

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

Зохиогчийн 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 Онолын судалгаа	4
2.2 Ижил төстэй системүүд	11
2.3 Технологийн судалгаа	16
2.4 Бүлгийн дүгнэлт	20
3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ	22
3.1 Системийн хэрэглэгчид	22
3.2 Системийн шаардлага	22
3.3 Системийн загвар	31
3.4 Бүлгийн дүгнэлт	38
4. ХЭРЭГЖҮҮЛЭЛТ	39
4.1 Файлын бүтэц	39
4.2 Хөгжүүлэлтийн орчин бүрдүүлэлт	41
4.3 Шаардлагын хэрэгжүүлэлт	46
5. ДҮГНЭЛТ	82
НОМ ЗҮЙ	84
ХАВСРАЛТ	85
А. BACK-END ХЭРЭГЖҮҮЛЭЛТ	85

ЗУРГИЙН ЖАГСААЛТ

3.1	Нэвтрэх хуудас	26
3.2	Админ дашбоард харагдац	27
3.3	Ажилтны жагсаалт	27
3.4	Төслийн харагдац	28
3.5	Даалгаварын харагдац	28
3.6	Гүйцэтгэлийн үнэлгээний харагдац	29
3.7	Тайлан гаргах компонентийн харагдац	29
3.8	Хэрэглэгчийн профайлын харагдац	30
3.9	Системийн архитектурын диаграмм	31
3.10	Системийн ажлын явцын диаграмм	32
3.11	Төсөл болон даалгаврын харилцан хамаарлын диаграмм	33
3.12	Ажилтны нэгж хоорондын харилцан хамаарлын диаграмм	34
3.13	Ажилтан болон түүний эрхийн харилцан хамаарлын диаграмм	35
3.14	Админ дарааллын диаграмм	36
3.15	Менежер дарааллын диаграмм	37
3.16	Ажилтан дарааллын диаграмм	38
4.1	Front-end файлын бүтэц	39
4.2	Back-end файлын бүтэц	40
4.3	Docker Desktop програмын интерфэйс	41
4.4	Postman файлын бүтэц	45
4.5	Postman хэрэглэгч үүсгэх тест	46

ХҮСНЭГТИЙН ЖАГСААЛТ

2.1	Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт	13
2.2	Интеграцын харьцуулалт	14
3.1	Ерөнхий шаардлага	23
3.2	Админ шаардлага	23
3.3	Менежер шаардлага	24
3.4	Ажилтан шаардлага	24
3.5	Функциональ биш шаардлага	25
3.6	Функциональ биш шаардлага	25

Кодын жагсаалт

4.1	Dockerfile	41
4.2	docker-compose.yaml	42
4.3	Bun суулгах	43
4.4	Nextjs суулгах	43
4.5	Shaden суулгах	43
4.6	Golang суулгах	44
4.7	src/app/protected/kpi/page.tsx [front-end	46
4.9	Тайлан автоматаар гаргах	70
A.1	Routes	85
A.2	Models	88

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, түүний 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.
- **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 шатаас 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: Админ, 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

2.1 Онолын судалгаа

Орчин үеийн байгууллагууд ажилтны ур чадвар, ажлын чанар, байгууллагын зорилгод оруулж буй хувь нэмрийг үнэлэх зорилгоор гүйцэтгэлийн үнэлгээний системийг тогтмол хэрэгжүүлж байна. Гүйцэтгэлийн үнэлгээ нь зөвхөн шагнал урамшуулал, тушаал дэвшүүлэлт, цалин нэмэгдүүлэлт зэрэг хүний нөөцийн шийдвэр гаргахад хэрэглэгдээд зогсохгүй, байгууллагын урт хугацааны стратегийн төлөвлөлтөд чухал үүрэгтэй. Энэхүү судалгаанд уламжлалт болон орчин үеийн гүйцэтгэлийн үнэлгээний арга, хэрэгсэл, давуу болон сул талыг харьцуулан судалж, АНР¹ зэрэг орчин үеийн шийдвэр гаргалтын аргачлалын давуу талыг ашиглан тодотгож өгнө.

1. Гүйцэтгэлийн үнэлгээний ерөнхий зорилго

- Байгууллагын зорилгод хүрэхэд ажилтны хувь нэмрийг үнэлэх
- Шагнал урамшуулал, карьерын өсөлт, цалинтай амралт зэрэг хүний нөөцийн шийдвэрт туслах
- Гүйцэтгэлийг нэмэх зорилготой хөгжлийн төлөвлөгөөг боловсруулах
- Ур чадварын зөрүүг илрүүлэх, түүнийг нөхөхөд хийгдэх сургалт, хөгжлийн төлөвлөгөө боловсруулах

2. Уламжлалт аргачлалууд

- (a) **Эрэмбэлэх арга (Ranking method):** Ажилтнуудыг шууд даргын зүгээс хамгийн сайн нь хэн болохыг харьцуулан эрэмбэлдэг. Гэвч үнэлгээний үндэслэл тодорхой бус, субъектив шинжтэй.

¹ АНР - Analytical Hierarchy Process

- Давуу тал
 - Энгийн бөгөөд ашиглахад хялбар.
 - Хурдан бөгөөд ил тод.
- Сул тал
 - Объектив байдал нь бага
 - Олон ажилтантай байгууллагад тохиромжгүй
 - Ажилтны давуу болон сул талыг тодорхойлоход хүндрэлтэй

(b) **График үнэлгээний хуваарь (Graphic Rating Scales):** Ажилтныг хэд хэдэн чанарын дагуу (жишээлбэл, харилцаа, ажлын гүйцэтгэл) тодорхой үнэлгээний шалгуураар дүгнэдэг.

- Давуу тал
 - Дасан зохицох чадвартай
 - Ашиглахад болон боловсруулахад хялбар
 - Зардал багатай
 - Бүх төрлийн ажлыг үнэлж болно
 - Олон тооны ажилтныг хамарч чадна
- Сул тал
 - Үнэлгээ гаргагчийн хувийн хандлага (субъектив байдал)
 - Бүх шалгуурыг адил жинтэйд тооцдог

(c) **Чухал тохиолдлын арга (Critical Incident Method):** Тухайн ажилтны ажлын явцад гаргасан онцгой, эерэг болон сөрөг зан төлөвийг тэмдэглэж, түүний дагуу үнэлгээ хийдэг.

- Давуу тал
 - Санал хүсэлт өгөхөд хялбар
 - Үнэлгээ нь бодит ажлын зан төлөвт үндэслэдэг

- Доод албан тушаалтнуудын сайжрах боломж өндөр
- Сул тал
 - Мэдээллийг шинжлэх, нэгтгэхэд их цаг зарцуулдаг
 - Судалгаагаар чухал үйл явдлын мэдээлэл цуглуулах нь хүндрэлтэй

(d) **Narrative Essay:** Удирдах албан тушаалтан ажилтны давуу, сул талыг бичгээр тайлбарлаж, хөгжүүлэх чиглэл өгөх зорилгоор хэрэглэнэ.

- Давуу тал
 - Ажилтантай холбоотой мэдээллийн хоосон зайг нөхдөг
 - Бүх хүчин зүйлийг хамардаг
 - Дэлгэрэнгүй, цогц санал хүсэлт өгдөг
- Сул тал
 - Цаг их шаарддаг
 - Үнэлэгчийн хувийн хандлагад амархан автдаг
 - Үр дүнтэй бичиж чаддаг үнэлэгээ гаргагч шаардлагатай

3. Орчин үеийн аргачилалууд

(a) **Зорилгоор удирдах арга (Management by Objectives – MBO):** Ажилтны гүйцэтгэлийг удирдлагын зүгээс тодорхойлсон зорилтуудын хэрэгжилттэй харьцуулан үнэлдэг. Зорилт тогтоох, хэрэгжүүлэх, санал хүсэлт өгөх гурван үндсэн үйл явцтай. Weihrich MBO-г системчилсэн 7 үе шаттайгаар тайлбарласан.

- Давуу тал
 - Хэрэгжүүлэх болон хэмжихэд хялбар
 - Ажилтнуудын үүрэг, хариуцлагыг тодорхой ойлгуулах боломжтой
 - Ажилтанд зөвлөгөө өгөх, чиглүүлэхэд дэмжлэг болдог
- Сул тал

- Зорилгыг өөрөөр ойлгоход ойлголтын зөрүү гарч болзошгүй
- Шударга байдал, чанар зэрэг чухал үнэт зүйлсийг орхигдуулах эрсдэлтэй
- Үнэлүүлж буй ажилтан зорилгод санал нийлэхгүй байх магадлалтай
- Ажлын бүх төрлөд тохиромжтой биш

(b) Зан үйлийн үнэлгээний шкал (Behaviorally Anchored Rating Scales – BARS):

Хувь хүний гүйцэтгэлийг зан төлөвийн жишээн дээр үндэслэн тодорхойлж, тоон үнэлгээтэй уялдуулан дүгнэдэг.

- Давуу тал
 - Ажилтны гүйцэтгэлийг мэргэжлийн үүднээс ажлын зан төлөвөөр тодорхойлдог
 - Үнэлэгч ба үнэлүүлж буй хүн хамтран оролцсоноор үнэлгээг илүү хүлээн зөвшөөрөх магадлалтай
 - Үнэлгээний алдааг багасгахад тусалдаг
- Сул тал
 - Хэмжээст хамааралгүй байдал нь зарим тохиолдолд хүчинтэй эсвэл найдвартай биш байж магадгүй
 - Зан төлөв нь үр дүн гэхээсээ илүү үйл ажиллагаанд чиглэсэн байдаг
 - Цаг их шаарддаг
 - Ажлын төрөл бүр тусдаа BARS (Behaviorally Anchored Rating Scale) хэмжих шаардлагатай

(c) Хүний нөөцийн бүртгэл, тооцоолол (Human Resource Accounting – HRA): Ажилтны

байгууллагад оруулж буй бодит хувь нэмэр болон өртгийг үнэлж, нягтлан бодох бүртгэлийн аргачлалаар илэрхийлдэг.

- Давуу тал
 - Хүний нөөцийг сайжруулах боломж олгодог
 - Хүний нөөцийн бодлогыг боловсруулах, хэрэгжүүлэхэд тусалдаг

- Хүний нөөцөд хийсэн хөрөнгө оруулалтын үр өгөөжийг үнэлдэг
- Ажилтны ур чадвар, чадамжийг дээшлүүлэхэд чиглэгддэг
- Сул тал
 - Хүний нөөцийн зардал ба үнэ цэнийг тодорхойлох тодорхой зааварчилгаа дутмаг
 - Зөвхөн байгууллагын зардлыг хэмждэг бөгөөд ажилтны байгууллагад оруулж буй бодит үнэ цэнийг тооцдоггүй
 - Тодорхойгүй нөхцөл байдалд ажилтны гүйцэтгэлийг бодитоор хэмжих нь бодит бус байдаг

(d) **Үнэлгээний төв (Assessment Center):** Ажилтныг мэргэжлийн ажиглагчдаар ажлын орчны дасгал, симуляци, бүлгийн хэлэлцүүлгээр дамжуулан үнэлдэг төвлөрсөн үнэлгээний хэлбэр.

- Давуу тал
 - Ирээдүйн гүйцэтгэл, ахиц дэвшлийг илүү нарийн таамаглах боломжтой
 - Үндсэн ойлголтууд нь энгийн
 - Уян хатан аргачлалтай
 - Албан тушаал дэвшүүлэх шийдвэр гаргалт болон ажилтны хөгжилд шаардлагатай хэрэгцээг тодорхойлоход дэмжлэг үзүүлдэг
 - Олон төрлийн шинж чанарыг зэрэг үнэлэх боломжтой
- Сул тал
 - Зардал өндөртэй, удирдахад хүндрэлтэй
 - Олон ажилтан, их хэмжээний цаг хугацаа шаарддаг
 - Нэг дор цөөн тооны хүнийг л үнэлэх боломжтой

(e) **360 хэмийн үнэлгээ (360 Degree Feedback):** Дарга, багийн гишүүд, хэрэглэгчид, хамт олон болон өөрийн үнэлгээ зэрэг олон талаас мэдээлэл авч, ажилтныг иж

бүрнээр үнэлнэ.

- Давуу тал
 - Ажилтнууд өдөр тутам харилцдаг хүмүүст үзүүлж буй нөлөөгөө илүү сайн ойлгох боломжтой
 - Ажилтны хөгжлийн маш сайн хэрэгсэл болдог
 - Нарийн, найдвартай систем
- Сул тал
 - Цаг хугацаа их шаарддаг, зардал өндөртэй
 - Өөр өөр бүлгүүдийн дүн шинжилгээ зөрүүтэй гарвал тайлбарлахад хүндрэлтэй
 - Хэлтсүүдийн хоорондын (cross-functional) багуудад хэрэгжүүлэхэд хүндрэлтэй
 - Нууцлалыг хадгалах нь бэрхшээлтэй

Эдгээр нь ажилтны гүйцэтгэлийг үнэлэхэд ашигладаг арга онолын талаасаа юм. Харин энэхүү системийг хөгжүүлэхэд бодит тоон утгаар хэмжигдэхүйц шалгуур хэрэгтэй болсон. Тиймээс үүнд жинлэсэн дундаж оноо (Weighted Average Score) , хэвийн оноо (Normalized Score), Бүдэг логик (Fuzzy logic) зэрэг тоон утгаар хэмжигдэхүйц шалгуур ашигласан. Энэ нь 360 хэмийн үнэлгээ болон зорилгоор удирдах аргачалалыг хэрэглэхэд туслана.

(Weighted Average Score) жинлэсэн дундаж оноо

$$\text{Нийт Оноо} = \sum_{i=1}^n (W_i \cdot S_i) \quad (2.1)$$

Энд:

- S_i : i -р шалгуурын оноо (жишээ нь, 1–10 хүртэл).
- W_i : i -р шалгуурын жин (жишээ нь, даалгаврын гүйцэтгэлд 0.5, ажлын чанарт 0.3, цаг баримтлалд 0.2; нийлбэр: $\sum W_i = 1$).
- n : Шалгуурын тоо.

(Normalized score) Хэвийн оноо Шалгууруудын онооны хуваарь өөр өөр байвал (жишээ нь, 1–5 эсвэл 1–100), оноог 0–1 хүртэл хэвийнжүүлнэ:

$$S_i^{\text{normalized}} = \frac{S_i - S_{\min}}{S_{\max} - S_{\min}} \quad (2.2)$$

Энд:

- S_i : Анхны оноо.
- S_{\min} : Шалгуурын хамгийн бага оноо.
- S_{\max} : Шалгуурын хамгийн их оноо.

Хэвийнжүүлсэн оноог жинтэй дундаж томъёонд ашиглана.

(Fuzzy logic) Бүдэг логик Субъектив үнэлгээг боловсруулахад бүдэг логик ашигладаг бөгөөд оноог гишүүнчлэлийн функцээр (жишээ нь, “Бага”, “Дунд”, “Өндөр”) илэрхийлнэ. Энэ арга нь нарийн төвөгтэй боловч субъектив байдлыг бууруулдаг. Жишээ: Оноог “Бага” (0–0.4), “Дунд” (0.4–0.7), “Өндөр” (0.7–1) гэж ангилна. Гишүүнчлэлийн функцуудыг нэгтгэн нийт оноог тооцоолно.

Ажилтны гүйцэтгэлийн үнэлгээ бол байгууллагын хүний нөөцийн бодлого, хөгжлийн гол хэрэгсэл юм. Үнэлгээ нь шударга, ил тод, системтэй байж чадвал байгууллагын бүтээмжид үнэтэй хувь нэмэр оруулна. Уламжлалт арга нь хялбар боловч субъектив, харин орчин үеийн арга нь илүү иж бүрэн, үнэн зөв, оролцоонд суурилсан байдаг. АНР зэрэг олон шалгуурт шийдвэр гаргалтын арга нь гүйцэтгэлийг илүү нарийвчлалтай, шударга үнэлэх боломжийг нээдэг.

Тиймээс байгууллагууд өөрсдийн онцлогт тохируулан гүйцэтгэлийн үнэлгээний системээ сайтар боловсруулж, тогтмол шинэчилж байх нь зүйтэй.

2.2 Ижил төстэй системүүд

Дипломын ажлын хүрээнд хөгжүүлж буй системтэй ижил төстэй үйл ажиллагаа явуулдаг хоёр системийг нарийвчлан судалж, тэдгээрийг харьцуулна. Судалгаанд Монголын Interactive.mn-ийн бүтээгдэхүүн болон олон улсын Asana-ийн онцлогуудыг авч үзэх бөгөөд эдгээр системүүдийн байгууллагын танилцуулга, ажилтны гүйцэтгэлийг үнэлэх үндсэн модулиудыг тодорхойлно. Дараа нь эдгээр системүүдийг хөгжүүлж буй EPES (Employee Performance Evaluation System) системтэй харьцуулж, давуу тал, онцлог, болон боломжит хязгаарлалтуудыг шинжилнэ. Энэхүү харьцуулалт нь системийн функциональ болон техникийн шинж чанаруудыг тодруулахад чиглэнэ.

2.2.1 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 хянах, мөн өдөр тутмын үйл ажиллагааг хялбаршуулахад чиглэсэн системүүдийг хамардаг. Вэбсайтаас (<https://interactive.mn/product>) харахад тэдний шийдэл нь хэрэглэгчдэд ээлтэй интерфэйс, өөрчлөн тохируулах боломж, болон хялбар нэвтрүүлэлтийн онцлогтой. Компани нь жижиг болон дунд хэмжээний бизнесүүдэд голчлон үйлчилдэг бөгөөд Монголын зах зээлд тохирсон шийдэл санал болгодог.
- **Ажилтны гүйцэтгэлийг үнэлэх үндсэн модуль:** Interactive.mn-ийн систем нь ажилтнуудын гүйцэтгэлийг хянахад чиглэсэн модультай бөгөөд энэ нь ажлын төлөвлөгөө боловсруулах, гүйцэтгэлийн үзүүлэлтүүдийг (KPI) тодорхойлох, мөн тогтмол тайлагнах боломжийг олгодог. Модуль нь менежерүүдэд ажилтны ажлын ахиц, үр дүнг хянах боломж олгодог бөгөөд энгийн тайлангуудыг үүсгэх боломжтой. Гэсэн хэдий ч уг модуль нь голчлон энгийн KPI-д суурилсан үнэлгээнд чиглэдэг бөгөөд нарийвчилсан аналитик, бодит цагийн

шинэчлэлт, эсвэл олон талт интеграцын боломжууд хязгаарлагдмал байж болно. Энэ нь жижиг хэмжээний байгууллагуудад тохиромжтой боловч том байгууллагуудын нарийн шаардлагыг хангахад хангалтгүй байж болох юм.

2.2.2 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 (Software as a Service) платформыг санал болгодог. 2025 оны байдлаар Asana нь дэлхий даяар 131,000 гаруй байгууллага, 2.7 сая гаруй хэрэглэгчтэй бөгөөд томоохон компаниуд (Uber, IBM, NASA гэх мэт)-д өргөн хэрэглэгддэг. Asana-ийн гол онцлог нь хэрэглэгчдэд ээлтэй интерфэйс, өндөр тохируулгатай ажлын урсгал, болон бусад гуравдагч талын хэрэгсэл (Slack, Google Drive гэх мэт)-тэй интеграцлах чадвар юм.
- **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 бөгөөд "Workload" онцлог нь ажилтнуудын ажлын ачааллыг хянах боломжийг олгодог. Гэсэн хэдий ч, Asana-ийн систем нь гүйцэтгэлийн үнэлгээний тусгай модульгүй тул нарийвчилсан KPI-д суурилсан үнэлгээ эсвэл бодит цагийн аналитикт чиглэсэн биш юм.

2.2.3 Харьцуулалт

Эндээс бид Interactive.mn болон Asana-ийн системүүдийг дипломын ажлын хүрээнд хөгжүүлж буй EPES системтэй харьцуулна. EPES нь Golang (Gin, GORM, JWT), Next.js (Tailwind CSS),

PostgreSQL, Docker зэрэг орчин үеийн технологиудыг ашиглан хөгжүүлэгдэж байгаа бөгөөд ажилтны гүйцэтгэлийг бодит цагийн мэдээлэлд суурилан үнэлэхэд чиглэсэн модультай. Энэхүү харьцуулалт нь функциональ болон техникийн онцлогуудыг хамарна.

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, Python	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, HTTPS, OAuth	JWT, HTTPS
Хэрэглээний хүрээ	Жижиг, дунд бизнес	Том байгууллага	Төрөл бүрийн байгууллага
Интеграцын боломж	Хязгаарлагдмал	Slack, Google Drive гэх мэт	REST API-аар дэмжигдэнэ
Скалируемость	Дунд зэрэг	Өндөр	Docker, Kubernetes-ээр өндөр

Хүснэгт 2.1: Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт

Интеграцын Харьцуулалт

Системүүдийн өргөтгөх боломж болон интеграцын чадварыг илүү гүнзгий харьцуулахын тулд доорх хүснэгтийг нэмэв:

Онцлог	Interactive.mn	Asana	EPES
Хэрэглэгчийн хэмжээ	Жижиг, дунд бизнес	Том байгууллага	Төрөл бүрийн хэмжээтэй
Серверын архитектур	Тодорхой бус	Cloud-based, өндөр скалируемость	Microservices, Docker, Kubernetes
Интеграцын API	Хязгаарлагдмал	REST API, Zapier	REST API
Гуравдагч талын хэрэгсэл	Хязгаарлагдмал	Slack, Google Drive, Microsoft Teams	Потенциалтай (нээлттэй эх)
Өгөгдлийн хэмжээний дэмжлэг	Дунд зэрэг	Өндөр	Өндөр (PostgreSQL)

Хүснэгт 2.2: Интеграцын харьцуулалт

2.2.4 Шинжилгээ ба Дүгнэлт

- **Interactive.mn:** Энгийн KPI-д суурилсан үнэлгээний модуль нь жижиг, дунд бизнесүүдэд тохиромжтой боловч бодит цагийн хяналт, нээлттэй эхийн технологи, болон өргөн хүрээтэй интеграцын боломжууд дутагдалтай. Системийн техникийн дэлгэрэнгүй мэдээлэл хязгаарлагдмал тул том хэмжээний өргөтгөлд хязгаарлалттай байж болно.
- **Asana:** Таскийн хяналт, зорилго тогтоох онцлог нь гүйцэтгэлийг шууд бусаар үнэлэх боломжтой боловч тусгай гүйцэтгэлийн үнэлгээний модульгүй. Том байгууллагуудад тохиромжтой, гэхдээ бодит цагийн шинэчлэлт болон нарийвчилсан KPI-д суурилсан аналитик хязгаарлагдмал. Asana-ийн интеграцын боломжууд болон хэрэглэгчийн интерфэйс нь өндөр чанартай.

- **EPES:** Бодит цагийн мэдээлэлд суурилсан үнэлгээний модуль, нээлттэй эхийн технологи, Docker болон Kubernetes-ийн дэмжлэгээрээ онцлог. Энэ нь илүү уян хатан, тохируулгатай боловч хөгжүүлэлтийн эхний шатандаа тул том хэмжээний туршилт, баталгаажуулалт шаардлагатай. REST API болон нээлттэй эхийн шинж чанар нь гуравдагч талын хэрэгсэлтэй интеграцлах боломжийг нэмэгдүүлнэ.

2.2.5 EPES-ийн Боломжит Сайжруулалт

EPES системийн одоогийн хөгжүүлэлтийн байдлыг харгалзан үзэхэд дараах чиглэлээр сайжруулалт хийх боломжтой:

- **Аналитикийн модуль:** Нарийвчилсан KPI-д суурилсан аналитикын алгоритмуудыг нэвтрүүлж, гүйцэтгэлийн үнэлгээ хийх боломжтой болгох.
- **Интеграцын өргөтгөл:** Slack, Microsoft Teams зэрэг алдартай SaaS платформтой шууд интеграцлах боломжийг нэмэх.
- **Хэрэглэгчийн туршлага:** UI/UX-ийг Asana-ийн түвшинд хүргэхийн тулд хэрэглэгчийн санал хүсэлтэд суурилсан туршилт, сайжруулалт хийх.
- **Хамгаалалт:** JWT-ийн хажуугаанд OAuth 2.0 болон нэмэлт шифрлэлтийн протоколуудыг нэвтрүүлж, өгөгдлийн аюулгүй байдлыг сайжруулах.

2.3 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.

Технологи тус бүр ямар онолын үндэслэлээр ажилладаг, яагаад энэ системд тохиромжтой вэ гэдгийг тодорхойлж, практик хэрэглээний жишээнүүдийг судална.

2.3.1 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 өгөгдлийн сангийн схемийн өөрчлөлтийг хялбаршуулна.
- **Сул тал:** Нарийн SQL асуулгад сул талтай тул зарим тохиолдолд SQL бичих шаардлага гарна.
- **Хэрэглээ:** PostgreSQL-тэй хослуулан өгөгдлийн сангийн удирдлагыг хэрэгжүүлнэ.

JWT

- JWT нь криптографийн HMAC алгоритмаар токены агуулгыг баталгаажуулдаг бөгөөд төлөвгүй (stateless) баталгаажуулалтыг хангадаг.
- **Давуу тал:**
 - Сервер дээр session 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.

2.3.2 *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.

2.3.3 *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.

2.3.4 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.

2.3.5 DBever

- Өгөгдлийн сангийн GUI удирдлага бөгөөд SQL стандартыг дэмждэг.
- **Давуу тал:**
 - Өгөгдлийн 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.

2.3.6 PostgreSQL

- Харилцааны өгөгдлийн 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: Өгөгдлийн сангийн үндэс болно.

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 системийн амжилтад хувь нэмэр оруулна. Гэсэн хэдий ч эдгээрийг хэрэгжүүлэхэд сургалт, нөөцийн удирдлага шаардлагатай бөгөөд энэ нь систем төлөвлөлтөд анхаарах ёстой хүчин зүйл болно.

2.4 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, ЗОХИОМЖ

3.1 Системийн хэрэглэгчид

- **Админ:** Системийн тохиргоо, хэрэглэгчийн удирдлага, мэдээллийн сангийн засвар үйлчилгээг хариуцна. Тэдэнд системийн бүрэн хандалтын эрхтэй.
- **Менежер:** Ажилтнуудын гүйцэтгэлийг хянах, тайлан гаргах, даалгавар хуваарилах үүрэгтэй.
- **Хүнийн нөөцийн мэргэжилтэн** Ажилтны бүртгэл хөтлөх, ажилд авах болон чөлөөлөх үйл явцыг удирдах, сургалт зохион байгуулах үүрэгтэй. Тэдэнд хэрэглэгчийн мэдээлэлд хандах, тайлан гаргах боломжтой хязгаарлагдмал хандалтын эрхтэй.
- **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 Системийн шаардлага

Функциональ шаардлага нь системийн гол үйл ажиллагааг тодорхойлж, хэрэглэгчдэд ямар боломж олгохыг заана. Эдгээр шаардлагууд нь системийн үндсэн зорилтыг хангахад чиглэсэн бөгөөд хэрэглэгчийн туршлагыг сайжруулах, гүйцэтгэлийн үнэлгээний процессийг автоматжуулахад тусална. Шаардлагуудыг ерөнхий шаардлага болон хэрэглэгчийн төрлүүдээр (Админ, Менежер (хүний нөөцийн мэргэжилтэн), Ажилтан) ангилан доорх хүснэгтүүдэд дэлгэрэнгүй харуулав.

3.2. СИСТЕМИЙН ШААРДЛАГА

3.2.1 Функциональ шаардлага

Ерөнхий шаардлага

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

ФШ100	Хэрэглэгч бүртгэдэг байх
ФШ101	Хэрэглэгчийн хувийн мэдээлэл харуулдаг байх
ФШ102	Хэрэглэгчийн оролцсон төсөл, даалгавар харуулдаг байх
ФШ103	Даалгавар үүсгэх
ФШ104	Системийн лог хөтлөдөг байх
ФШ105	Хэрэглэгчийн session удирддаг байх

Хүснэгт 3.1: Ерөнхий шаардлага

Админ шаардлага

Системийн удирдлага, аюулгүй байдал, засвар үйлчилгээтэй холбоотой бөгөөд системийн тогтвортой байдлыг хангахад чиглэнэ.

АФШ200	Хэрэглэгчийн эрхийг удирддаг байх
АФШ201	Системийн тохиргоог өөрчилдөг байх
АФШ202	Мэдээллийн сангийн нөөцлөлт, сэргээлт хийдэг байх
АФШ203	Бүх хэрэглэгчийн үйлдлийн түүхийг хянах

Хүснэгт 3.2: Админ шаардлага

3.2. СИСТЕМИЙН ШААРДЛАГА

Менежер шаардлага

Ажилтны гүйцэтгэлийг удирдах, хянах, тайлагнахад чиглэсэн бөгөөд системийн гол зорилгыг хэрэгжүүлэхэд тусална.

МФШ300	Даалгавар үүсгэж, хуваарилдаг байх
МФШ301	Даалгаврын гүйцэтгэлийн явцыг хянадаг байх
МФШ302	Ажилтны гүйцэтгэлийг KPI-д суурилан үнэлгээг автоматаар гаргадаг байх
МФШ303	Тайлан гаргадаг байх
МФШ304	Тайланг PDF эсвэл CSV татаж авах боломжтой байх

Хүснэгт 3.3: Менежер шаардлага

Ажилтан шаардлага

Хувь хүний гүйцэтгэлийг хянах, даалгавар удирдахад чиглэсэн бөгөөд хэрэглэгчийн идэвхийг дэмжинэ.

АФШ400	Даалгавар үүсгэдэг байх
АФШ401	Даалгаврын үйл явцыг удирдах
АФШ402	Өөртөө үнэлгээ өгөх
АФШ403	Өөрийн үнэлгээг хянах

Хүснэгт 3.4: Ажилтан шаардлага

3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

3.2.2 Функциональ биш шаардлага

ФБШ100	Систем нь 24/7 ажиллах чадвартай байх
ФБШ101	Систем нь веб сайтын стандартыг дагаж мөрдөх
ФБШ102	Систем нь аюулгүй байдал сайтай байх
ФБШ103	Хэрэглэгчийн нэвтрэх мэдээллийг хамгаалдаг байх
ФБШ104	Хэрэглэгчийн нууц үгийг шифрлэдэг байх
ФБШ105	Системийн хариулах хугацаа 3 секундээс бага байх
ФБШ106	Хэрэглэгчийн хандалтад хязгаарлалт тавих

Хүснэгт 3.5: Функциональ биш шаардлага

3.2.3 Системийн шаардлага

СШ500	Систем нь өгөгдлийн санг үр дүнтэй удирдаж, мэдээллийг хадгалах, устгах, шинэчлэх функцуудтай байх
СШ501	Систем нь хэрэглэгчийн эрхүүдийг хянаж, админ, ажилтан зэрэг түвшингийн хэрэглэгчдэд тохирсон эрх олгох
СШ502	Эрхээс шалтгаалж харагдац өөр байх
СШ503	Хэрэглэгчийн нэвтрэх мэдээллийг хамгаалдаг байх
СШ504	Хэрэглэгчийн нууц үгийг шифрлэдэг байх

Хүснэгт 3.6: Функциональ биш шаардлага

3.2.4 UI/UX шаардлага

Уг системийн гол үйл ажилгаа гүйцэтгэлийн үнэлгээг ихэвчлэн суурин компьютер дээр гаргах учир админ болон менежерийн харагдац үүнд тохирсон байх. Харин ажилтны хувьд

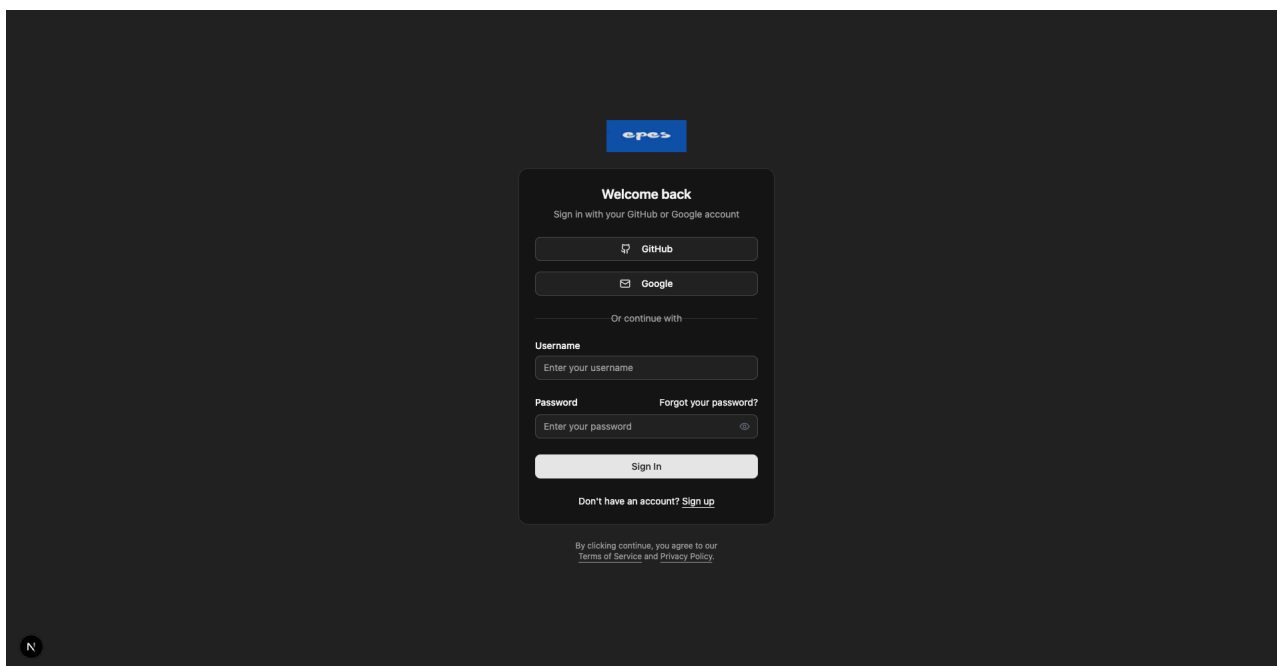
3.2. СИСТЕМИЙН ШААРДЛАГА

БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

гар утаснаас хандах боломжыг нэмж өгөх хэрэгтэй. Суурь веб-д апп нь динамик хариу үйлдэл үзүүлдэг тул үүн дээр асуудал гарахгүй.

Нэвтрэх хуудас

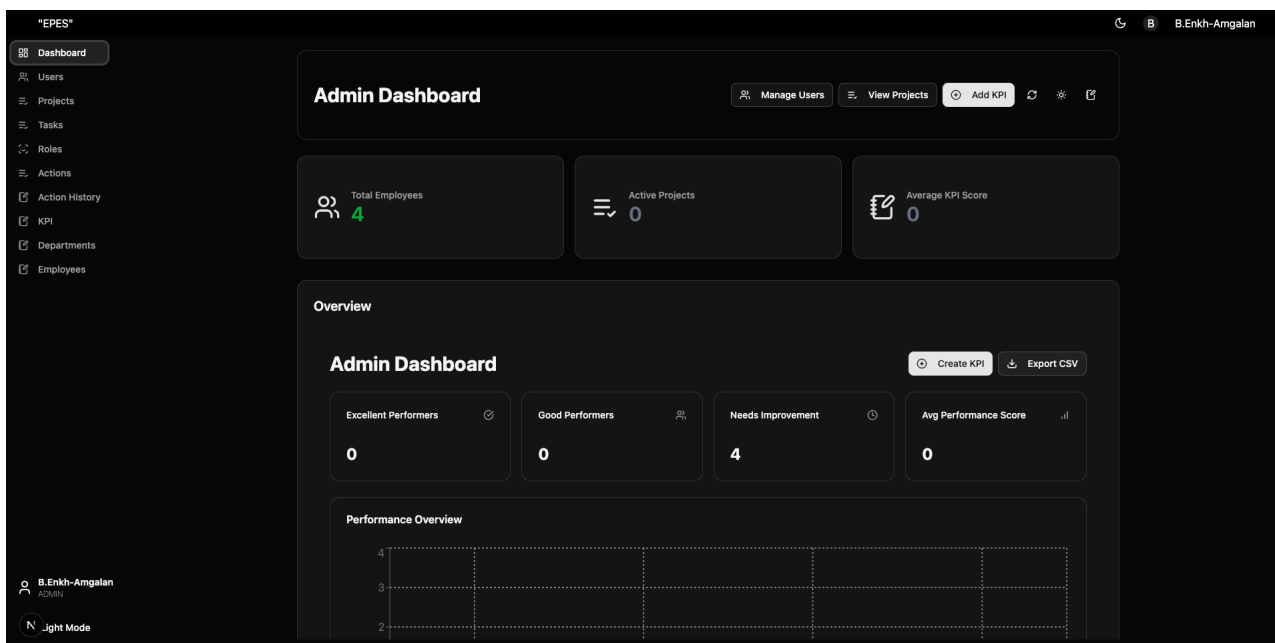
Системийн нэвтрэх хэсэг хэрэглэгчийн хандах эрхээс үл шалтгаалж нэгэн адил харагдацтай байна.



Зураг 3.1: Нэвтрэх хуудас

3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

Системийн ерөнхий хуудаснууд



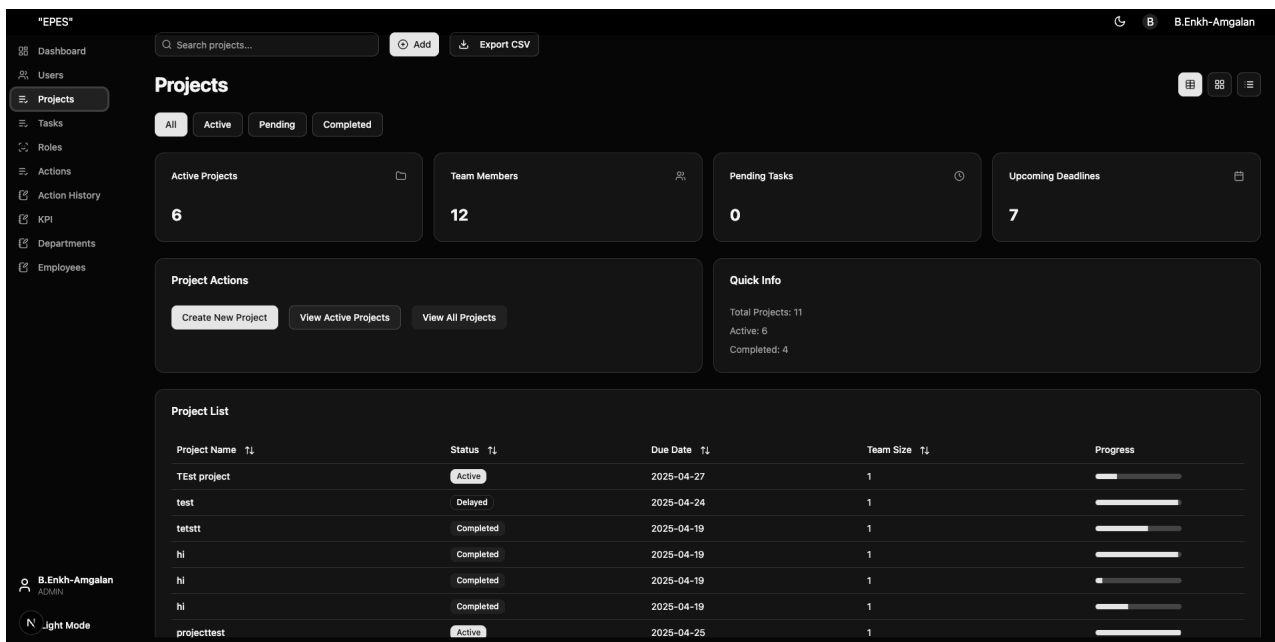
Зураг 3.2: Админ дашбоард харагдац

The screenshot shows the Users management page in the EPES system. The page title is 'Ажилчид' (Employees) with a subtitle 'Энэ хэсэгт ажилчдийг удирдана.' (Manage employees in this section). The table below lists the users:

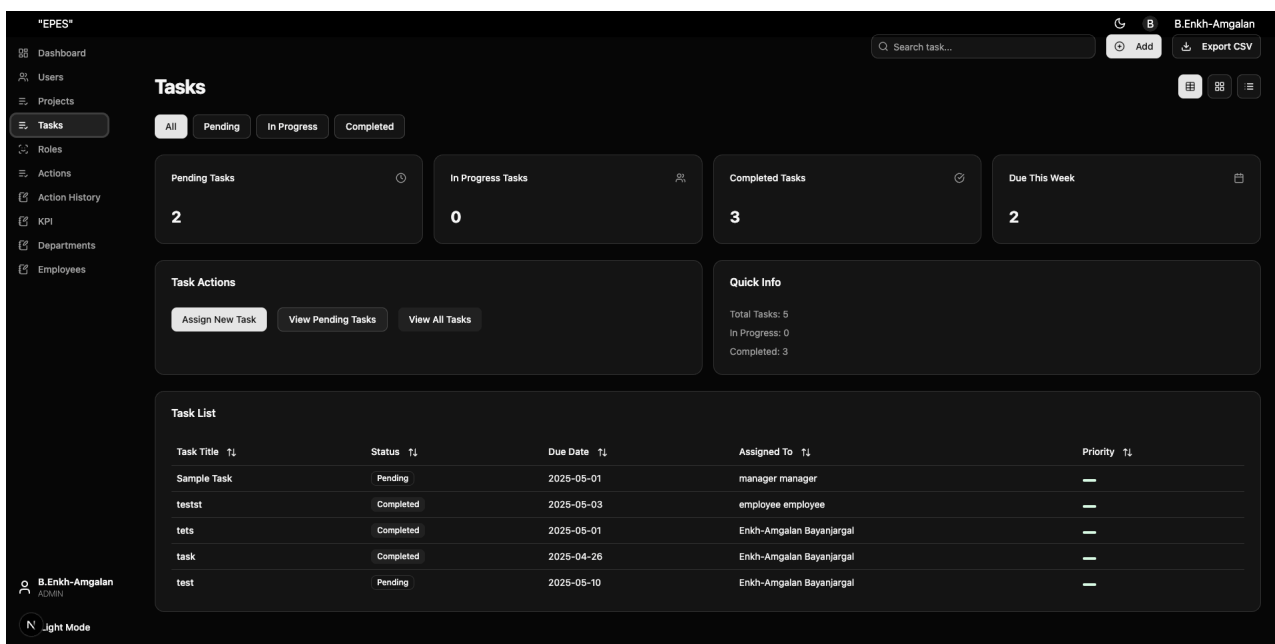
Имэйл	Нэр	Нөөгрэх нэр	Дугаар	Төлөв	Бүртгүүлсэн хугацаа	Шинэчлэсэн хугацаа
enkhmgalan.b@ondo.mn	B.Enkh-Amgalan	test	60096966	Идэвхтэй	3/19/2025	3/26/2025
amgalandw0822@gmail.com	e.employee	employee	95560899	Идэвхтэй	3/25/2025	4/15/2025
amgalandw0822@gmail.com	m.manager	manager	95560899	Идэвхтэй	3/25/2025	4/15/2025
afmamf@gmail.com	t.tets	teseeeet		Идэвхгүй	4/24/2025	4/24/2025
amgalandw0822@gmail.com	B.Enkh-Amgalan	amgalan	95560899	Идэвхтэй	4/23/2025	4/24/2025

Зураг 3.3: Ажилтны жагсаалт

3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



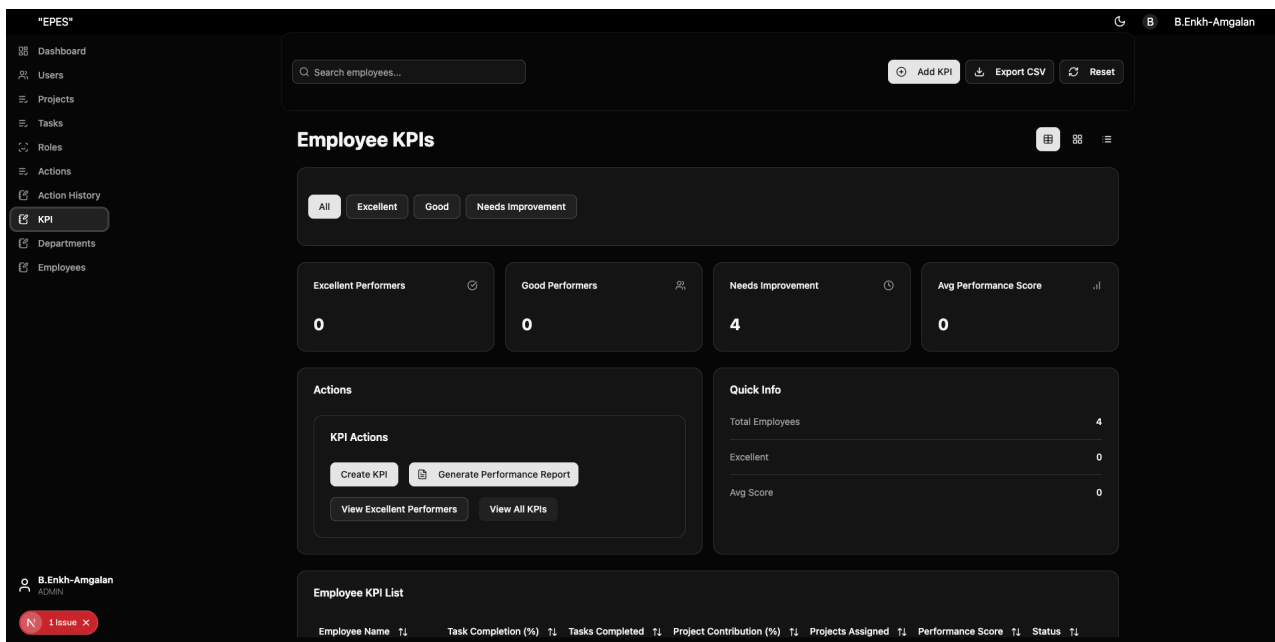
Зураг 3.4: Төслийн харагдац



Зураг 3.5: Даалгаварын харагдац

3.2. СИСТЕМИЙН ШААРДЛАГА

3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.6: Гүйцэтгэлийн үнэлгээний харагдац

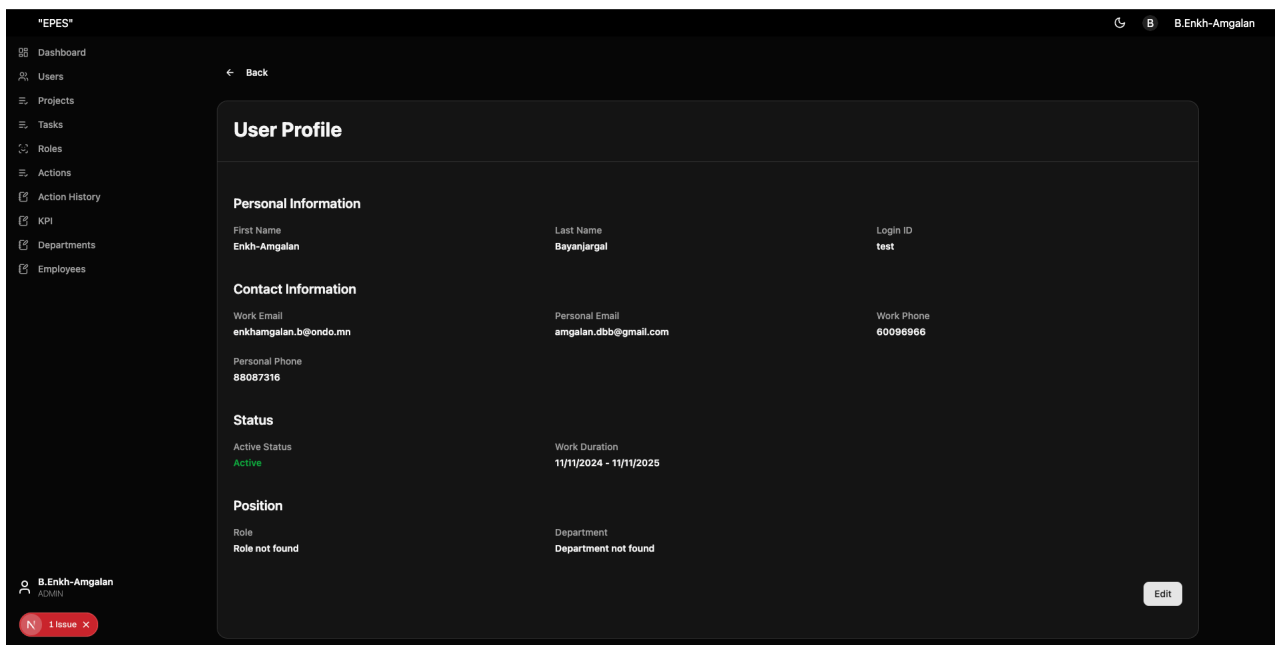
The dialog box is titled "Generate Performance Report" and contains the following fields and options:

- Employee: A dropdown menu with "All Employees" selected.
- Time Period: A dropdown menu with "All Time" selected.
- Include Task Details: A checked checkbox.
- Include Project Details: A checked checkbox.
- Employee Performance Comments: An unchecked checkbox.
- Buttons: "Cancel" and "Generate PDF Report".

Зураг 3.7: Тайлан гаргах компонентийн харагдац

3.2. СИСТЕМИЙН ШААРДЛАГА

БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

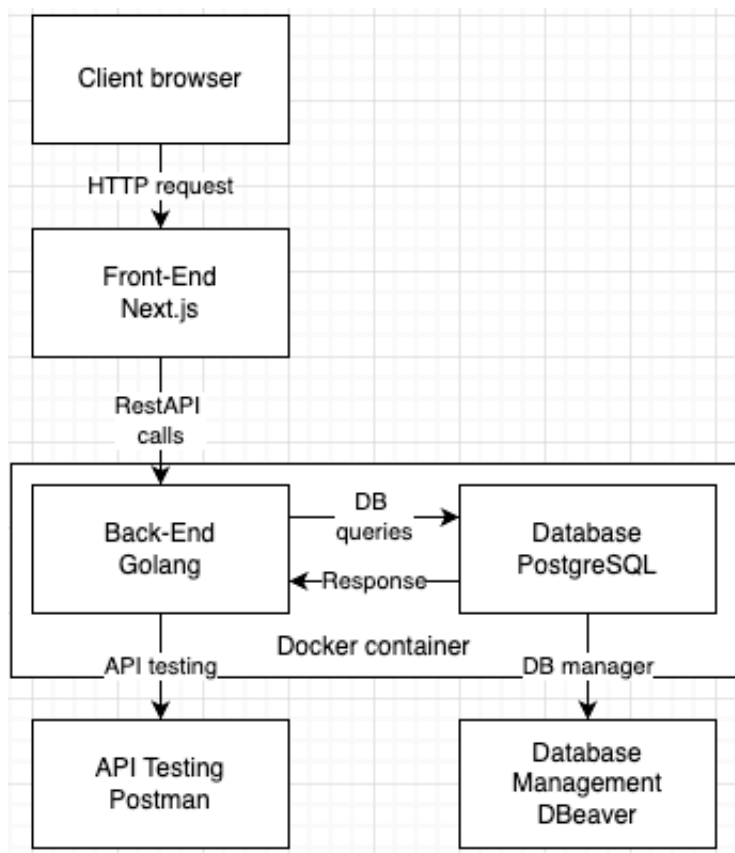


Зураг 3.8: Хэрэглэгчийн профайлын харагдац

3.3 Системийн загвар

3.3.1 Системийн архитектур

Системийн үндсэн архитектур нь 3 шатлалт аргыг ашиглан хөгжүүлсэн. Хэрэглэгч талаас веб хөтөч ашиглан HTTP хүсэлт илгээнэ. Сервер талаас өгөгдлийн санг ашиглан API бэлдэн хэрэглэгчрүү HTTP хүсэлтийн хариу өгөх зарчимаар ажиллана.



Зураг 3.9: Системийн архитектурын диаграмм

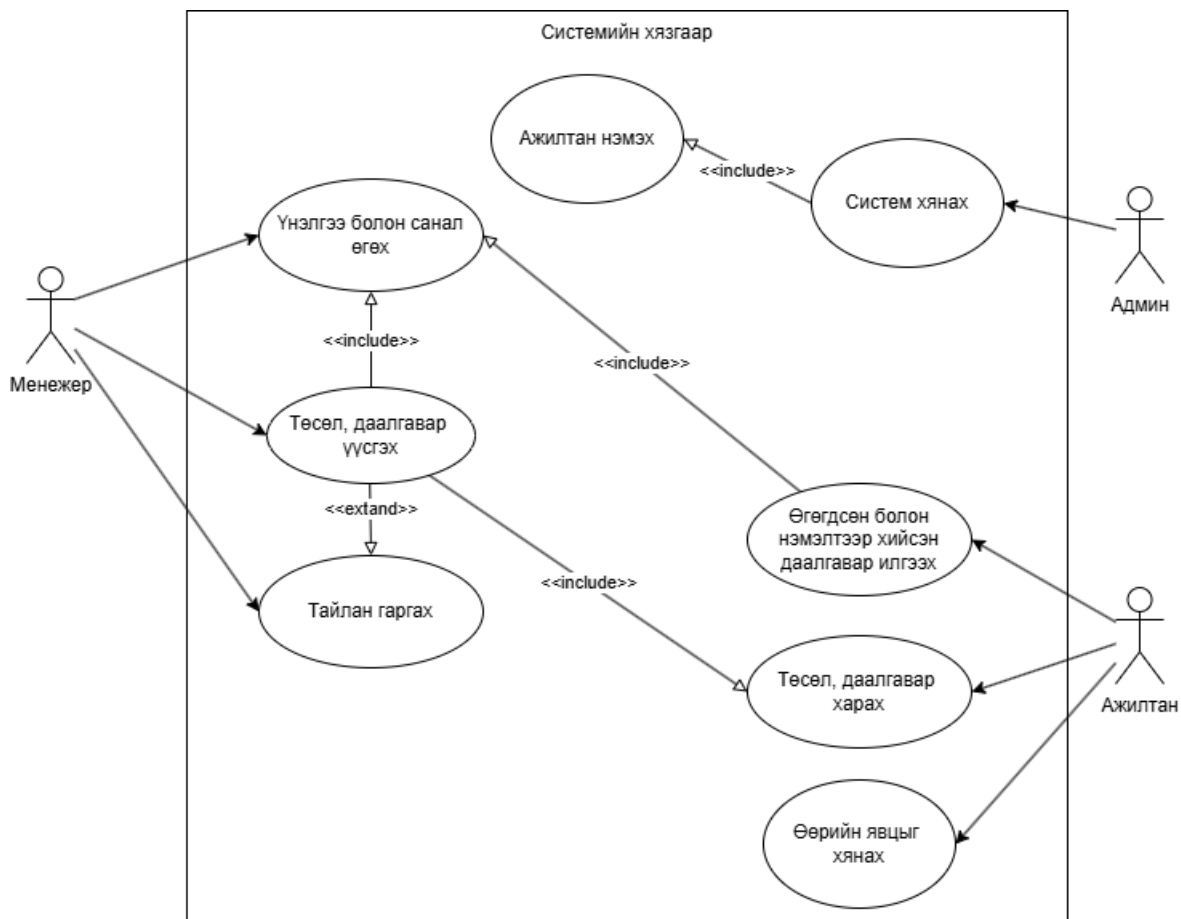
3.3.2 Системийн зохиомж

Системийн ажлын явцын диаграмм

Энэхүү диаграммд админ, менежер, ажилтан гэсэн гурван төрлийн оролцогч (тоглогч) бий. Диаграмм нь оролцогчдын хийж болох үйлдлүүдийг болон тэдгээрийн хоорондын харилцаа,

3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

холбоог харуулж байна. Үйлчлүүлэгч, админ, ажилтан тус бүр нь өөрийн хариуцсан үйлдлүүдийг гүйцэтгэх бөгөөд тэдгээрийн харилцан үйлчлэл, эрхийн түвшин, үүрэг хариуцлагын хил хязгаарыг диаграммаас ойлгомжтой байдлаар харах боломжтой.



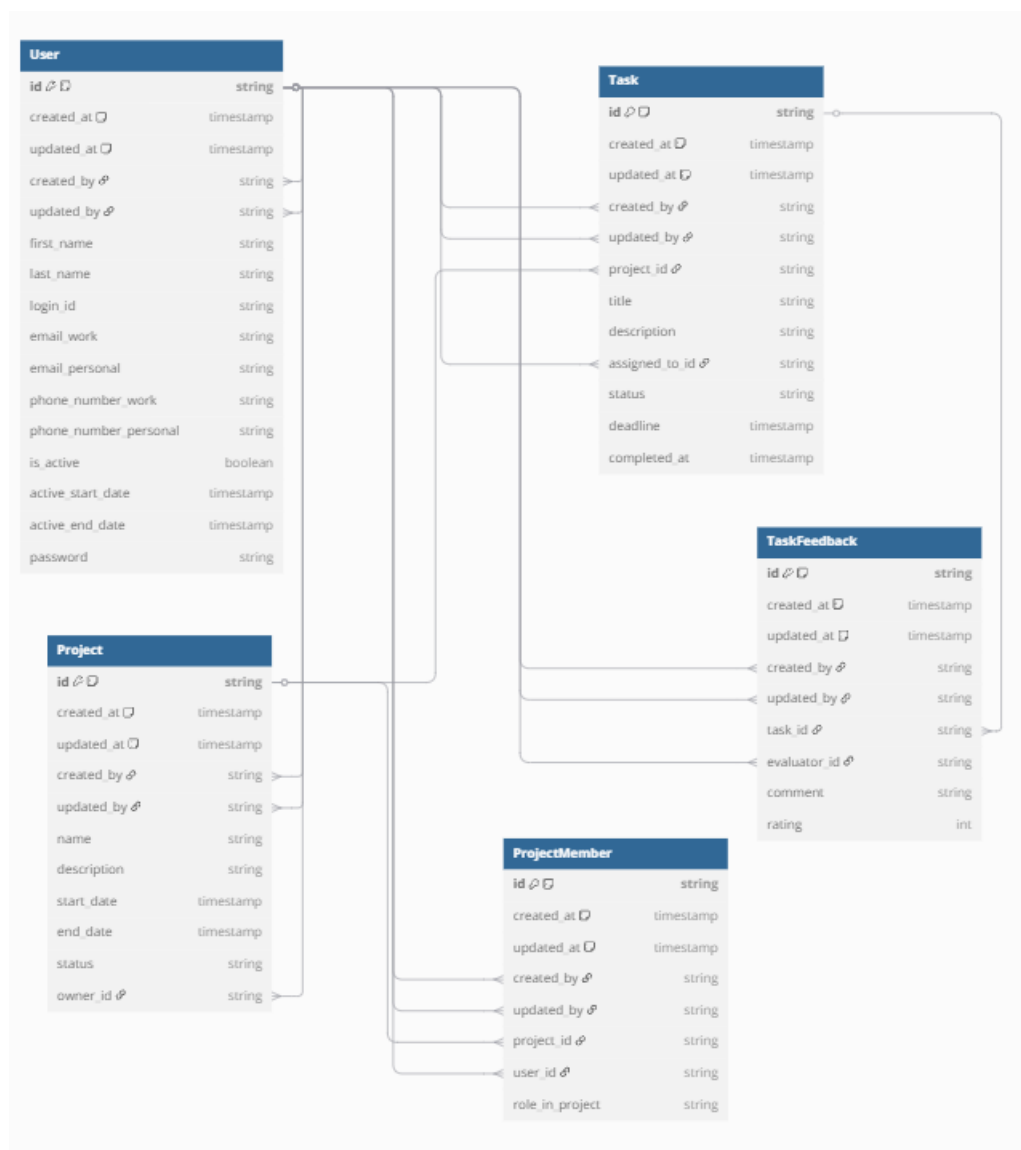
Зураг 3.10: Системийн ажлын явцын диаграмм

Системийн нэгж хоорондын харилцаа хамаарлын диаграмм

Уг диаграмм нь системийн өгөгдөл бүрдүүлэх бүх гол мэдээллүүдийн ерөнхий бүтцийг харуулж байна. Системийг цаашид хөгжүүлэх үед уян хатан, өргөтгөх боломжтой байдлаар диаграммыг боловсруулсан. Объектуудын хоорондын харилцаа, тэдгээрийн холбоосууд нь системийг нэмэлт функц, боломжуудаар өргөжүүлэхэд хялбар, логик уялдаатай байлгах зорилготой. Энэ нь системийг илүү үр ашигтай удирдах, өгөгдлийг найдвартай зохицуулах үндсэн суурь

3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

БОЛЖ ӨГНӨ.

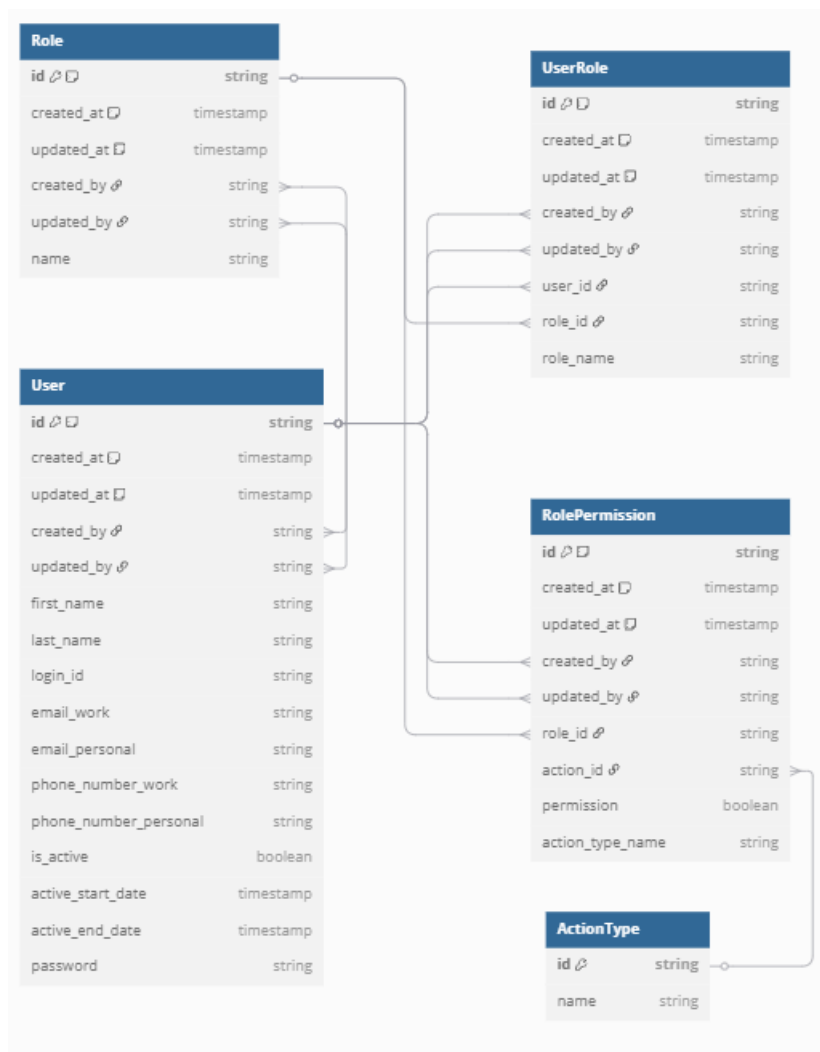


Зураг 3.11: Төсөл болон даалгаврын харилцан хамаарлын диаграмм

3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.12: Ажилтны нэгж хоорондын харилцан хамаарлын диаграмм

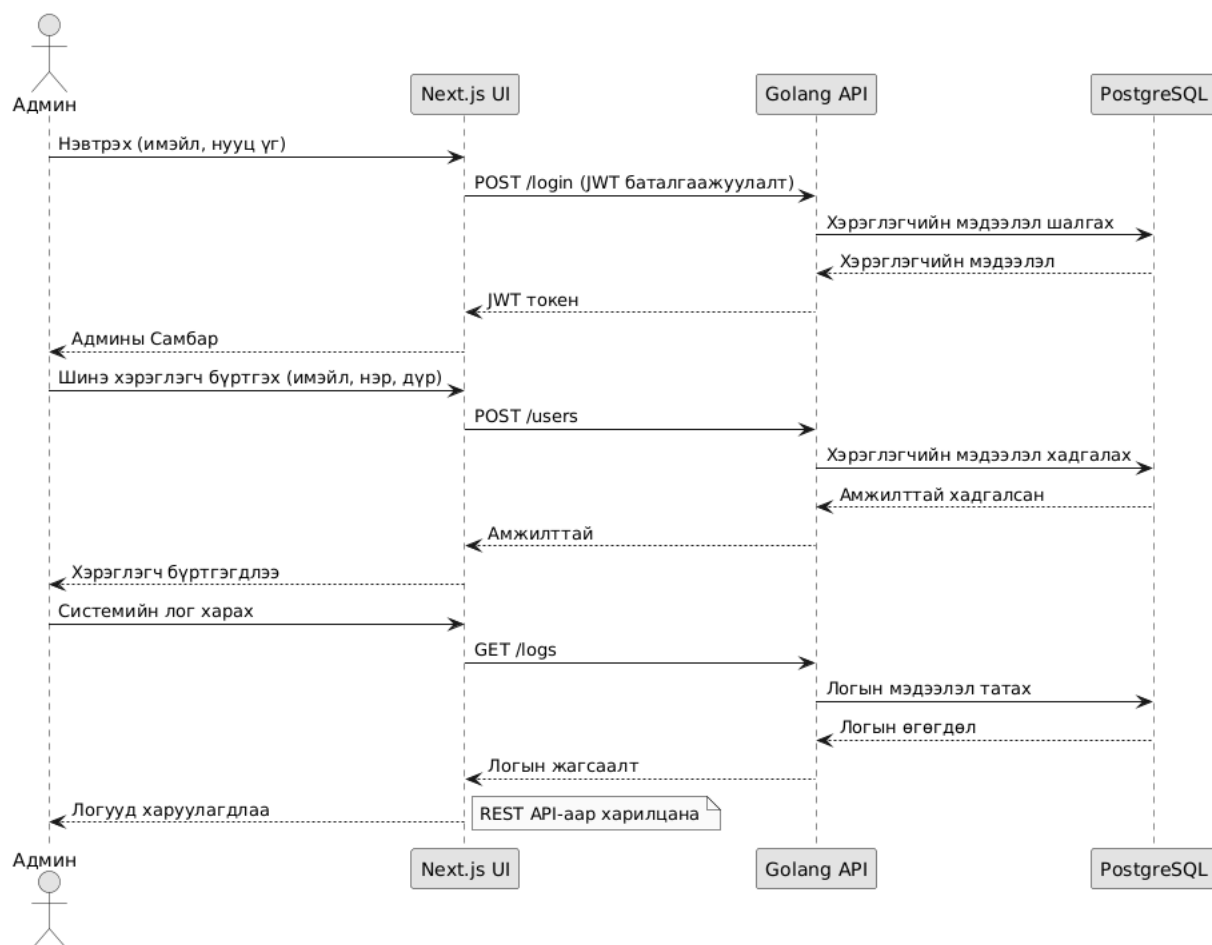


Зураг 3.13: Ажилтан болон түүний эрхийн харилцан хамаарлын диаграмм

Дарааллын диаграмм

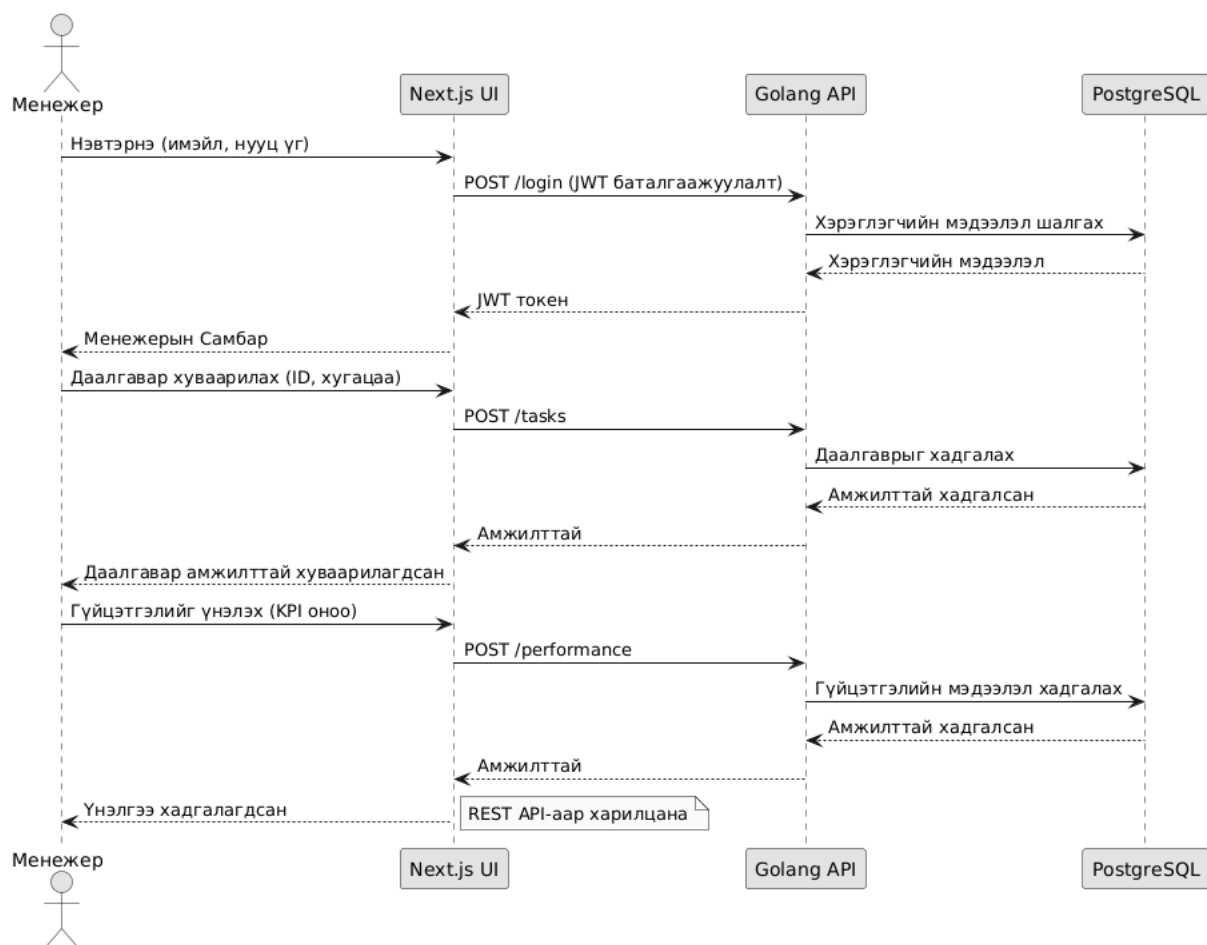
Дарааллын диаграмм нь системийн бүрэлдэхүүн хэсгүүдийн хоорондын харилцан үйлчлэлийг цаг хугацааны дарааллаар харуулдаг UML-ийн нэг төрлийн диаграмм юм. Энэ нь тодорхой үйлдэл, жишээ нь даалгаврын илгээлт гэх мэт процессын явцад объектуудын хоорондох мессежийн урсгалыг тодорхойлдог.

3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

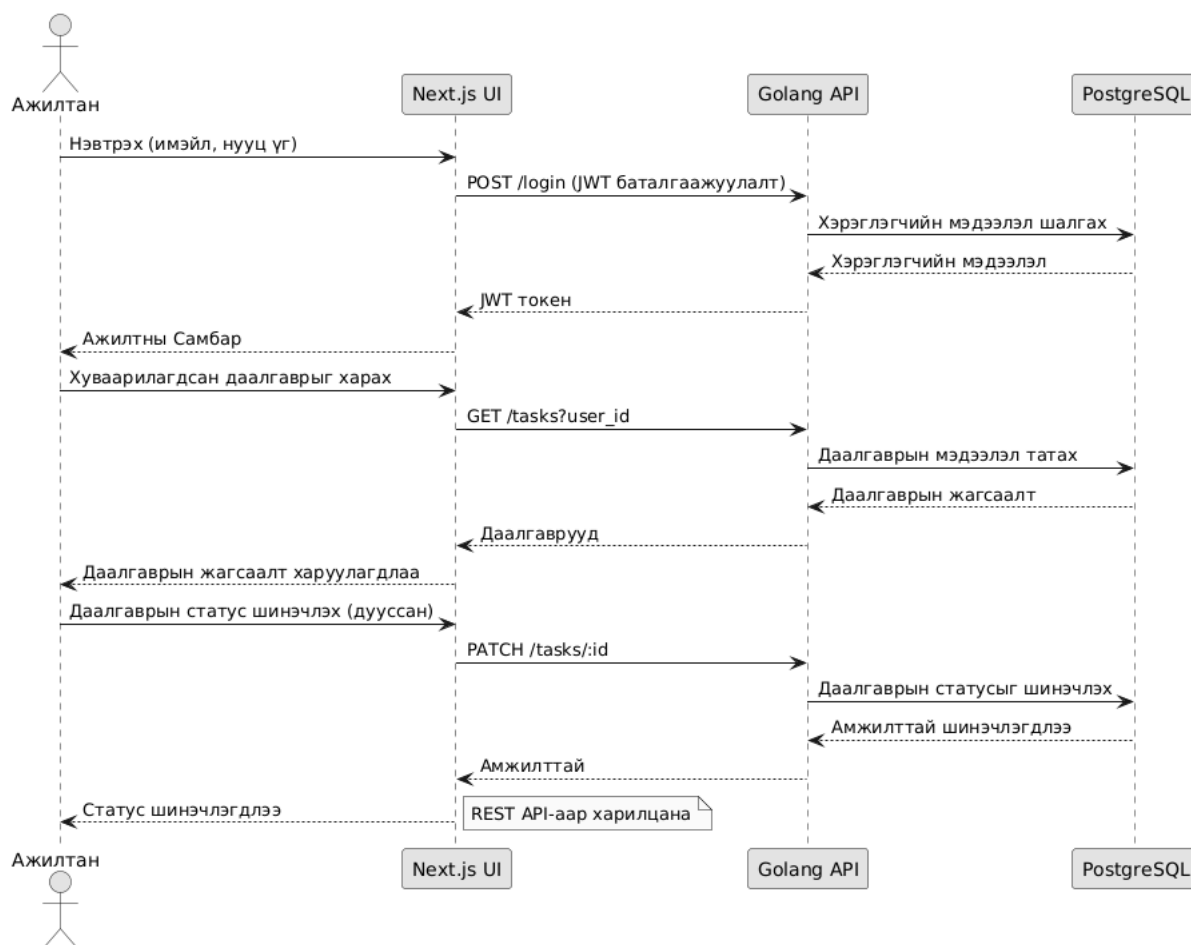


Зураг 3.14: Админ дарааллын диаграмм

3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.15: Менежер дарааллын диаграмм



Зураг 3.16: Ажилтан дарааллын диаграмм

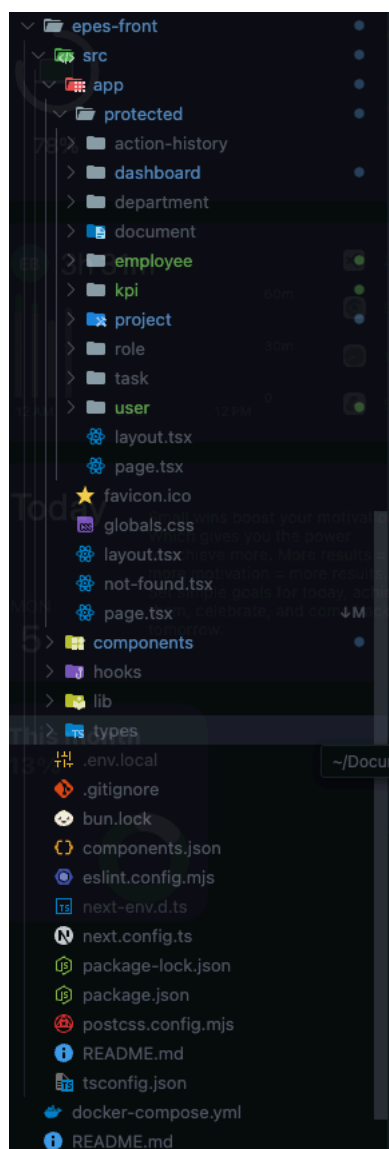
3.4 Бүлгийн дүгнэлт

Энэ бүлэгт системийн хэрэглэгч тодорхойлон түүн дээр тулгуурлан шаардлагуудаа гаргасан. Функциональ болон функциональ бус шаардлагуудыг мөн системийн шаардлагыг гаргасан. Мөн функциональ шаардлагад нийцсэн UI/UX шаардлага тодорхойлж тэдгээрийн загварыг гаргасан. Харин системийн загвар талаас системийн архитектур, системийн ажлын явцын диаграмм, системийн нэгж хоорондын харилцаа хамаарлын диаграмм, дарааллын диаграммуудыг гаргасан. Эдгээр шаардлагуудыг гаргасанаар хөгжүүлэлтийн явцыг хурдасгах, төөрөгдөлийг арилгах зорилготой. Цаашид эдгээр шаардлагуудыг системийн хөгжүүлэлтэд ашиглах болно.

4. ХЭРЭГЖҮҮЛЭЛТ

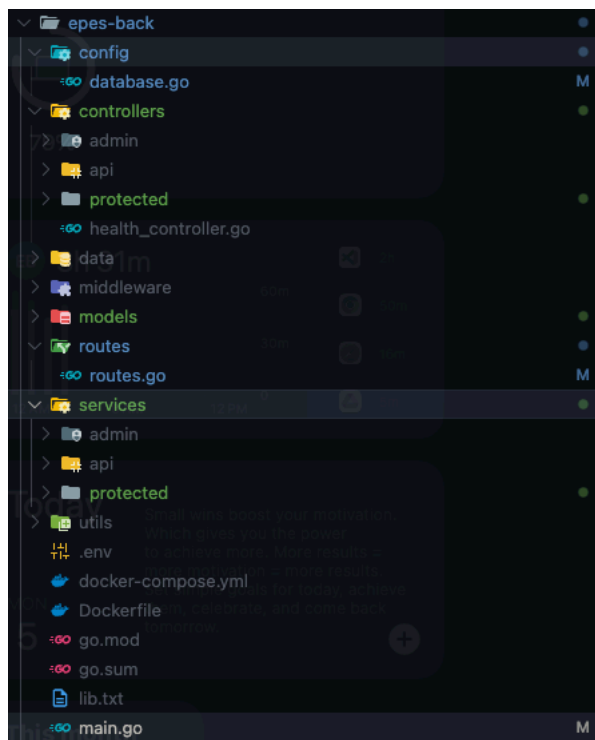
4.1 Файлын бүтэц

4.1.1 *Front-end*



Зураг 4.1: Front-end файлын бүтэц

4.1.2 Back-end

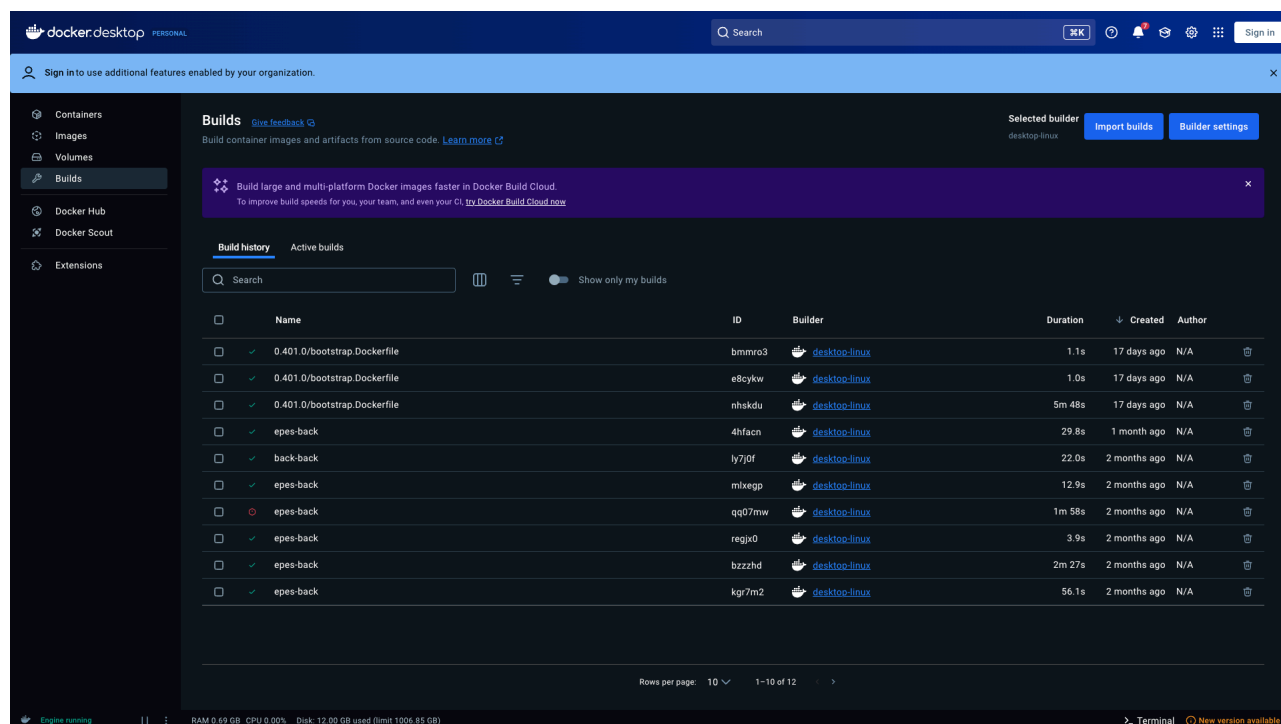


Зураг 4.2: Back-end файлын бүтэц

4.2 Хөгжүүлэлтийн орчин бүрдүүлэлт

Өмнө системийн шинжилгээ хэсэгт тодорхойлсон системийн шаардлагуудыг хэрэгжүүлэх үүднээс өмнө судалсан технологиудын дагуу хөгжүүлэлтийн орчинг бэлдсэн.

Docker Desktop суулган дараах container-үүдийг үүсгэсэн.



Зураг 4.3: Docker Desktop програмын интерфэйс

4.2.1 Docker container ашиглан өгөгдлийн сан үүсгэж, түүний серверийг ажиллуулахад ашигласан.

```

1 FROM golang:1.24
2
3 WORKDIR /app
4
5 COPY go.mod go.sum ./
6 RUN go mod download

```



```
7
8 COPY . .
9
10 RUN go build -o main .
11
12 EXPOSE 8080
13
14 CMD ["/main"]
```

Код 4.1: Dockerfile

```
1 version: "3.8"
2 services:
3   app:
4     build: .
5     ports:
6       - "8080:8080"
7     depends_on:
8       - db
9     environment:
10       - DB_HOST=db
11       - DB_USER=root
12       - DB_PASSWORD=rootpass
13       - DB_NAME=epes_db
14       - DB_PORT=5432
15
16   db:
17     image: postgres:latest
18     environment:
```

```

19     - POSTGRES_USER=root
20     - POSTGRES_PASSWORD=rootpass
21     - POSTGRES_DB=epes_db
22     ports:
23     - "5432:5432"
24     volumes:
25     - postgres_data:/var/lib/postgresql/data
26
27 volumes:
28     postgres_data:

```

Код 4.2: docker-compose.yaml

4.2.2 Github орчин бүрдүүлэлт

Github-д репо үүсгэж түүндээ системийн кодыг байршуулсан. <https://github.com/amgaland/epes> Ингэснээр системид version control хийх боломжтой болсон.

4.2.3 Front-end орчин бүрдүүлэлт

Front-end хэсэгт Nextjs болон Shadcn UI болон Bun ашигласан.

```
1 curl -fsSL https://bun.sh/install | bash
```

Код 4.3: Bun суулгах

```
1 bun install next@latest
```

Код 4.4: Nextjs суулгах

```
1 bunx --bun shadcn@latest init
```

Код 4.5: Shadcn суулгах

4.2.4 *Back-end орчин бүрдүүлэлт*

Back-end хэсэгт go lang болон gin framework ашигласан. Golang-ийг <https://golang.org/dl/> хаягаас татаж суулгаж болно.

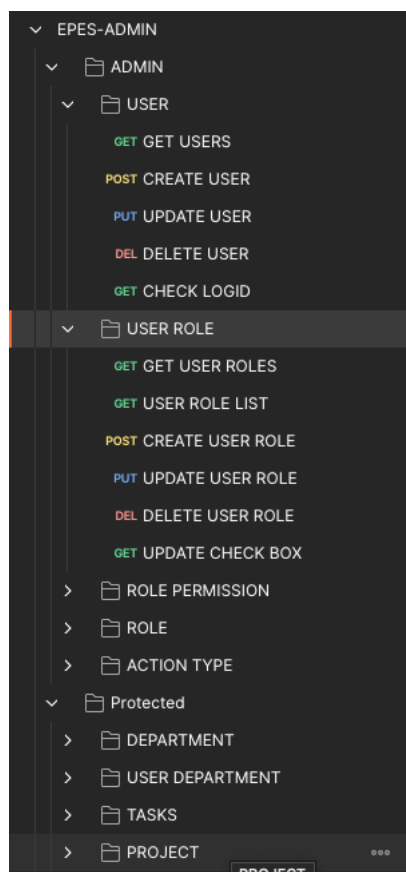
```
1 brew install go
2 brew install gin
```

Код 4.6: Golang суулгах

Хэрэгцээт технологиудаа суулгасны дараа шаардлагад тодорхойлсон архитектурын дагуу код бичнэ.

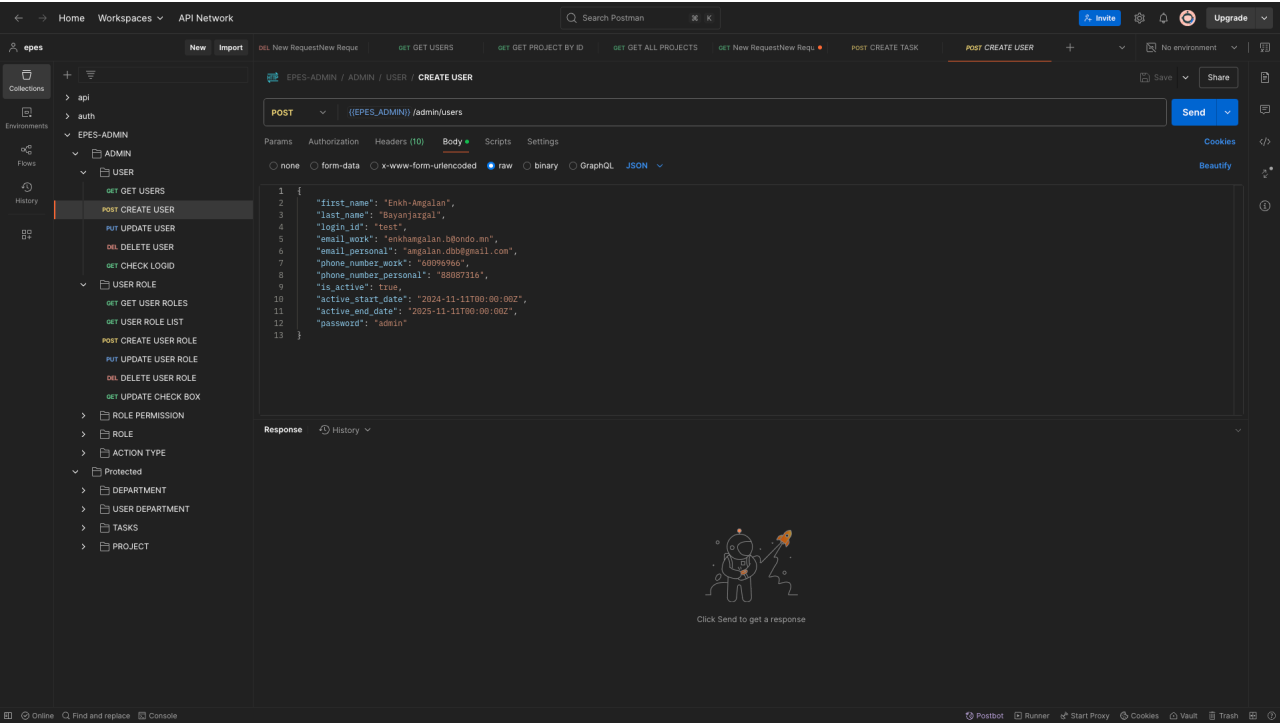
4.2.5 *Test орчин бүрдүүлэлт*

Back-end хэсгийн тестийг Postman ашиглан хийсэн. Postman-д системийн API-уудыг тестлэх орчин бүрдүүлсэн.



Зураг 4.4: Postman файлын бүтэц

Хэрэглэгч үүсгэх тестийн жишээ



Зураг 4.5: Postman хэрэглэгч үүсгэх тест

4.3 Шаардлагын хэрэгжүүлэлт

Өмнө системийн шинжилгээ хэсэгт тодорхойлсон системийн шаардлагуудаас бүх шаардлагыг хэрэгжүүлсэн. Хэрэгжүүлэлтийг харуулахдаа эхлээд шаардлагын дугаар түүний доор хэрэгжүүлэлтийн back-end болон font-end хэсгийг харуулав.

МФШ302	Ажилтны гүйцэтгэлийг KPI-д суурилан үнэлгээг автоматаар гаргадаг байх
--------	---

```
1 "use_client";
2
3 import { useState, useEffect, useMemo, useCallback } from "react";
```

```
4   import { useSession } from "next-auth/react";
5   import { useRouter } from "next/navigation";
6   import { useToast } from "@/hooks/use-toast";
7   import { Button } from "@/components/ui/button";
8   import { Input } from "@/components/ui/input";
9   import { Skeleton } from "@/components/ui/skeleton";
10  import { Card, CardContent, CardHeader, CardTitle } from "@/
      components/ui/card";
11  import { Separator } from "@/components/ui/separator";
12  import {
13    BarChart,
14    CheckCircle,
15    CirclePlus,
16    Clock,
17    Download,
18    LayoutGrid,
19    List,
20    Table as TableIcon,
21    Users,
22    RefreshCw,
23  } from "lucide-react";
24  import { Search } from "lucide-react";
25  import { KPIStats } from "../components/KPIStats";
26  import { KPIActions } from "../components/KPIActions";
27  import { KPITable } from "../components/KPITable";
28  import { KPIGrid } from "../components/KPIGrid";
29  import { KPIList } from "../components/KPIList";
30  import { KPIReportDialog } from "../components/KPIReportDialog";
```

```
31 import { DeleteDialog } from "../components/DeleteDialog";
32 import { fetchEmployeeKPIs, deleteKPI } from "../services/kpiService";
33
34 import {
35     sortKPIs,
36     filterKPIs,
37     exportToCSV,
38     generatePerformanceReport,
39 } from "../utils/kpiUtils";
40
41 import { EmployeeKPI, KPIStat, ReportConfig } from "../types";
42
43 // Custom debounce function to avoid lodash dependency
44 const debounce = <F extends (...args: any[]) => void>(
45     func: F,
46     wait: number
47 ) => {
48     let timeout: NodeJS.Timeout;
49     return (...args: Parameters<F>) => {
50         clearTimeout(timeout);
51         timeout = setTimeout(() => func(...args), wait);
52     };
53 };
54
55 const KPISPage: React.FC = () => {
56     const { data: session, status } = useSession();
57     const router = useRouter();
58     const { toast } = useToast();
59     const [isLoading, setIsLoading] = useState(true);
```

```

58     const [viewMode, setViewMode] = useState<"table" | "grid" | "list
        ">("table");
59     const [searchTerm, setSearchTerm] = useState("");
60     const [searchInput, setSearchInput] = useState("");
61     const [kpis, setKPIs] = useState<EmployeeKPI[]>([]);
62     const [stats, setStats] = useState<KPIStat[]>([]);
63     const [sortField, setSortField] = useState<keyof EmployeeKPI |
        null>(null);
64     const [sortDirection, setSortDirection] = useState<"asc" | "desc"
        >("asc");
65     const [filterStatus, setFilterStatus] = useState<
66         "All" | "Excellent" | "Good" | "Needs_Improvement"
67     >("All");
68     const [deleteDialogOpen, setDeleteDialogOpen] = useState(false);
69     const [employeeToDelete, setEmployeeToDelete] = useState<string |
        null>(null);
70     const [reportDialogOpen, setReportDialogOpen] = useState(false);
71     const [reportConfig, setReportConfig] = useState<ReportConfig>({
72         employeeId: "all",
73         period: "allTime",
74         includeTasks: true,
75         includeProjects: true,
76         includeComments: false,
77     });
78
79     // Centralized KPI metrics calculation
80     const calculateKPIMetrics = useCallback((kpiData: EmployeeKPI[])
        => {

```



```

81     const excellentPerformers = kpiData.filter(
82       (k) => k.status === "Excellent"
83     ).length;
84     const goodPerformers = kpiData.filter((k) => k.status === "Good
85       ").length;
86     const needsImprovement = kpiData.filter(
87       (k) => k.status === "Needs_Improvement"
88     ).length;
89     const avgPerformanceScore =
90       kpiData.length > 0
91         ? Math.round(
92             kpiData.reduce((sum, k) => sum + k.performanceScore, 0)
93             /
94             kpiData.length
95           )
96         : 0;
97     const totalEmployees = kpiData.length;
98
99     return {
100       stats: [
101         {
102           title: "Excellent_Performers",
103           value: excellentPerformers,
104           icon: CheckCircle,
105         },
106         { title: "Good_Performers", value: goodPerformers, icon:
107           Users },
108       ],
109     };
110   }
111 }

```

```

105         { title: "NeedsImprovement", value: needsImprovement, icon
           : Clock },
106     {
107         title: "AvgPerformanceScore",
108         value: avgPerformanceScore,
109         icon: BarChart,
110     },
111 ],
112 metrics: {
113     totalEmployees,
114     excellentPerformers,
115     avgPerformanceScore,
116 },
117 };
118 }, []);
119
120 // Debounced search handler
121 const debouncedSearch = useMemo(
122     () =>
123     debounce((value: string) => {
124         setSearchTerm(value);
125     }, 300),
126     []
127 );
128
129 // Handle search input change
130 const handleSearchChange = useCallback(
131     (e: React.ChangeEvent<HTMLInputElement>) => {

```

```
132     const value = e.target.value;
133     setSearchInput(value);
134     debouncedSearch(value);
135   },
136   [debouncedSearch]
137 );
138
139 // Reset all filters
140 const resetFilters = useCallback(() => {
141   setSearchInput("");
142   setSearchTerm("");
143   setFilterStatus("All");
144   setSortField(null);
145   setSortDirection("asc");
146   toast({
147     title: "Filters Reset",
148     description: "All filters have been cleared.",
149   });
150 }, [toast]);
151
152 useEffect(() => {
153   const fetchData = async () => {
154     if (!session?.user?.token) {
155       toast({
156         title: "Authentication Error",
157         description: "Authentication token missing. Please log in again.",
158         variant: "destructive",
```

```
159     });
160     router.push("/login");
161     return;
162 }
163
164 try {
165     setIsLoading(true);
166     const kpiData = await fetchEmployeeKPIs(session.user.token)
167     ;
168     setKPIs(kpiData);
169
170     const { stats } = calculateKPIMetrics(kpiData);
171     setStats(stats);
172 } catch (error: any) {
173     console.error("Failed to fetch KPIs:", error);
174     toast({
175         title: "Error",
176         description: "Failed to load KPIs: " + error.message,
177         variant: "destructive",
178     });
179     setKPIs([]);
180     setStats([
181         { title: "Excellent Performers", value: 0, icon:
182             CheckCircle },
183         { title: "Good Performers", value: 0, icon: Users },
184         { title: "Needs Improvement", value: 0, icon: Clock },
185         { title: "Avg Performance Score", value: 0, icon:
186             BarChart },
```

```
184         ]));
185     } finally {
186         setIsLoading(false);
187     }
188 };
189
190     if (session) {
191         fetchData();
192     }
193 }, [session, router, toast, debouncedSearch, calculateKPIMetrics
194     ]));
195
196     const roles = session?.user?.roles
197       ? Array.isArray(session.user.roles)
198         ? session.user.roles
199         : [session.user.roles]
200       : [];
201
202     const isAdmin = roles.includes("ADMIN");
203
204     const handleSort = useCallback(
205       (field: keyof EmployeeKPI) => {
206         if (sortField === field) {
207           setSortDirection(sortDirection === "asc" ? "desc" : "asc");
208         } else {
209           setSortField(field);
210           setSortDirection("asc");
211         }
212       },
```

```
211     [sortField, sortDirection]
212   );
213
214   const handleKPIClick = useCallback(
215     (employeeId: string) => {
216       router.push(`/${protected}/kpi/employee/${employeeId}`);
217     },
218     [router]
219   );
220
221   const handleEditKPI = useCallback(
222     (employeeId: string) => {
223       router.push(`/${protected}/kpi/edit/${employeeId}`);
224     },
225     [router]
226   );
227
228   const handleDeleteKPI = useCallback(async () => {
229     if (!employeeToDelete || !session?.user?.token) return;
230
231     try {
232       await deleteKPI(employeeToDelete, session.user.token);
233       setKPIs((prev) =>
234         prev.filter((kpi) => kpi.employeeId !== employeeToDelete)
235       );
236       const { stats } = calculateKPIMetrics(
237         kpis.filter((kpi) => kpi.employeeId !== employeeToDelete)
238       );
```

```
239         setStats(stats);
240
241         toast({
242             title: "Success",
243             description: "KPI_record_deleted_successfully.",
244         });
245     } catch (error: any) {
246         console.error("Failed_to_delete_KPI:", error);
247         toast({
248             title: "Error",
249             description: "Failed_to_delete_KPI:" + error.message,
250             variant: "destructive",
251         });
252     } finally {
253         setDeleteDialogOpen(false);
254         setEmployeeToDelete(null);
255     }
256 }, [employeeToDelete, session, toast, kpis, calculateKPIMetrics])
257 ;
258
259 const sortedKPIs = useMemo(
260     () => sortKPIs(kpis, sortField, sortDirection),
261     [kpis, sortField, sortDirection]
262 );
263
264 const filteredKPIs = useMemo(
265     () => filterKPIs(sortedKPIs, searchTerm, filterStatus),
266     [sortedKPIs, searchTerm, filterStatus]
267 );
```

```

266 // Calculate metrics for Quick Info
267 const { metrics } = useMemo(
268   () => calculateKPIMetrics(kpis),
269   [kpis, calculateKPIMetrics]
270 );
271
272 if (status === "loading") {
273   return (
274     <div className="flex_min-h-screen_bg-background">
275       <div className="flex-1_flex_flex-col">
276         <main className="p-4_sm:p-6_flex-1">
277           <Skeleton className="h-8_w-[200px]_mb-6" />
278           <KPIStats stats={[]} isLoading={true} />
279         </main>
280       </div>
281     </div>
282   );
283 }
284
285 if (status === "unauthenticated" || !session) {
286   router.push("/login");
287   return null;
288 }
289
290 if (!isAdmin) {
291   toast({
292     title: "Access_Denied",
293     description: "Only_admins_can_access_this_page.",

```



```

294     variant: "destructive",
295   });
296   router.push("/protected");
297   return null;
298 }
299
300 return (
301   <div className="flex_min-h-screen_bg-background">
302     <div className="flex-1_flex_flex-col_max-w-7xl_mx-auto_w-full
303       ">
304       { /* Search and Actions */ }
305       <Card className="sticky_top-0_z-30_border-b_border-border
306         /50_bg-background/95_backdrop-blur_supports-[backdrop-
307         filter]:bg-background/60">
308         <CardContent className="flex_flex-col_sm:flex-row_items-
309           center_gap-4_p-4">
310           <div className="relative_w-full_sm:w-auto_flex-1">
311             <Search className="absolute_left-2.5_top-2.5_h-4_w-4_
312               text-muted-foreground" />
313             <Input
314               type="search"
315               value={searchInput}
316               onChange={handleSearchChange}
317               placeholder="Search_employees..."
318               className="pl-8_w-full_sm:w-[250px]_lg:w-[350px]"
319             />
320           </div>
321         </CardContent>
322       </div>
323     </div>
324   </div>
325 )

```

```

316     <div className="flex_items-center_gap-2_w-full_sm:w-
        auto_justify-between_sm:justify-end">
317         <Button onClick={() => router.push("/protected/kpi/
            create")}>
318             <CirclePlus className="mr-2_h-4_w-4" />
319             Add KPI
320         </Button>
321         <Button
322             variant="outline"
323             onClick={() => exportToCSV(filteredKPIs)}
324             disabled={isLoading}
325         >
326             <Download className="mr-2_h-4_w-4" />
327             Export CSV
328         </Button>
329         <Button variant="outline" onClick={resetFilters}>
330             <RefreshCw className="mr-2_h-4_w-4" />
331             Reset
332         </Button>
333     </div>
334 </CardContent>
335 </Card>
336
337 <main className="p-4_sm:p-6_flex-1">
338     <div className="flex_justify-between_items-center_mb-6">
339         <h1 className="text-2xl_sm:text-3xl_font-bold_text-
            foreground">
340             Employee KPIs

```

```

341     </h1>
342     <div className="flex_gap-2">
343         <Button
344             variant={viewMode === "table" ? "default" : "ghost"}
345             size="icon"
346             onClick={() => setViewMode("table")}
347         >
348             <TableIcon className="h-4_w-4" />
349         </Button>
350         <Button
351             variant={viewMode === "grid" ? "default" : "ghost"}
352             size="icon"
353             onClick={() => setViewMode("grid")}
354         >
355             <LayoutGrid className="h-4_w-4" />
356         </Button>
357         <Button
358             variant={viewMode === "list" ? "default" : "ghost"}
359             size="icon"
360             onClick={() => setViewMode("list")}
361         >
362             <List className="h-4_w-4" />
363         </Button>
364     </div>
365 </div>
366
367 {/* Quick Filters */}

```

```

368     <Card className="mb-6">
369       <CardContent className="flex flex-wrap gap-2 p-4">
370         <Button
371           variant={filterStatus === "All" ? "default" : "
372             outline"}
373           onClick={() => setFilterStatus("All")}
374         >
375           All
376         </Button>
377         <Button
378           variant={filterStatus === "Excellent" ? "default" :
379             "outline"}
380           onClick={() => setFilterStatus("Excellent")}
381         >
382           Excellent
383         </Button>
384         <Button
385           variant={filterStatus === "Good" ? "default" : "
386             outline"}
387           onClick={() => setFilterStatus("Good")}
388         >
389           Good
390         </Button>
391         <Button
392           variant={
393             filterStatus === "NeedsImprovement" ? "default"
394             : "outline"
395           }

```

```

392         onClick={() => setFilterStatus("Needs_Improvement")}
393     }
394     >
395         Needs Improvement
396     </Button>
397 </CardContent>
398 </Card>
399
400 <KPIStats stats={stats} isLoading={isLoading} />
401
402 <div className="grid_gap-4_md:grid-cols-2_mb-6">
403     <Card>
404         <CardHeader>
405             <CardTitle>Actions</CardTitle>
406         </CardHeader>
407         <CardContent>
408             <KPIActions
409                 isLoading={isLoading}
410                 onCreate={() => router.push("/protected/kpi/create")}
411                 onGenerateReport={() => setReportDialogOpen(true)}
412                 onViewExcellent={() =>
413                     router.push("/protected/kpi/excellent")
414                 }
415                 onViewAll={() => router.push("/protected/kpi/all")}
416             />

```

```

416         </CardContent>
417     </Card>
418     <Card>
419         <CardHeader>
420             <CardTitle>Quick Info</CardTitle>
421         </CardHeader>
422         <CardContent>
423             {isLoading ? (
424                 <Skeleton className="h-20 w-full" />
425             ) : (
426                 <div className="space-y-4">
427                     <div className="flex justify-between items-
428                         center">
429                         <span className="text-sm text-muted-
430                             foreground">
431                             Total Employees
432                         </span>
433                         <span className="text-sm font-medium">
434                             {metrics.totalEmployees}
435                         </span>
436                     </div>
437                     <Separator />
438                     <div className="flex justify-between items-
439                         center">
440                         <span className="text-sm text-muted-
441                             foreground">
442                             Excellent
443                         </span>

```

```

440         <span className="text-sm font-medium">
441             {metrics.excellentPerformers}
442         </span>
443     </div>
444     <Separator />
445     <div className="flex justify-between items-
446         center">
447         <span className="text-sm text-muted-
448             foreground">
449             Avg Score
450         </span>
451         <span className="text-sm font-medium">
452             {metrics.avgPerformanceScore}
453         </span>
454     </div>
455 </div>
456     </div>
457     </div>
458
459 <Card>
460     <CardHeader>
461         <CardTitle>Employee KPI List</CardTitle>
462     </CardHeader>
463     <CardContent>
464         {isLoading ? (
465             <div className="space-y-4">

```

```

466         <Skeleton className="h-8 w-full" />
467         <Skeleton className="h-32 w-full" />
468     </div>
469 ) : (
470     <>
471         {viewMode === "table" && (
472             <KPITable
473                 kpis={filteredKPIs}
474                 onSort={handleSort}
475                 onClick={handleKPIClick}
476                 onEdit={handleEditKPI}
477                 onDelete={(id) => {
478                     setEmployeeToDelete(id);
479                     setDeleteDialogOpen(true);
480                 }}
481             />
482         )}
483         {viewMode === "grid" && (
484             <KPIGrid
485                 kpis={filteredKPIs}
486                 onClick={handleKPIClick}
487                 onEdit={handleEditKPI}
488                 onDelete={(id) => {
489                     setEmployeeToDelete(id);
490                     setDeleteDialogOpen(true);
491                 }}
492             />
493         )}

```



```

494         {viewMode === "list" && (
495             <KPIList
496                 kpis={filteredKPIs}
497                 onClick={handleKPIClick}
498                 onEdit={handleEditKPI}
499                 onDelete={(id) => {
500                     setEmployeeToDelete(id);
501                     setDeleteDialogOpen(true);
502                 }}
503             />
504         )}
505     </>
506 )}
507 </CardContent>
508 </Card>
509 </main>
510
511 <DeleteDialog
512     open={deleteDialogOpen}
513     onOpenChange={setDeleteDialogOpen}
514     onConfirm={handleDeleteKPI}
515 />
516
517 <KPIReportDialog
518     open={reportDialogOpen}
519     onOpenChange={setReportDialogOpen}
520     kpis={kpis}
521     config={reportConfig}

```

```

522         setConfig={setReportConfig}
523         onGenerate={() => {
524             generatePerformanceReport(kpis, reportConfig);
525             toast({
526                 title: "Success",
527                 description: "Performance_report_generated_
                    successfully_as_PDF.",
528             });
529             setReportDialogOpen(false);
530         }}
531     />
532 </div>
533 </div>
534 );
535 };
536
537 export default KPIPage;

```

Код 4.7: src/app/protected/kpi/page.tsx [front-end]

```

539 package services
540
541 import (
542     "errors"
543     "time"
544
545     "github.com/amgaland/epes/epes-back/config"
546     "github.com/amgaland/epes/epes-back/models"
547     "github.com/google/uuid"

```

```
548 )
549
550 func CreateNewKPI(kpi models.KPI) (models.KPI, error) {
551     kpi.ID = uuid.New().String()
552     kpi.CreatedAt = time.Now()
553     if err := config.DB.Create(&kpi).Error; err != nil {
554         return models.KPI{}, err
555     }
556     return kpi, nil
557 }
558
559 func CreateNewEmployeeKPI(empKPI models.EmployeeKPI) (models.
    EmployeeKPI, error) {
560     empKPI.ID = uuid.New().String()
561     empKPI.EvaluatedAt = time.Now()
562
563     score := empKPI.PerformanceScore
564     if score >= 90 {
565         empKPI.Status = "Excellent"
566     } else if score >= 75 {
567         empKPI.Status = "Good"
568     } else {
569         empKPI.Status = "Needs_Improvement"
570     }
571
572     if err := config.DB.Create(&empKPI).Error; err != nil {
573         return models.EmployeeKPI{}, err
574     }
}
```

```
575     return empKPI, nil
576 }
577
578 func GetAllEmployeeKPIs() ([]models.EmployeeKPI, error) {
579     var kpis []models.EmployeeKPI
580     if err := config.DB.Preload("Employee").Find(&kpis).Error; err != nil
581     {
582         return nil, err
583     }
584     return kpis, nil
585 }
586
587 func GetEmployeeKPIByID(empID string) (models.EmployeeKPI, error) {
588     var kpi models.EmployeeKPI
589     if err := config.DB.Where("employee_id_=? ", empID).Preload("Employee
590     ").First(&kpi).Error; err != nil {
591         return models.EmployeeKPI{}, err
592     }
593     return kpi, nil
594 }
595
596 func DeleteKPIByID(id string) error {
597     if err := config.DB.Delete(&models.EmployeeKPI{}, "id_=? ", id).Error
598     ; err != nil {
599         return err
600     }
601     return nil
602 }
```

```

600
601 func UpdateEmployeeKPIByID(id string, input models.EmployeeKPI) (models
    .EmployeeKPI, error) {
602     var existing models.EmployeeKPI
603     if err := config.DB.First(&existing, "id=?", id).Error; err != nil
        {
604         return models.EmployeeKPI{}, errors.New("KPI not found")
605     }
606
607     input.Status = "NeedsImprovement"
608     if input.PerformanceScore >= 90 {
609         input.Status = "Excellent"
610     } else if input.PerformanceScore >= 75 {
611         input.Status = "Good"
612     }
613
614     if err := config.DB.Model(&existing).Updates(input).Error; err != nil
        {
615         return models.EmployeeKPI{}, err
616     }
617     return existing, nil
618 }

```

Код 4.8: kpi api services [backend]

МФШ303	Тайлан гаргадаг байх
МФШ304	Тайланг PDF эсвэл CSV татаж авах боломжтой байх

```

1 // src/app/protected/kpi/utils/kpiUtils.ts

```

```
2 import jsPDF from "jspdf";
3 import { EmployeeKPI, ReportConfig } from "../types";
4
5 export function sortKPIs(
6   kpis: EmployeeKPI[],
7   sortField: keyof EmployeeKPI | null,
8   sortDirection: "asc" | "desc"
9 ): EmployeeKPI[] {
10   if (!sortField) return kpis;
11   return [...kpis].sort((a, b) => {
12     const aValue = a[sortField];
13     const bValue = b[sortField];
14
15     if (sortField === "employeeName") {
16       return sortDirection === "asc"
17         ? typeof aValue === "string" && typeof bValue === "string"
18           ? aValue.localeCompare(bValue)
19           : 0
20         : typeof aValue === "string" && typeof bValue === "string"
21           ? bValue.localeCompare(aValue)
22           : 0;
23     }
24
25     if (typeof aValue === "number" && typeof bValue === "number") {
26       return sortDirection === "asc" ? aValue - bValue : bValue -
27         aValue;
28     }
29   });
30 }
```

```

29     if (typeof aValue === "string" && typeof bValue === "string") {
30         return sortDirection === "asc"
31             ? aValue.localeCompare(bValue)
32             : bValue.localeCompare(aValue);
33     }
34
35     return 0;
36 });
37 }
38
39 export function filterKPIs(
40     kpis: EmployeeKPI[],
41     searchTerm: string,
42     filterStatus: "All" | "Excellent" | "Good" | "Needs_Improvement"
43 ): EmployeeKPI[] {
44     return kpis.filter(
45         (kpi) =>
46             kpi.employeeName.toLowerCase().includes(searchTerm.toLowerCase())
47             &&
48             (filterStatus === "All" || kpi.status === filterStatus)
49     );
50 }
51
52 export function exportToCSV(kpis: EmployeeKPI[]): void {
53     const headers = [
54         "Employee_ID,Employee_Name,Task_Completion_Rate_%,Tasks_Completed
55         ,Tasks_Assigned,Project_Contribution_%,Projects_Assigned,
56         Performance_Score,Status",

```

```
54 ];
55 const rows = kpis.map(
56   (k) =>
57     `${k.employeeId},${k.employeeName},${k.taskCompletionRate},${k.
58       tasksCompleted},${k.tasksAssigned},${k.projectContribution},${
59       k.projectsAssigned},${k.performanceScore},${k.status}`
60 );
61 const csvContent = [headers, ...rows].join("\n");
62 const blob = new Blob([csvContent], { type: "text/csv;charset=utf-8;"
63   });
64 const link = document.createElement("a");
65 link.href = URL.createObjectURL(blob);
66 link.setAttribute("download", "employee_kpis.csv");
67 document.body.appendChild(link);
68 link.click();
69 document.body.removeChild(link);
70 }
71
72 export function generatePerformanceReport(
73   kpis: EmployeeKPI[],
74   config: ReportConfig
75 ): void {
76   const { employeeId, period, includeTasks, includeProjects,
77     includeComments } =
78     config;
79   let filteredKPIs = kpis;
80   if (employeeId !== "all") {
81     filteredKPIs = kpis.filter((kpi) => kpi.employeeId === employeeId);
```



```
78   }
79
80   const periodFilter = (date: string) => {
81     if (date === "N/A") return true;
82     const taskDate = new Date(date);
83     const now = new Date();
84     if (period === "last30days") {
85       return taskDate >= new Date(now.getTime() - 30 * 24 * 60 * 60 *
86         1000);
87     } else if (period === "last90days") {
88       return taskDate >= new Date(now.getTime() - 90 * 24 * 60 * 60 *
89         1000);
90     }
91     return true;
92   };
93
94   const doc = new jsPDF();
95   const pageWidth = doc.internal.pageSize.getWidth();
96   const margin = 15;
97   let yOffset = margin;
98
99   doc.setFillColor(0, 102, 204);
100  doc.rect(0, 0, pageWidth, 30, "F");
101  doc.setFont("helvetica", "bold");
102  doc.setFontSize(18);
103  doc.setTextColor(255, 255, 255);
104  doc.text("Performance_Evaluation_Report", pageWidth / 2, 20, {
105    align: "center",
```

```
104   });
105
106   doc.setTextColor(0, 0, 0);
107   doc.setFont("helvetica", "normal");
108   doc.setFontSize(10);
109   yOffset += 25;
110
111   doc.setFont("helvetica", "italic");
112   doc.text(
113     `Generated on: ${new Date().toISOString().split("T")[0]}`,
114     margin,
115     yOffset
116   );
117   yOffset += 5;
118   doc.text(
119     `Period: ${
120       period === "allTime"
121         ? "All-Time"
122         : period === "last30days"
123           ? "Last 30 Days"
124           : "Last 90 Days"
125     }`,
126     margin,
127     yOffset
128   );
129   yOffset += 10;
130
131   filteredKPIs.forEach((kpi, index) => {
```

```
132     if (yOffset > 260) {
133         doc.addPage();
134         yOffset = margin;
135     }
136
137     doc.setFont("helvetica", "bold");
138     doc.setFontSize(14);
139     doc.text(kpi.employeeName, margin, yOffset);
140     yOffset += 5;
141     doc.setLineWidth(0.5);
142     doc.line(margin, yOffset, pageWidth - margin, yOffset);
143     yOffset += 10;
144
145     doc.setFont("helvetica", "normal");
146     doc.setFontSize(10);
147     const metrics = [
148         { label: "Performance□Score", value: `${kpi.performanceScore}` },
149         { label: "Status", value: kpi.status },
150         { label: "Task□Completion□Rate", value: `${kpi.taskCompletionRate}%` },
151         {
152             label: "Tasks□Completed",
153             value: `${kpi.tasksCompleted}/${kpi.tasksAssigned}`,
154         },
155         { label: "Project□Contribution", value: `${kpi.projectContribution}%` },
156         { label: "Projects□Assigned", value: `${kpi.projectsAssigned}` },
157     ];
```

```
158
159     doc.setFillColor(230, 230, 230);
160     doc.rect(margin, yOffset, 80, 8, "F");
161     doc.rect(margin + 80, yOffset, 80, 8, "F");
162     doc.setFont("helvetica", "bold");
163     doc.text("Metric", margin + 2, yOffset + 6);
164     doc.text("Value", margin + 82, yOffset + 6);
165     yOffset += 8;
166
167     doc.setFont("helvetica", "normal");
168     metrics.forEach((metric, idx) => {
169         doc.setFillColor(idx % 2 === 0 ? 240 : 255, 255, 255);
170         doc.rect(margin, yOffset, 80, 8, "F");
171         doc.rect(margin + 80, yOffset, 80, 8, "F");
172         doc.text(metric.label, margin + 2, yOffset + 6);
173         doc.text(metric.value, margin + 82, yOffset + 6);
174         yOffset += 8;
175     });
176     yOffset += 10;
177
178     if (includeTasks && kpi.tasks) {
179         doc.setFont("helvetica", "bold");
180         doc.text("Tasks", margin, yOffset);
181         yOffset += 5;
182         doc.setLineWidth(0.2);
183         doc.line(margin, yOffset, pageWidth - margin, yOffset);
184         yOffset += 5;
185         doc.setFont("helvetica", "normal");
```

```
186     const filteredTasks = kpi.tasks.filter((task) =>
187         periodFilter(task.dueDate)
188     );
189     filteredTasks.forEach((task) => {
190         if (yOffset > 260) {
191             doc.addPage();
192             yOffset = margin;
193         }
194         const taskText = `${task.title} (Status: ${task.status}, Due: $
195             {task.dueDate})`;
196         const splitText = doc.splitTextToSize(
197             taskText,
198             pageWidth - 2 * margin - 5
199         );
200         doc.text(splitText, margin + 5, yOffset);
201         yOffset += splitText.length * 6 + 2;
202     });
203     yOffset += 5;
204
205     if (includeProjects && kpi.projects) {
206         doc.setFont("helvetica", "bold");
207         doc.text("Projects", margin, yOffset);
208         yOffset += 5;
209         doc.setLineWidth(0.2);
210         doc.line(margin, yOffset, pageWidth - margin, yOffset);
211         yOffset += 5;
212         doc.setFont("helvetica", "normal");
```

```

213     kpi.projects.forEach((project) => {
214         if (yOffset > 260) {
215             doc.addPage();
216             yOffset = margin;
217         }
218         const projectText = `${project.name} (Progress: ${project.
                progress}%)`;
219         const splitText = doc.splitTextToSize(
220             projectText,
221             pageWidth - 2 * margin - 5
222         );
223         doc.text(splitText, margin + 5, yOffset);
224         yOffset += splitText.length * 6 + 2;
225     });
226     yOffset += 5;
227 }
228
229 if (includeComments) {
230     doc.setFont("helvetica", "bold");
231     doc.text("Comments", margin, yOffset);
232     yOffset += 5;
233     doc.setLineWidth(0.2);
234     doc.line(margin, yOffset, pageWidth - margin, yOffset);
235     yOffset += 5;
236     doc.setFont("helvetica", "normal");
237     const comment =
238         kpi.status === "Excellent"
239         ? "Outstanding performance with consistent task completion"

```

```

        and_significant_project_contributions."
240     : kpi.status === "Good"
241     ? "Solid performance, meeting expectations in tasks and
        projects."
242     : "Improvement needed in task completion and/or project
        contributions.";

    const splitComment = doc.splitTextToSize(
243         comment,
244         pageWidth - 2 * margin - 5
245     );
246
247     doc.text(splitComment, margin + 5, yOffset);
248     yOffset += splitComment.length * 6 + 5;
249 }
250
251 if (index < filteredKPIs.length - 1) {
252     yOffset += 10;
253 }
254 });
255
256 const pageCount = doc.getNumberOfPages();
257 for (let i = 1; i <= pageCount; i++) {
258     doc.setPage(i);
259     doc.setFontSize(8);
260     doc.setTextColor(100, 100, 100);
261     doc.text(
262         `Page ${i} of ${pageCount}`,
263         pageWidth - margin,
264         doc.internal.pageSize.getHeight() - 10,

```

```
265     { align: "right" }
266   );
267 }
268
269 doc.save(
270   `performance_report_${
271     employeeId === "all"
272       ? "all_employees"
273       : kpis.find((k) => k.employeeId === employeeId)?.employeeName
274       ||
275       "employee"
276   }_${period}.pdf`
277 );
```

Код 4.9: Тайлан автоматаар гаргах

5. ДҮГНЭЛТ

Энэхүү дипломын ажил нь бизнесийн байгууллагад зориулсан вэб-д суурилсан Гүйцэтгэлийн Үнэлгээний Системийг (EPES) хөгжүүлэх зорилготой байв. Уг систем нь төсөл, даалгаврын удирдлага, гүйцэтгэлийн үнэлгээ, тайлангийн автоматаар үүсгэх зэрэг функцуудыг хангаж, орчин үеийн технологийн шийдлүүдийг ашиглан бизнесийн үр ашгийг дээшлүүлэхэд чиглэсэн юм. Энэ хэсэгт судалгааны зорилго, хүрсэн үр дүн, тулгарсан сорилтууд болон ирээдүйн хөгжлийн чиглэлийг нэгтгэн дүгнэнэ.

5.0.1 Зорилго ба Хувь Нэмэр

Судалгааны гол зорилго нь Монголын бизнесийн орчинд тохирсон, хэрэглэхэд хялбар гүйцэтгэлийн үнэлгээний системийг бий болгох байв. EPES нь Next.js, Golang, PostgreSQL, Docker зэрэг технологиудыг ашиглан хөгжүүлэгдсэн бөгөөд Админ, Менежер, Ажилтан гэсэн хэрэглэгчийн гурван эрхийг дэмждэг. Систем нь жинлэсэн дундаж аргыг ашиглан гүйцэтгэлийн оноог тооцоолж, PDF болон CSV форматаар тайлан гаргах боломжийг олгодог. Энэхүү ажлын гол хувь нэмэр нь:

- Бизнесийн онцлогт нийцсэн, нээлттэй эхийн технологи ашигласан системийн загвар.
- Даалгаврын удирдлага, гүйцэтгэлийн үнэлгээний автоматжуулалтыг нэгтгэсэн цогц шийдэл.
- Interactive.mn, Asana, Lattice зэрэг системүүдтэй харьцуулсан шинжилгээ, EPES-ийн давуу талыг онцолсон судалгаа.

5.0.2 Хүрсэн Үр Дүн

Системийн хөгжүүлэлт амжилттай хэрэгжиж, функциональ болон функциональ бус шаардлагуудыг хангасан. Тестийн үр дүнгээс харахад EPES нь хэрэглэгчийн интерфэйсийн хувьд хялбар, хариу өгөх хугацаа хурдан, өгөгдлийн бүрэн байдлыг хангасан байв. Жинлэсэн дундаж аргыг ашиглан гүйцэтгэлийн оноог тооцоолох нь үнэлгээний ил тод, шударга байдлыг хангаж, Монголын

жижиг болон дунд бизнесийн хэрэгцээнд нийцсэн болохыг харуулсан. Харьцуулсан шинжилгээгээр EPES нь Lattice, BambooHR зэрэг системүүдтэй харьцуулахад нээлттэй эхийн уян хатан байдал, хямд өртөгөөрөө давуу талтай болохыг тогтоосон.

5.0.3 Тулгарсан Сорилтууд

Хөгжүүлэлтийн явцад зарим сорилтуудтай тулгарсан, үүнд:

- Next.js болон Golang-ийн хоорондын API интеграцын нарийн тохиргоо.
- Жижиг бизнесүүдийн хязгаарлагдмал техникийн мэдлэгт нийцүүлэн хэрэглэгчийн интерфэйсийг хялбаршуулах шаардлага.

GORM ашиглан өгөгдлийн сангийн хандалтыг оновчтой болгож, UI/UX дизайныг хэрэглэгчид нийцүүлэн өөрчилсөн.

5.0.4 Ирээдүйн Хөгжлийн Чиглэл

EPES системийн ирээдүйн дараах чиглэлүүдээр нэмж хөгжүүлэх боломжтой:

- Хиймэл оюун ухааны алгоритмуудыг нэвтрүүлж, гүйцэтгэлийн үнэлгээнд урьдчилсан таамаглал, автоматжуулсан санал болгох функцуудыг нэмэх.
- 360 хэмжээний санал хүсэлтийн модулийг хөгжүүлж, илүү цогц үнэлгээний загварыг дэмжих.
- Мобайл апп хувилбарыг хөгжүүлж, хэрэглэгчийн хүртээмжийг нэмэгдүүлэх.

EPES систем нь бизнесийн орчинд гүйцэтгэлийн үнэлгээний процессыг автоматжуулах, ил тод байдлыг хангах чухал алхам болсон. Энэхүү ажил нь нээлттэй эхийн технологиудын давуу талыг харуулж, жижиг болон дунд бизнесүүдэд хямд, үр ашигтай шийдэл санал болгосон. Цаашид хиймэл оюун ухаан, тасралтгүй санал хүсэлтийн загваруудыг нэвтрүүлснээр системийн чадавхийг улам өргөжүүлэх боломжтой. Энэхүү судалгаа нь Монголын аж ахуйн нэгжүүдийн гүйцэтгэлийн удирдлагын ирээдүйн хөгжилд хувь нэмэр оруулна гэдэгт итгэлтэй байна.

Bibliography

- [1] R. Islam and S. M. Rasad, “Employee performance evaluation by the AHP: A case study,” *Asia Pacific Management Review*, vol. 11, no. 3, pp. 163–176, Jun. 2006. [Online]. Available: <https://rafikulislam.com/uploads/myworks/27066075955b8a1b374434.pdf>
- [2] A. Shaout and M. K. Yousif, “Performance evaluation – Methods and techniques survey,” *MCST Journal*, vol. 3, no. 5, pp. 66–74, Sep. 2014. [Online]. Available: <https://d1wqtxts1xz1e7.cloudfront.net/41797022/Paper030516-libre.pdf>
- [3] Next.js Team, “Next.js documentation,” Vercel Inc., 2025. [Online]. Available: <https://nextjs.org/docs>
- [4] The Go Authors, “The Go programming language documentation,” 2025. [Online]. Available: <https://go.dev/doc/>
- [5] PostgreSQL Global Development Group, “PostgreSQL documentation,” 2025. [Online]. Available: <https://www.postgresql.org/docs/>
- [6] Docker Inc., “Docker documentation,” 2025. [Online]. Available: <https://docs.docker.com/>
- [7] M. Armstrong, *Performance Management: Key Strategies and Practical Guidelines*, 5th ed. London, UK: Kogan Page, 2015.
- [8] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 6th ed. Upper Saddle River, NJ: Pearson, 2021.

A. BACK-END ХЭРЭГЖҮҮЛЭЛТ

Өгөгдлийн санд байрлах бүхий л өгөгдлүүд рүү хандах API-ууд route-үүд

```
1 package routes
2
3 import (
4     "github.com/amgaland/epes/epes-back/controllers"
5     admin "github.com/amgaland/epes/epes-back/controllers/admin"
6     api "github.com/amgaland/epes/epes-back/controllers/api"
7     protected "github.com/amgaland/epes/epes-back/controllers/protected"
8     "github.com/gin-gonic/gin"
9 )
10
11 func RegisterRoutes(router *gin.Engine) {
12     router.GET("/health", controllers.HealthCheck)
13
14     authRoutes := router.Group("/auth")
15     {
16         authRoutes.POST("/signin", api.SignIn)
17     }
18
19     adminRoutes := router.Group("/admin")
20     {
21         userRoutes := adminRoutes.Group("/users")
22         {
23             userRoutes.GET("/", admin.GetAllUsers)
24             userRoutes.POST("/", admin.CreateUser)
25             userRoutes.PUT("/:id", admin.UpdateUser)
26             userRoutes.DELETE("/:id", admin.DeleteUser)
27             userRoutes.GET("/check-login-id", admin.CheckLoginIDExists)
28         }
29
30         userRoleRoutes := adminRoutes.Group("/user/roles")
31         {
32             userRoleRoutes.GET("/", admin.GetAllUserRoles)
33             userRoleRoutes.POST("/", admin.CreateUserRole)
34             userRoleRoutes.PUT("/:id", admin.UpdateUserRole)
35             userRoleRoutes.DELETE("/:id", admin.DeleteUserRole)
36             userRoleRoutes.GET("/list", admin.UserRoleHandler)
37             userRoleRoutes.PUT("/update", admin.UpdateUserRoleHandler)
38         }
39
40         roleRoutes := adminRoutes.Group("/roles")
41         {
42             roleRoutes.GET("/", admin.GetAllRoles)
43             roleRoutes.POST("/", admin.CreateRole)
44             roleRoutes.PUT("/:id", admin.UpdateRole)
45             roleRoutes.DELETE("/:id", admin.DeleteRole)
46         }
47     }
```

```

48     rolePermissionRoutes := adminRoutes.Group("/role-permissions")
49     {
50         rolePermissionRoutes.GET("/", admin.GetAllRolePermissions)
51         rolePermissionRoutes.POST("/", admin.CreateRolePermission)
52         rolePermissionRoutes.PUT("/:id", admin.UpdateRolePermission
53             )
54         rolePermissionRoutes.DELETE("/:id", admin.
55             DeleteRolePermission)
56         rolePermissionRoutes.GET("/list", admin.
57             RolePermissionHandler)
58         rolePermissionRoutes.PUT("/update", admin.
59             UpdateRolePermissionHandler)
60     }
61
62     actionTypeRoutes := adminRoutes.Group("/action-types")
63     {
64         actionTypeRoutes.GET("/", admin.GetAllActionTypes)
65         actionTypeRoutes.POST("/", admin.CreateActionType)
66         actionTypeRoutes.PUT("/:id", admin.UpdateActionType)
67         actionTypeRoutes.DELETE("/:id", admin.DeleteActionType)
68     }
69 }
70
71 protectedRoutes := router.Group("/protected")
72 {
73     departmentRoutes := protectedRoutes.Group("/departments")
74     {
75         departmentRoutes.GET("/", protected.GetAllDepartments)
76         departmentRoutes.POST("/", protected.CreateDepartment)
77         departmentRoutes.PUT("/:id", protected.UpdateDepartment)
78         departmentRoutes.DELETE("/:id", protected.DeleteDepartment)
79     }
80
81     userDepartmentRoutes := protectedRoutes.Group("/user/
82         departments")
83     {
84         userDepartmentRoutes.GET("/", protected.
85             GetAllUserDepartments)
86         userDepartmentRoutes.POST("/", protected.
87             CreateUserDepartment)
88         userDepartmentRoutes.PUT("/:id", protected.
89             UpdateUserDepartment)
90         userDepartmentRoutes.DELETE("/:id", protected.
91             DeleteUserDepartment)
92         userDepartmentRoutes.GET("/list", protected.
93             UserDepartmentHandler)
94         userDepartmentRoutes.PUT("/update", protected.
95             UpdateUserDepartmentHandler)
96     }
97     taskRoutes := protectedRoutes.Group("/tasks")
98     {
99         taskRoutes.GET("/", protected.GetAllTasks)

```

```

89     taskRoutes.POST("/", protected.CreateTask)
90     taskRoutes.PUT("/:id", protected.UpdateTask)
91     taskRoutes.DELETE("/:id", protected.DeleteTask)
92     taskRoutes.GET("/check-task-id", protected.
        CheckTaskIDExists)
93 }
94 projectRoutes := protectedRoutes.Group("/projects")
95 {
96     projectRoutes.GET("/", protected.GetAllProjects)
97     projectRoutes.POST("/", protected.CreateProject)
98     projectRoutes.PUT("/:id", protected.UpdateProject)
99     projectRoutes.DELETE("/:id", protected.DeleteProject)
100    projectRoutes.GET("/tasks/:id", protected.
        GetAllProjectTasks)
101 }
102
103 projectMemberRoutes := projectRoutes.Group("/members")
104 {
105     projectMemberRoutes.GET("/project/member", protected.
        GetProjectMember)
106     projectMemberRoutes.POST("/project/member", protected.
        CreateProjectMember)
107     projectMemberRoutes.DELETE("/project/member/:id", protected.
        DeleteProjectMember)
108 }
109
110
111 kpiRoutes := protectedRoutes.Group("/kpi")
112 {
113     kpiRoutes.POST("/", protected.CreateKPI)
114     kpiRoutes.POST("/employee-kpi", protected.CreateEmployeeKPI
        )
115     kpiRoutes.POST("/employee-kpi/:id", protected.
        CreateEmployeeKPI)
116     kpiRoutes.GET("/", protected.GetEmployeeKPIs)
117     kpiRoutes.GET("/employee-kpi/:id", protected.
        GetEmployeeKPIByID)
118     kpiRoutes.DELETE("/employee-kpi/:id", protected.
        DeleteEmployeeKPI)
119     kpiRoutes.PUT("/employee-kpi/:id", protected.
        UpdateEmployeeKPI)
120 }
121 employeeRoutes := protectedRoutes.Group("/employees")
122 {
123     employeeRoutes.GET("/", protected.GetAllEmployees)
124     employeeRoutes.POST("/", protected.CreateEmployee)
125     employeeRoutes.PUT("/:id", protected.UpdateEmployee)
126     employeeRoutes.DELETE("/:id", protected.DeleteEmployee)
127     employeeRoutes.GET("/:id", protected.GetEmployeeByID)
128 }
129
130 feedbackRoutes := protectedRoutes.Group("/feedback")

```

```

131     {
132         feedbackRoutes.GET("/", protected.GetAllFeedback)
133         feedbackRoutes.POST("/", protected.CreateFeedback)
134         feedbackRoutes.PUT("/:id", protected.UpdateFeedback)
135         feedbackRoutes.DELETE("/:id", protected.DeleteFeedback)
136         feedbackRoutes.GET("/:id", protected.GetFeedbackByID)
137     }
138 }
139 }
140
141 }

```

Код A.1: Routes

Back-End моделүүд

```

1
2 type Model struct {
3     ID          string          `json:"id" gorm:"default:gen_random_uuid()"`
4     CreatedAt    time.Time      `json:"created_at" gorm:"default:now()"`
5     UpdatedAt    time.Time      `json:"updated_at" gorm:"default:now()"`
6     CreatedBy    *string        `json:"created_by" gorm:"references:User:ID"`
7     UpdatedBy    *string        `json:"updated_by" gorm:"references:User:ID"`
8 }
9
10
11 type User struct {
12     Model
13     FirstName    string          `json:"first_name"`
14     LastName     string          `json:"last_name"`
15     LoginID      string          `json:"login_id" gorm:"unique"`
16     EmailWork    string          `json:"email_work"`
17     EmailPersonal *string        `json:"email_personal"`
18     PhoneNumberWork *string        `json:"phone_number_work"`
19     PhoneNumberPersonal *string        `json:"phone_number_personal"`
20     IsActive     *bool          `json:"is_active"`
21     ActiveStartDate time.Time      `json:"active_start_date"`
22     ActiveEndDate *time.Time     `json:"active_end_date"`
23     Password     string          `json:"password"`
24 }
25
26 type LoginUser struct {
27     LoginID string `json:"login_id"`
28     Password string `json:"password"`
29 }
30
31 type UserWithRoles struct {
32     ID          string          `json:"id"`
33     FirstName    string          `json:"first_name"`
34     LastName     string          `json:"last_name"`
35     EmailPersonal string          `json:"email_personal"`
36     EmailWork    string          `json:"email_work"`
37     LoginID      string          `json:"login_id"`

```

```

38  PhoneNumberPersonal string    `json:"phone_number_personal"`
39  PhoneNumberWork     string    `json:"phone_number_work"`
40  IsActive            bool      `json:"is_active"`
41  Token               string    `json:"token"`
42  ActiveStartDate      time.Time `json:"active_start_date"`
43  ActiveEndDate        *time.Time `json:"active_end_date,omitempty"`
44  Roles               string    `json:"roles"`
45 }
46
47 type Role struct {
48     Model
49     Name string `json:"name" gorm:"unique"`
50 }
51
52 type Project struct {
53     Model
54     Name          string    `json:"name"`
55     Description    string    `json:"description"`
56     StartDate      time.Time `json:"start_date"`
57     EndDate        *time.Time `json:"end_date"`
58     Status         string    `json:"status"` // e.g., "
59     Ongoing", "Completed", "Delayed"
60     OwnerID        string    `json:"owner_id"` // Reference to
61     User           User      `json:"owner" gorm:"foreignKey:OwnerID
62     ;references:ID"`
63     // Relationship to team members
64     TeamMembers    []ProjectMember `json:"team_members" gorm:"foreignKey:
65     ProjectID"`
66 }
67
68 type ProjectMember struct {
69     Model
70     ProjectID      string    `json:"project_id"`
71     Project        Project   `json:"project" gorm:"foreignKey:ProjectID;
72     references:ID"`
73     UserID         string    `json:"user_id"`
74     User          User      `json:"user" gorm:"foreignKey:UserID;
75     references:ID"`
76     RoleInProject  string    `json:"role_in_project" // Optional: "
77     Manager", "Developer", "QA", etc.
78 }
79
80 type EmployeeEvaluationReport struct {
81     Model
82     EmployeeID     string    `json:"employee_id"`
83     Employee       Employee  `gorm:"foreignKey:EmployeeID"`
84     Period         string    `json:"period" // e.g., "Q1 2025"
85     ReportData     string    `json:"report_data" // JSON or HTML summary
86     GeneratedBy    string    `json:"generated_by"`
87     GeneratedAt    time.Time `json:"generated_at"`
88 }

```



```

83 type Task struct {
84     Model
85     ProjectID    string    `json:"project_id"`
86     Project      Project   `json:"project" gorm:"foreignKey:ProjectID;
      references:ID"`
87     Title        string    `json:"title"`
88     Description  string    `json:"description"`
89     AssignedToID string    `json:"assigned_to_id"`
90     AssignedTo   User      `json:"assigned_to" gorm:"foreignKey:
      AssignedToID;references:ID"`
91     Status       string    `json:"status"`           // e.g., "Pending",
      "In Progress", "Completed"
92     Deadline     *time.Time `json:"deadline"`
93     CompletedAt  *time.Time `json:"completed_at"`
94 }
95
96 type TaskFeedback struct {
97     Model
98     TaskID       string    `json:"task_id"`
99     Task         Task      `json:"task" gorm:"foreignKey:TaskID;references:ID
      "`
100     EvaluatorID  string    `json:"evaluator_id"`
101     Evaluator    User      `json:"evaluator" gorm:"foreignKey:EvaluatorID;
      references:ID"`
102     Comment      string    `json:"comment"`
103     Rating       int       `json:"rating"`   // 1-5 or percentage
104 }

```

Код A.2: Models