын тулд системийн ажиллагааг алхам алхмаар тайлбарлана:

1. Хэрэглэгч вэб интерфэйсээр нэвтэрнэ (JWT токен ашиглан баталгаажуулалт хийгдэнэ).
2. Менежер даалгавар үүсгэж, ажилтанд хуваарилна (Gin API-ээр дамжина).
3. Даалгаврын мэдээлэл PostgreSQL-д хадгалагдаж, GORM ашиглан удирдлага хийгдэнэ.
4. Ажилтан даалгаврыг гүйцэтгэж, статусыг шинэчилнэ (Next.js интерфэйсээр).
5. Систем гүйцэтгэлийн тайлан гаргаж, хэрэглэгчдэд харуулна.

Эндээс системийн гол процесс болох хэрэглэгчийн баталгаажуулалт, даалгаврын удирдлага, гүйцэтгэлийн хяналт тодорхойлогдож байна.

3.6 Бүлгийн дүгнэлт

Системийн шинжилгээний үр дүнд хэрэглэгчид (администратор, менежер, ажилтан)-ын хэрэгцээг хангах функциональ болон технологийн шаардлагууд тодорхойлогдлоо. Функциональ загвар нь системийн бүрэлдэхүүн хэсгүүдийн харилцан үйлчлэлийг тодорхой харуулж, хөгжүүлэлтийн

төлөвлөгөөнд суурь болно. Golang, Next.js, Docker зэрэг технологиудыг ашигласнаар систем нь хурдан, тогтвортой, хялбар хөгжүүлэлттэй байх боломжтой гэж дүгнэгдэв. Гэсэн хэдий ч системийн гүйцэтгэлийг олон хэрэглэгчийн орчинд турших, нөөцийн удирдлагыг сайжруулах шаардлага тавигдаж байна.

Bibliography

- [1] Daemen, J., & Rijmen, V. (2002). "The Design of Rijndael: AES - The Advanced Encryption Standard." Springer. p.1-2.
- [2] Д. Гармаа (2022). "Криптографын үндэс." Улаанбаатар хот.
- [3] Bellare, Mihir; Rogaway, Phillip (11 May 2005), Introduction to Modern Cryptography (Lecture notes), archived (PDF) from the original on 2023-10-30, chapter 3.
- [4] Simmons, G. J. (2022, December 29). PCA (RSA) encryption. Encyclopedia Britannica. [https://www.britannica.com/topic/PCA \(RSA\)-encryption](https://www.britannica.com/topic/PCA-(RSA)-encryption)
- [5] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., & Zimmermann, P. (2020, February). A 829-bit factorization. Retrieved from <https://members.loria.fr/PZimmermann/records/factor.html>
- [6] Mahto, Dindayal; YADAV, DILIP. (2017). RSA and ECC: A comparative analysis. International Journal of Applied Engineering Research, Vol. 12, pp. 9053-9061.