

**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 Үндэслэл .....	4
2.2 Онолын судалгаа .....	6
2.3 Ижил төстэй системүүд .....	14
2.4 Технологийн судалгаа .....	19
2.5 Бүлгийн дүгнэлт .....	25
3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ .....	27
3.1 Системийн хэрэглэгчид .....	27
3.2 Системийн шаардлага .....	27
3.3 Системийн загвар .....	39
3.4 Бүлгийн дүгнэлт .....	46
4. ХЭРЭГЖҮҮЛЭЛТ .....	47
4.1 Файлын бүтэц .....	47
4.2 Хөгжүүлэлтийн орчин бүрдүүлэлт .....	49
4.3 Хэрэгжүүлэлт .....	54
5. ДҮГНЭЛТ .....	58
5.1 Дүгнэлт .....	58
НОМ ЗҮЙ .....	61
ХАВСРАЛТ .....	62
A. REST API-ИЙН ЖИШЭЭ КОД (GOLANG, GIN) .....	62

В. BACK-END ХЭРЭГЖҮҮЛЭЛТ .....	63
С. FRONT-END ЖИШЭЭ КОД (NEXT.JS).....	69

## ЗУРГИЙН ЖАГСААЛТ

3.1	Нэвтрэх хуудас .....	31
3.2	Админ дашбоард харагдац .....	32
3.3	Ажилтны жагсаалт .....	33
3.4	Төслийн харагдац .....	34
3.5	Даалгаварын харагдац .....	35
3.6	Гүйцэтгэлийн үнэлгээний харагдац .....	36
3.7	Тайлан гаргах компонентийн харагдац .....	37
3.8	Хэрэглэгчийн профайлын харагдац .....	38
3.9	Системийн архитектурын диаграмм .....	39
3.10	Системийн ажлын явцын диаграмм .....	40
3.11	Төсөл болон даалгаврын харилцан хамаарлын диаграмм .....	41
3.12	Ажилтны нэгж хоорондын харилцан хамаарлын диаграмм .....	42
3.13	Ажилтан болон түүний эрхийн харилцан хамаарлын диаграмм .....	43
3.14	Админ дарааллын диаграмм .....	44
3.15	Менежер дарааллын диаграмм .....	45
3.16	Ажилтан дарааллын диаграмм .....	46
4.1	Front-end файлын бүтэц .....	47
4.2	Back-end файлын бүтэц .....	48
4.3	Docker Desktop програмын интерфэйс .....	49
4.4	Postman файлын бүтэц .....	53
4.5	Postman хэрэглэгч үүсгэх тест .....	54

## ХҮСНЭГТИЙН ЖАГСААЛТ

2.1	Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт .....	16
2.2	Интеграцын харьцуулалт .....	17
3.1	Ерөнхий шаардлага .....	28
3.2	Админ шаардлага .....	28
3.3	Менежер шаардлага .....	29
3.4	Ажилтан шаардлага .....	29
3.5	Функциональ биш шаардлага .....	30
3.6	Функциональ биш шаардлага .....	30

# Кодын жагсаалт

4.1	Dockerfile . . . . .	49
4.2	docker-compose.yaml . . . . .	50
4.3	Bun суулгах . . . . .	51
4.4	Nextjs суулгах . . . . .	51
4.5	Shaden суулгах . . . . .	51
4.6	Golang суулгах . . . . .	52
A.1	Routes . . . . .	62
B.1	Routes . . . . .	63
B.2	Models . . . . .	66
C.1	Routes . . . . .	69



## 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 вебд 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 Үндэслэл

Бизнесийн байгууллагын ажилчны гүйцэтгэлийг үнэлэх системийн сэдвийг сонгох нь нийгмийн, эдийн засгийн, технологийн шалтгаан нөхцөл, шаардлага, түүнчлэн энэхүү судалгааны үр нөлөөтэй нягт холбоотой юм. Эдгээр хүчин зүйлсийг доор тодорхой тайлбарлав.

#### 2.1.1 Нийгмийн шалтгаан нөхцөл

Орчин үеийн нийгэмд ажилтны гүйцэтгэлийн үнэлгээ нь зөвхөн байгууллагын зорилгод хүрэхээс гадна ажилтны хувь хүний хөгжил, сэтгэл ханамжийг дэмжихэд чухал үүрэг гүйцэтгэж байна. Монголын хөдөлмөрийн зах зээлд ажилтны ажлын байрны тогтвортой байдал, сэдэлжилд, ур чадварыг дээшлүүлэх нь нийгмийн хөгжлийн нэгэн чухал хэсэг болоод байна. Судалгаагаар, гүйцэтгэлийн үнэлгээний ил тод, шударга систем ашигладаг байгууллагуудын ажилтны ажлын сэтгэл ханамж 18 хувиар өсдөг [1]. Иймээс энэхүү сэдвийг сонгосон нь нийгэмд ажилтны хөгжил, байгууллагын соёлыг сайжруулахад хувь нэмэр оруулах зорилготой юм.

#### 2.1.2 Эдийн засгийн шалтгаан нөхцөл

Ажилчны гүйцэтгэлийн үнэлгээний систем нь байгууллагын эдийн засгийн үр ашгийг нэмэгдүүлэхэд шууд нөлөөтэй. Бизнесийн байгууллагуудын хувьд уламжлалт үнэлгээний аргууд нь цаг хугацаа их шаардлагатай, алдаа гарах эрсдэлтэй байдаг бөгөөд энэ нь ажилтан ажилдаа сэтгэл ханамжгүй байх мөн түүнчлэн ажил хаялтад хүргэдэг. Автоматжуулсан гүйцэтгэлийн үнэлгээний систем нэвтрүүлсэн байгууллагуудын үйл ажиллагааны зардал дунджаар 15 хувиар буурч, ажилтны бүтээмж 20 хувиар өсдөг болохыг олон улсын судалгаа харуулж байна [2]. Энэ сэдвийг сонгосон нь бизнесийн байгууллагуудын өрсөлдөх чадварыг нэмэгдүүлж, эдийн засгийн хувьд илүү үр ашигтай шийдэл санал болгоход чиглэгдэнэ.

**2.1.3 Технологийн шалтгаан нөхцөл**

Мэдээллийн технологийн хурдацтай хөгжил нь бизнесийн процессуудыг автоматжуулах, өгөгдөлд суурилсан шийдвэр гаргалтыг нэмэгдүүлэх боломжийг олгож байна. Next.js, Golang, PostgreSQL, Docker зэрэг орчин үеийн технологиудыг ашигласан вэбд суурилсан гүйцэтгэлийн үнэлгээний систем нь бодит цагийн мэдээлэл боловсруулалт, хэрэглэгчийн ээлтэй интерейсээр хангадаг. Монголын бизнесийн байгууллагуудын дийлэнх нь ийм технологийн шийдлийг харахан нэвтрүүлээгүй байгаа нь энэ сэдвийн технологийн хэрэгцээг онцолж байна. Судалгаагаар, гүйцэтгэлийн үнэлгээний процессыг автоматжуулсанаар процессын хугацаа 35 хувиар буурдаг [3].

**2.1.4 Шаардлага**

Монголын бизнесийн орчинд гүйцэтгэлийн үнэлгээний автоматжуулсан системийн хэрэгцээ улам бүр нэмэгдэж байна. Уламжлалт аргууд нь субъектив байдал, мэдээллийн алдагдал, процессын удаан байдал зэрэг асуудлуудыг дагуулдаг. Түүнчлэн 360 хэмжээний санал хүсэлт, OKR, KPI зэрэг орчин үеийн үнэлгээний аргуудыг нэвтрүүлэх шаардлага Монголын байгууллагуудад тулгарч байна. Энэхүү судалгаа нь эдгээр шаардлагыг хангах, Монголын бизнесийн онцлогт тохирсон, найдвартай, хялбар хэрэглэгдэхүйц системийн загварыг боловсруулахыг зорьж байна.

**2.1.5 Үр нөлөө**

Энэхүү судалгааны үр дүнд боловсруулагдсан гүйцэтгэлийн үнэлгээний систем нь дараах үр нөлөөг бий болгоно:

- **Нийгмийн хувьд:** ажилтны ажлын сэтгэл ханамж, сэдэлжилтийг дээшлүүлж, ажилтны хөгжил, байгууллагын соёлыг сайжруулна.
- **Эдийн засгийн хувьд:** Байгууллагын үйл ажиллагааны зардлыг бууруулж, бүтээмжийг нэмэгдүүлснээр эдийн засгийн үр ашгийг дээшлүүлнэ.

- **Технологийн хувьд:** Монголын мэдээллийн технологийн салбарт шинэ загвар, технологийн шийдлийг нэвтрүүлж, дижитал шилжилтийг дэмжинэ.
- **Практикийн хувьд:** Байгууллагуудын гүйцэтгэлийн үнэлгээний процессыг ил тод, хурдан, үр ашигтай болгоно.

Эдгээр шалтгаан, шаардлага, үр нөлөөний үндсэн дээр ”Бизнесийн байгууллагын ажилчны гүйцэтгэлийг үнэлэх систем” сэдвийг сонгосон нь Монголын бизнесийн орчинд шинжлэх ухаан, практикийн хувьд онцгой ач холбогдолтой юм.

## 2.2 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, АНР<sup>1</sup> 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

---

<sup>1</sup> АНР - Analytical Hierarchy Process

- Ур чадварын зөрүүг илрүүлэх, түүнийг нөхөхөд хийгдэх сургалт, хөгжлийн төлөвлөгөө боловсруулах

## 2. Уламжлалт аргачилалууд

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

- 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
  - Үнэлэгээ гаргагчийн хувийн хандлага (субъектив байдал)

- Бүх шалгуурыг адил жинтэйд тооцдог

(с) **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. Wehrich МВО-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

(с) **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

(е) **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.

## 2.3 Ижил төстэй системүүд

Дипломын ажлын хүрээнд хөгжүүлж буй ажилчны гүйцэтгэлийн үнэлгээний систем (Employee Performance Evaluation System, EPES)-тэй ижил төстэй үйл ажиллагаа явуулдаг хоёр системийг нарийвчлан судалж, харьцуулна. Судалгаанд олон улсын зах зээлд танигдсан **Lattice** болон **BambooHR** системүүдийг сонгож, тэдгээрийн байгууллагын танилцуулга, гүйцэтгэлийн үнэлгээний үндсэн модулиудыг тодорхойлно. Эдгээр системүүдийг EPES-тэй харьцуулж, функциональ болон техникийн шинж чанаруудын давуу тал, хязгаарлалтуудыг шинжилнэ. Энэхүү харьцуулалт нь Монголын бизнесийн орчинд тохирсон гүйцэтгэлийн үнэлгээний системийн загварыг боловсруулахад чиглэнэ.

### 2.3.1 Lattice-ийн онцлогууд

- **Байгууллагын товч танилцуулга:** Lattice нь АНУ-д байрладаг, ажилчны гүйцэтгэлийн удирдлага болон ажилтны оролцоог дэмжих чиглэлээр мэргэшсэн программ хангамжийн компани юм (<https://lattice.com>). 2015 онд байгуулагдсан тус компани нь жижиг, дунд, том хэмжээний байгууллагуудад зориулсан SaaS платформ санал болгодог бөгөөд 2025 оны байдлаар дэлхий даяар 5,000 гаруй байгууллага тус системийг ашиглаж байна. Lattice-ийн гол онцлог нь гүйцэтгэлийн үнэлгээ, зорилго тогтоох (OKR), 360 хэмжээний санал хүсэлт, ажилтны хөгжлийн төлөвлөгөөг нэгтгэсэн хэрэглэгчдэд ээлтэй платформ юм.
- **Ажилтны гүйцэтгэлийг үнэлэх үндсэн модуль:** Lattice-ийн гүйцэтгэлийн удирдлагын модуль нь KPI болон OKR-д суурилсан зорилго тогтоох, 360 хэмжээний санал хүсэлтийн систем, бодит цагийн гүйцэтгэлийн хяналт, аналитикийн хэрэгслүүдийг багтаадаг. Хэрэглэгчид ажилтны гүйцэтгэлийн талаар тогтмол санал хүсэлт өгч, хувь хүний болон багийн зорилгын ахицыг хянах боломжтой. Мөн уг систем нь Slack, Microsoft Teams зэрэг гуравдагч талын платформтой интеграцлагддаг. Гэсэн хэдий ч Lattice-ийн системийн өндөр өртөг болон том хэмжээний тохируулгын хязгаарлалт нь жижиг байгууллагуудад саад болж

болно.

### 2.3.2 *BambooHR-ийн онцлогууд*

- **Байгууллагын товч танилцуулга:** BambooHR нь АНУ-ын Юта мужид байрладаг, хүний нөөцийн удирдлагын (HRM) программ хангамжийн чиглэлээр үйл ажиллагаа явуулдаг компани юм (<https://www.bamboohr.com>). 2008 онд байгуулагдсан тус компани нь голчлон жижиг болон дунд хэмжээний байгууллагуудад зориулсан цогц HRM шийдэл санал болгодог. BambooHR-ийн платформ нь ажилтны гүйцэтгэлийн үнэлгээ, хүний нөөцийн мэдээллийн удирдлага, ажилд авах процессыг автоматжуулахад чиглэдэг бөгөөд 2025 оны байдлаар 30,000 гаруй байгууллага уг системийг ашиглаж байна.
- **Ажилтны гүйцэтгэлийг үнэлэх үндсэн модуль:** BambooHR-ийн гүйцэтгэлийн үнэлгээний модуль нь ажилтны зорилго тогтоох, KPI-д суурилсан үнэлгээ, тогтмол санал хүсэлтийн систем, гүйцэтгэлийн тайлан зэргийг багтаадаг. Уг модуль нь менежерүүдэд ажилтны гүйцэтгэлийн талаар хялбаршуулсан тайлан гаргах боломжийг олгодог бөгөөд хэрэглэгчдэд ээлтэй интерфэйстэй. Гэсэн хэдий ч, 360 хэмжээний санал хүсэлтийн боломж хязгаарлагдмал бөгөөд бодит цагийн аналитикийн хувьд Lattice-ээс харьцангуй сул юм. Мөн интеграцын боломжууд нь хязгаарлагдмал бөгөөд том байгууллагуудын нарийн шаардлагыг хангахад хангалтгүй байж болно.

### 2.3.3 *Харьцуулалт*

Lattice болон BambooHR-ийн системүүдийг дипломын ажлын хүрээнд хөгжүүлж буй EPES системтэй харьцуулж, функциональ болон техникийн онцлогуудыг шинжилнэ. EPES нь Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), PostgreSQL, Docker зэрэг орчин үеийн технологиудыг ашиглан хөгжүүлэгдэж байгаа бөгөөд бодит цагийн мэдээлэлд суурилсан гүйцэтгэлийн үнэлгээ, 360 хэмжээний санал хүсэлт, OKR, KPI-д чиглэсэн модультай.

## Дэлгэрэнгүй харьцуулалтын хүснэгт

Онцлог	Lattice	BambooHR	EPES
Танилцуулга	АНУ-ын гүйцэтгэлийн удирдлагын SaaS	АНУ-ын HRM платформ	Дипломын ажлын хүрээнд хөгжүүлэгдсэн вэб систем
Үнэлгээний модуль	OKR, KPI, 360 хэмжээний санал хүсэлт	KPI, тогтмол санал хүсэлт	OKR, KPI, 360 хэмжээний санал хүсэлт
Технологи	Cloud-based, JavaScript	Cloud-based, тодорхой бус	Golang, Next.js, PostgreSQL
Интерфэйс	Хэрэглэгчдэд ээлтэй	Хялбар, энгийн	Tailwind CSS-ээр хариу үйлдэлтэй
Бодит цагийн хяналт	Бүрэн дэмждэг	Хязгаарлагдмал	Бүрэн дэмждэг
Тохируулга	Дунд зэрэг	Дунд зэрэг	Өндөр (нээлттэй эх)
Аюулгүй байдал	JWT, OAuth, HTTPS	HTTPS, тодорхой бус	JWT, HTTPS
Хэрэглээний хүрээ	Жижиг, дунд, том бизнес	Жижиг, дунд бизнес	Төрөл бүрийн байгууллага
Интеграцын боломж	Slack, Microsoft Teams	Хязгаарлагдмал	REST API-аар дэмжигдэнэ
Скалируемость	Өндөр	Дунд зэрэг	Docker, Kubernetes-ээр өндөр

Хүснэгт 2.1: Ижил төстэй системүүдийн дэлгэрэнгүй харьцуулалт

## Интеграцын харьцуулалт

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



Онцлог	Lattice	BambooHR	EPES
Хэрэглэгчийн хэмжээ	Жижиг, дунд, том бизнес	Жижиг, дунд бизнес	Жижиг, дунд бизнес
Серверын архитектур	Cloud-based	Cloud-based	Microservices, Docker, Kubernetes
Интеграцын API	REST API	Хязгаарлагдмал API	REST API
Гуравдагч талын хэрэгсэл	Slack, Microsoft Teams, Workday	Хязгаарлагдмал	Потенциалтай (нээлттэй эх)
Өгөгдлийн хэмжээний дэмжлэг	Өндөр	Дунд зэрэг	Өндөр (PostgreSQL)

Хүснэгт 2.2: Интеграцын харьцуулалт

#### 2.3.4 Шинжилгээ ба дүгнэлт

- **Lattice:** Гүйцэтгэлийн удирдлагын чиглэлээр мэргэшсэн, OKR болон 360 хэмжээний санал хүсэлтийн модуль нь том байгууллагуудад тохиромжтой. Гэсэн хэдий ч өндөр өртөг, тохируулгын хязгаарлалт нь Монголын жижиг, дунд бизнесүүдэд саад болж болно. Интеграцын боломж өндөр боловч нээлттэй эхийн шинж чанаргүй.
- **BambooHR:** Жижиг, дунд бизнесүүдэд зориулсан энгийн, хэрэглэгчдэд ээлтэй гүйцэтгэлийн үнэлгээний модультай. Гэсэн хэдий ч 360 хэмжээний санал хүсэлт, бодит цагийн аналитикийн хувьд хязгаарлагдмал бөгөөд интеграцын боломж сул. Монголын зах зээлд нэвтрүүлэхэд тохируулга шаардлагатай.
- **EPES:** Бодит цагийн мэдээлэлд суурилсан үнэлгээ, нээлттэй эхийн технологи, өндөр тохируулгатай байдал зэргээрээ онцлог. Docker болон Kubernetes-ийн дэмжлэг нь том хэмжээний байгууллагуудад тохиромжтой болгодог. Гэсэн хэдий ч хөгжүүлэлтийн эхний шатандаа байгаа тул туршилт, баталгаажуулалт шаардлагатай. REST API болон нээлттэй

эхийн шинж чанар нь ирээдүйд гуравдагч талын интеграцыг өргөжүүлэх боломжтой.

### 2.3.5 EPES-ийн боломжит сайжруулалт

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

- **Аналитикийн модуль:** Нарийвчилсан KPI болон OKR-д суурилсан аналитикийн алгоритмуудыг нэмж, гүйцэтгэлийн урьдчилсан таамаглал хийх боломжтой болгох.
- **Интеграцын өргөтгөл:** Slack, Microsoft Teams зэрэг алдартай SaaS платформтой шууд интеграцлах боломжийг нэмэх.
- **Хэрэглэгчийн туршлага:** Lattice-ийн түвшний UI/UX-ийг хангахын тулд хэрэглэгчийн санал хүсэлтэд суурилсан туршилт хийх.
- **Хамгаалалт:** JWT-ийн хажуугаанд OAuth 2.0 болон нэмэлт шифрлэлтийн протоколуудыг нэвтрүүлж, өгөгдлийн аюулгүй байдлыг сайжруулах.

## 2.4 Технологийн судалгаа

Энэ хэсэгт дипломын ажлын хүрээнд хөгжүүлж буй ажилчны гүйцэтгэлийн үнэлгээний систем (Employee Performance Evaluation System, EPES)-д ашиглагдах технологийн судалгааг хийж, тэдгээрийн онолын үндэс, практик хэрэглээ, давуу тал, сул тал, түүнчлэн EPES системийн шаардлагад хэрхэн тохиромжтой болохыг шинжилнэ. Судалгаа нь **Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), Docker, Postman, DBeaver, PostgreSQL** зэрэг технологийг хамарна. Эдгээр технологиудыг сонгосон нь EPES системийн бодит цагийн гүйцэтгэлийн хяналт, 360 хэмжээний санал хүсэлт, OKR болон KPI-д суурилсан үнэлгээний зорилгод нийцсэн, Монголын бизнесийн орчинд тохирсон, өндөр гүйцэтгэлтэй, өргөтгөх боломжтой вэб систем хөгжүүлэхэд чиглэгдсэн болно.

### 2.4.1 Golang

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

#### Gin

- **Онолын үндэс ба практик хэрэглээ:** Gin нь Go-д зориулсан хөнгөн, өндөр гүйцэтгэлтэй HTTP вэб фреймворк бөгөөд Radix модны алгоритмаар HTTP чиглүүлэлтийг хурдан гүйцэтгэдэг. Энэ нь RESTful API-уудыг хялбархан хөгжүүлэхэд зориулагдсан бөгөөд middleware дэмжлэг, хүсэлтийн боловсруулалтыг хангадаг.
- **Давуу тал:**
  - Хамгийн бага нөөцийн зарцуулалттай, хурдан гүйцэтгэлтэй.
  - Микро үйлчилгээний архитектурт тохиромжтой.
  - Middleware-ийн дэмжлэгээр аюулгүй байдал, лог хөтлөлтийг хялбаршуулна.
- **Сул тал:** Нарийн тохиргоо шаардлагагүй жижиг төслүүдэд хэт хүнд байж болно.

- **EPES-д яагаад тохиромжтой вэ:** EPES систем нь бодит цагийн гүйцэтгэлийн хяналт, олон хэрэглэгчийн хүсэлтийг зэрэг боловсруулах шаардлагатай тул Gin-ийн хурдан гүйцэтгэл, бага нөөцийн зарцуулалт нь системийн backend-ийн найдвартай ажиллагааг хангана. Монголын жижиг, дунд бизнесүүдэд хямд, хурдан API хөгжүүлэхэд Gin-ийн хялбар хэрэглээ тохиромжтой. Жишээлбэл, ажилтны гүйцэтгэлийн мэдээллийг бодит цагт боловсруулж, хэрэглэгчдэд хурдан хүргэхэд Gin-ийн чиглүүлэлтийн хурд онцгой ач холбогдолтой.

## GORM

- **Онолын үндэс ба практик хэрэглээ:** GORM нь Go-д зориулсан ORM (Object-Relational Mapping) сан бөгөөд PostgreSQL-ийн өгөгдлийн сан хоорондын харилцааг автоматжуулж, хөгжүүлэлтийн хугацааг хэмнэдэг. Энэ нь migrations, queries, transactions зэрэг үйлдлүүдийг хялбаршуулдаг.
- **Давуу тал:**
  - SQL код бичих хугацааг хэмнэж, хөгжүүлэлтийг хурдасгана.
  - Preloading, hooks зэрэг онцлог нь нийлмэл өгөгдлийн удирдлагыг хялбаршуулна.
  - Автомат миграци нь өгөгдлийн сангийн схемийн өөрчлөлтийг хялбар болгоно.
- **Сул тал:** Нарийн SQL асуулгад хязгаарлалттай тул зарим тохиолдолд гараар SQL бичих шаардлага гарна.
- **EPES-д яагаад тохиромжтой вэ:** EPES систем нь гүйцэтгэлийн үнэлгээний өгөгдөл (KPI, OKR, санал хүсэлт) хадгалах, хурдан хайлт хийх шаардлагатай. GORM-ийн автомат миграци, хялбаршуулсан асуулгын боловсруулалт нь өгөгдлийн сангийн удирдлагыг хурдасгаж, хөгжүүлэлтийн явцад өгөгдлийн загварыг хялбар өөрчлөх боломжийг олгоно. Монголын бизнесийн байгууллагуудын хувьд энгийн бөгөөд хурдан хөгжүүлэлтийн шийдэл шаардлагатай бөгөөд GORM энэ шаардлагыг хангана.

## JWT

- **Онолын үндэс ба практик хэрэглээ:** JWT (JSON Web Token) нь HMAC криптографийн алгоритмаар токены агуулгыг баталгаажуулж, төлөвгүй (stateless) баталгаажуулалтыг хангадаг. Энэ нь хэрэглэгчийн нэвтрэлтийг найдвартай удирдахад ашиглагддаг.
- **Давуу тал:**
  - Сервер дээр session хадгалах шаардлагагүй тул өргөтгөхөд хялбар.
  - REST API-д нийцтэй, хялбар хэрэгжүүлэлттэй.
- **Сул тал:** Токеныг хулгайлах эрсдэл бий тул HTTPS-ийн хамт ашиглах шаардлагатай.
- **EPES-д яагаад тохиромжтой вэ:** EPES системд хэрэглэгчийн (менежер, ажилтан) баталгаажуулалт гүйцэтгэлийн мэдээллийн аюулгүй хандалтыг хангах нь чухал. JWT-ийн төлөвгүй баталгаажуулалт нь олон хэрэглэгчийн хүсэлтийг зэрэг боловсруулахад серверийн ачааллыг бууруулж, Монголын жижиг, дунд байгууллагуудын хязгаарлагдмал нөөцөд тохирно. Жишээлбэл, 360 хэмжээний санал хүсэлтийн мэдээллийг зөвхөн баталгаажсан хэрэглэгчид хандах боломжтой болгоход JWT онцгой үүрэгтэй.

### 2.4.2 Next.js ба Tailwind CSS

Next.js нь React-д суурилсан фреймворк бөгөөд сервер талын рендеринг (SSR) болон статик сайтын үүсгэлтийг дэмждэг бол Tailwind CSS нь utility-first зарчимд суурилсан CSS фреймворк юм.

- **Онолын үндэс ба практик хэрэглээ:** Next.js нь виртуал DOM болон SSR-ийн хослолоор хуудасны ачааллыг хурдасгадаг бөгөөд Tailwind CSS нь урьдчилан тодорхойлсон utility классуудыг ашиглан UI хөгжүүлэлтийг хялбаршуулдаг.
- **Давуу тал:**

- Next.js: SEO-д ээлтэй, хэрэглэгчийн туршлагыг сайжруулна.
- Tailwind: Кодын давхцлыг багасгаж, загварыг хурдан өөрчлөх боломжтой.
- **Сул тал:**
  - Next.js: SSR нь серверын ачааллыг нэмэгдүүлнэ.
  - Tailwind: Том төслүүдэд классын удирдлага төвөгтэй болж болно.
- **EPES-д яагаад тохиромжтой вэ:** EPES системийн фронтенд хэсэг нь менежер, ажилтнуудад зориулсан хэрэглэгчдэд ээлтэй, хариу үйлдэлтэй интерфэйсийг шаарддаг. Next.js-ийн SSR болон хурдан рендеринг нь гүйцэтгэлийн тайлан, санал хүсэлтийн хуудсыг хурдан харуулахад тохиромжтой бөгөөд SEO-д ээлтэй байдал нь системийн хүртээмжийг нэмэгдүүлнэ. Tailwind CSS-ийн уян хатан загварчлал нь Монголын бизнесийн байгууллагуудын брэндийн онцлогт тохируулан интерфэйсийг хурдан өөрчлөх боломжийг олгоно, жишээлбэл, KPI хяналтын самбарыг байгууллагын шаардлагад нийцүүлэн загварчлахад хялбар.

### 2.4.3 Docker

- **Онолын үндэс ба практик хэрэглээ:** Docker нь контейнержуулалтын технологи бөгөөд OS-ийн виртуалчлалын зарчмаар ажилладаг бөгөөд програмыг тусгаарлагдсан орчинд ажиллуулна.
- **Давуу тал:**
  - Орчны тогтвортой байдлыг хангана.
  - Хувилбарын хяналт, нэвтрүүлэлтийг хялбаршуулна.
- **Сул тал:** Нөөцийн хэрэглээ ихтэй тул жижиг төслүүдэд хэт хүнд байж болно.
- **EPES-д яагаад тохиромжтой вэ:** EPES системийн хөгжүүлэлт, нэвтрүүлэлтэнд орчны тогтвортой байдал чухал бөгөөд Docker-ийн контейнержуулалт нь Golang, PostgreSQL

зэрэг бүрэлдэхүүнийг ижил орчинд найдвартай ажиллуулна. Монголын бизнесийн байгууллагуудыг хувьд хязгаарлагдмал серверын нөөцтэй ажиллах шаардлага байдаг тул Docker-ийн хөнгөн, стандартчилагдсан орчин нь нэвтрүүлэлтийг хялбаршуулж, засвар үйлчилгээний зардлыг бууруулна. Жишээлбэл, системийн шинэчлэлтийг контейнерээр хурдан нэвтрүүлэх боломжтой.

#### 2.4.4 Postman

- **Онолын үндэс ба практик хэрэглээ:** Postman нь API туршилтын хэрэгсэл бөгөөд REST архитектурын стандартыг дагаж, API-ийн хүсэлт, хариуг автоматжуулан шалгадаг.
- **Давуу тал:**
  - API-ийн гүйцэтгэлийг хурдан шалгах боломжтой.
  - Хамтын ажиллагааг дэмжиж, туршилтын автоматжуулалтыг хялбаршуулна.
- **Сул тал:** Том төслүүдэд скриптүүдийн удирдлага нарийн болж болно.
- **EPES-д яагаад тохиромжтой вэ:** EPES системийн Gin-д суурилсан REST API-ийн найдвартай байдлыг баталгаажуулах нь чухал. Postman-ийн хэрэглэгчдэд ээлтэй интерфэйс, автоматжуулсан туршилтын боломж нь API-ийн гүйцэтгэлийн үнэлгээний хүсэлтийг (жишээлбэл, KPI тайлангийн асуулга) хурдан шалгахад тусална. Монголын хөгжүүлэгчдийн багуудын хувьд Postman-ийн энгийн хэрэглээ нь туршилтын процессыг хялбаршуулж, хөгжүүлэлтийн хугацааг хэмнэнэ.

#### 2.4.5 DBever

- **Онолын үндэс ба практик хэрэглээ:** DBever нь өгөгдлийн сангийн GUI удирдлагын хэрэгсэл бөгөөд SQL стандартыг дэмжиж, өгөгдлийн сангийн схемийг графикаар удирддаг.
- **Давуу тал:**

- Схемийн визуалчлал, асуулгын дибаг хийхэд хялбар.
- Олон төрлийн өгөгдлийн санг дэмждэг.
- **Сул тал:** Том хэмжээний өгөгдөлтэй ажиллахад удаан байж болно.
- **EPES-д яагаад тохиромжтой вэ:** EPES системийн PostgreSQL өгөгдлийн сан нь гүйцэтгэлийн мэдээлэл, санал хүсэлтийн өгөгдлийг хадгалдаг тул DBeaver-ийн график удирдлага нь өгөгдлийн загварыг хялбар шалгаж, GORM-ийн асуулгын үр дүнг баталгаажуулахад тусална. Монголын хөгжүүлэгчдийн хувьд DBeaver-ийн хэрэглэгчдэд ээлтэй интерфэйс нь өгөгдлийн сангийн засвар үйлчилгээг хялбаршуулж, хөгжүүлэлтийн явцад алдааг хурдан илрүүлэх боломжийг олгоно.

#### 2.4.6 PostgreSQL

- **Онолын үндэс ба практик хэрэглээ:** PostgreSQL нь харилцааны өгөгдлийн сан бөгөөд ACID (Atomicity, Consistency, Isolation, Durability) зарчмыг хангаж, найдвартай өгөгдлийн удирдлага хийдэг. JSON дэмжлэгтэй тул нийлмэл өгөгдлийг хадгалах боломжтой.
- **Давуу тал:**
  - Индексжүүлэлт, өндөр гүйцэтгэл нь том хэмжээний өгөгдөлтэй ажиллахад тохиромжтой.
  - JSON болон бусад өгөгдлийн төрлийг дэмждэг.
- **Сул тал:** Том ачаалалд нарийн тохиргоо шаардлагатай.
- **EPES-д яагаад тохиромжтой вэ:** EPES систем нь гүйцэтгэлийн үнэлгээний нийлмэл өгөгдөл (KPI, OKR, санал хүсэлтийн түүх) хадгалах, хурдан хайлт хийх шаардлагатай. PostgreSQL-ийн JSON дэмжлэг нь 360 хэмжээний санал хүсэлтийн бүтэцтэй болон бүтэцгүй өгөгдлийг хадгалахад тохиромжтой бөгөөд индексжүүлэлтийн гүйцэтгэл нь бодит цагийн тайлан гаргахад дэмжлэг болно. Монголын бизнесийн байгууллагуудын



хувьд PostgreSQL-ийн нээлттэй эхийн шинж чанар нь лицензийн зардлыг хэмнэж, уян хатан байдлыг хангана.

Эдгээр технологиудыг сонгосон нь EPES системийн өндөр гүйцэтгэл, бодит цагийн хяналт, аюулгүй байдал, хэрэглэгчдэд ээлтэй интерфэйсийн шаардлагыг хангахад чиглэгдсэн бөгөөд Монголын бизнесийн орчинд хямд, хурдан, найдвартай шийдэл санал болгоход тохиромжтой юм. Гэсэн хэдий ч технологийн хэрэгжилтэд сургалт, нөөцийн удирдлага шаардлагатай бөгөөд энэ нь системийн төлөвлөлтөд анхаарах чухал хүчин зүйл болно.

## 2.5 Бүлгийн дүгнэлт

Эндээс Lattice болон BambooHR-ийн системүүдийг дипломын ажлын хүрээнд хөгжүүлж буй ажилчны гүйцэтгэлийн үнэлгээний систем (EPES)-тэй харьцуулан судалсны үр дүнг нэгтгэн дүгнэв. Lattice нь олон улсын хэмжээнд гүйцэтгэлийн удирдлагын чиглэлээр мэргэшсэн, 360 хэмжээний санал хүсэлт, OKR-д суурилсан зорилго тогтоох зэрэг өндөр түвшний функцуудыг санал болгодог болохыг тогтоосон боловч өндөр өртөг, тохируулгын хязгаарлалт нь Монголын жижиг, дунд байгууллагуудад саад болж болно. Харин BambooHR нь жижиг, дунд бизнесүүдэд зориулсан хэрэглэгчдэд ээлтэй, KPI-д суурилсан гүйцэтгэлийн үнэлгээний модультай болох нь тодорхойлогдсон ч 360 хэмжээний санал хүсэлт, бодит цагийн аналитикийн хувьд хязгаарлагдмал, том байгууллагуудын нарийн шаардлагыг хангахад хангалтгүй байж болох юм.

Хөгжүүлж буй EPES систем нь эдгээр хоёр системийн давуу талыг хослуулсан, Монголын бизнесийн орчинд тохирсон шийдэл болохыг харууллаа. Lattice-тэй харьцуулахад EPES нь нээлттэй эхийн технологи (Golang, Next.js, PostgreSQL, Docker), өндөр тохируулгатай байдал, хямд өртөгөөрөө онцлог бөгөөд Монголын жижиг, дунд байгууллагуудын хязгаарлагдмал нөөцөд нийцнэ. BambooHR-тай харьцуулахад EPES нь бодит цагийн гүйцэтгэлийн хяналт, 360 хэмжээний санал хүсэлтийн модуль, REST API-д суурилсан интеграцын боломжоороо илүү уян хатан, өргөтгөх чадвартай. Гэсэн хэдий ч EPES-д Lattice-ийн нарийвчилсан аналитик хэрэгслүүд эсвэл BambooHR-ийн хялбаршуулсан хэрэглэгчийн туршлага бүрэн хэрэгжээгүй

байгаа нь том хэмжээний байгууллагуудад хэрэглэхэд хязгаарлалт болж болзошгүй.

Эцэст нь, энэхүү судалгаа нь EPES системийн давуу тал болох хямд байдал, локал хэрэглээнд тохирсон модульчлагдсан загвар, орчин үеийн технологийн бат бөх байдлыг онцолж, зах зээл дээрх ижил төстэй системүүдээс ялгарах боломжийг харууллаа. Цаашид системийн хөгжүүлэлтэд Lattice-ийн аналитикийн онцлогуудыг нэмж, BambooHR-ийн хялбаршуулсан интерфэйсийн элементүүдийг тусгах нь Монголын бизнесийн орчинд илүү өрсөлдөх чадвартай, цогц 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

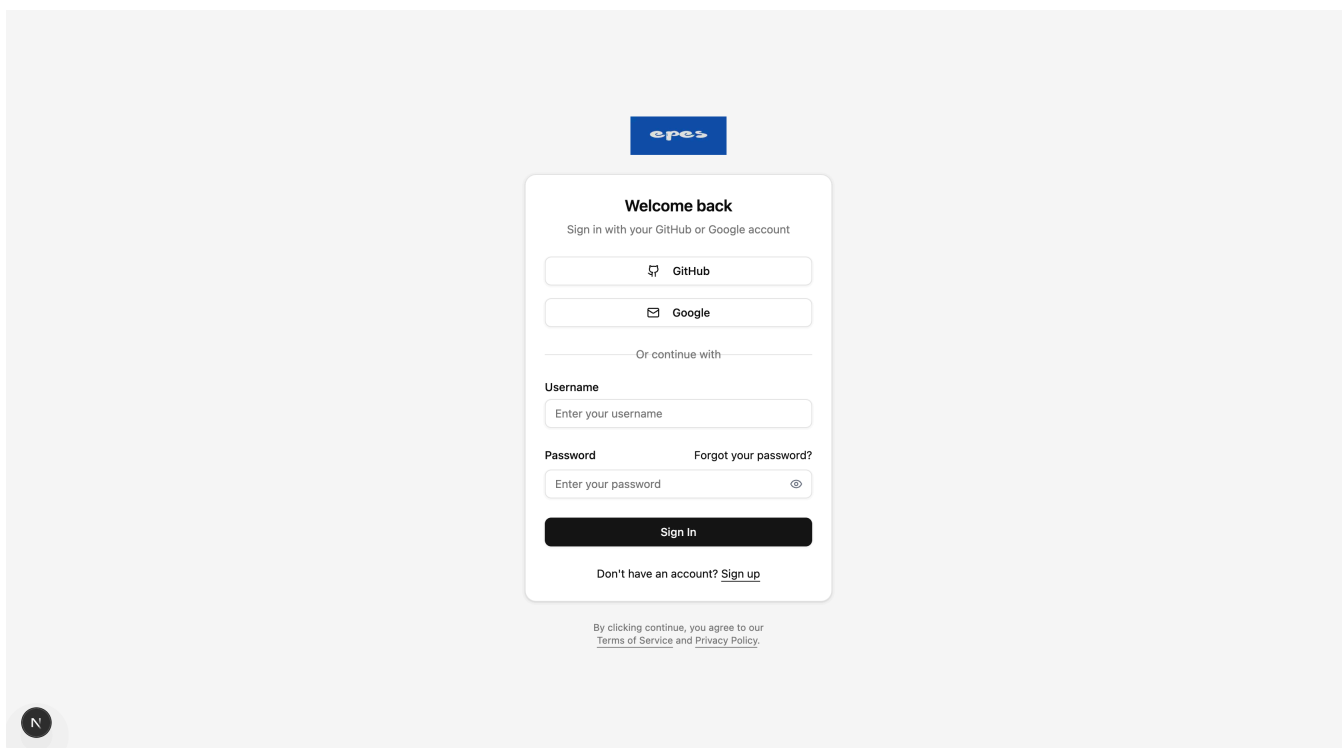
### 3.2. СИСТЕМИЙН ШААРДЛАГА

### БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

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

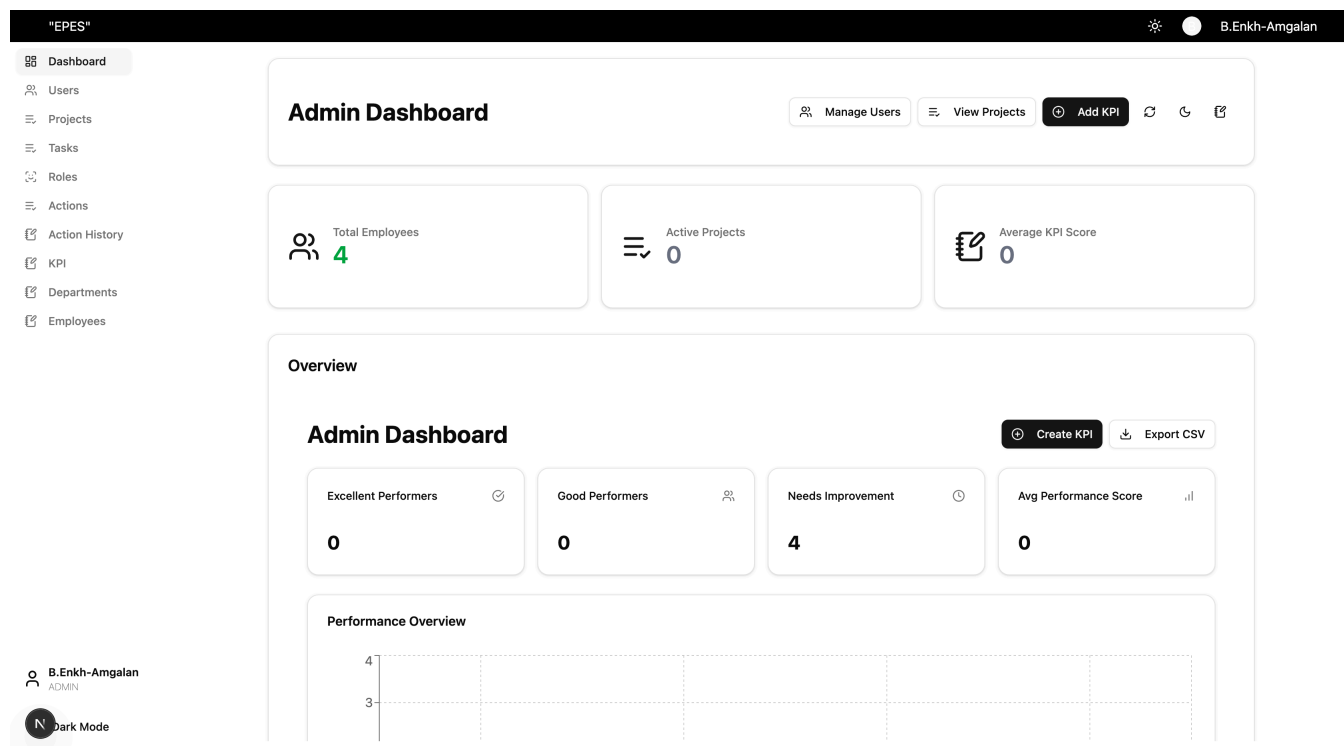
#### Нэвтрэх хуудас

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



Зураг 3.1: Нэвтрэх хуудас

## Системийн ерөнхий хуудаснууд



Зураг 3.2: Админ дашбоард харагдац



### 3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

"EPES"

Dashboard

Users

Projects

Tasks

Roles

Actions

Action History

KPI

Departments

Employees

B.Enkh-Amgalan

ADMIN

Dark Mode

Мэйл хаягаар search...

Add

Ажилчид

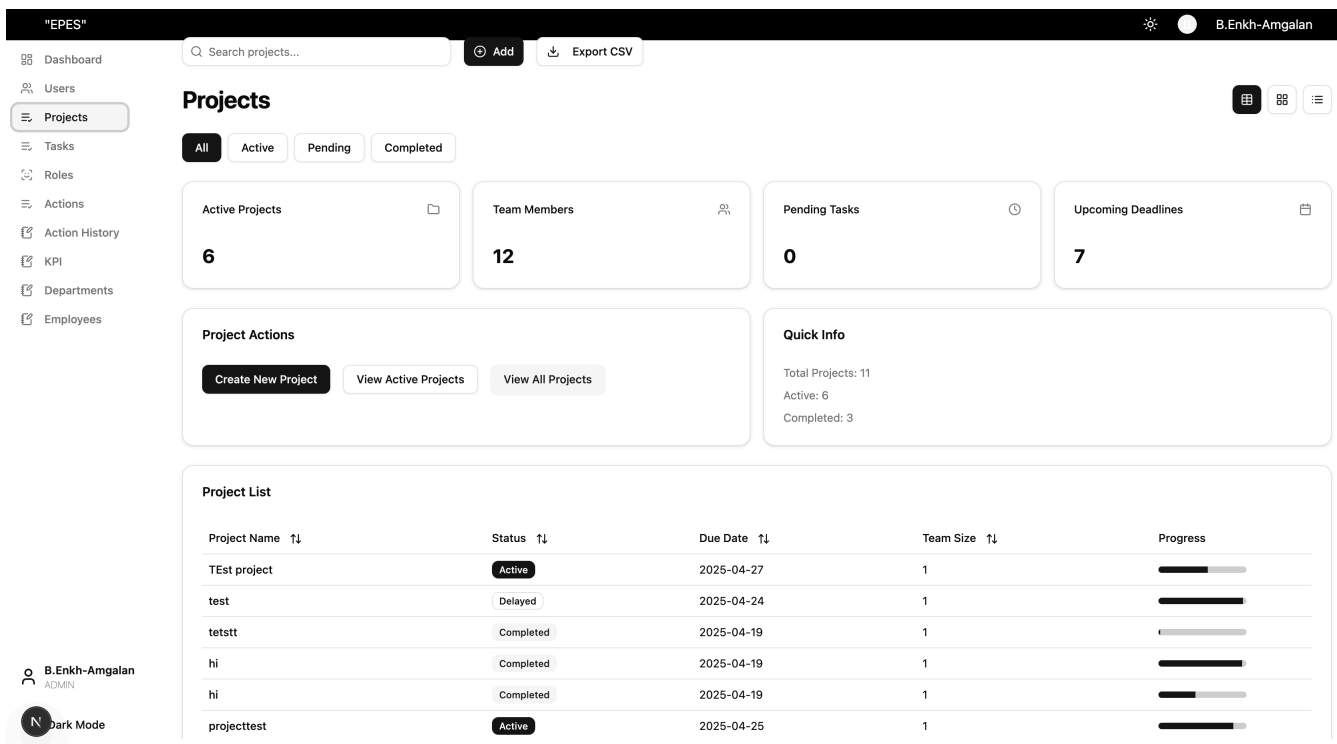
Энэ хэсэгт ажилчдийг удирдана.

Имэйл	Нэр	Нэвтрэх нэр	Дугаар	Төлөв	Бүртгүүлсэн хугацаа	Шинэчлэсэн хугацаа	
enkhmngalan.b@ondo.mn	B.Enkh-Amgalan	test	60096966	Идэвхтэй	3/19/2025	3/26/2025	...
amgalandw0822@gmail.com	m.manager	manager	95560899	Идэвхтэй	3/25/2025	4/15/2025	...
afmamf@gmail.com	t.tets	teseeet		Идэвхгүй	4/24/2025	4/24/2025	...
amgalandw0822@gmail.com	B.Enkh-Amgalan	amgalan	95560899	Идэвхтэй	4/23/2025	4/24/2025	...
amgalandw0822@gmail.com	e.employee	employee	95560899	Идэвхтэй	3/25/2025	5/7/2025	...
amgalan.dbb@gmail.com	e.emptest	emptest	95560899	Идэвхтэй	5/7/2025	5/7/2025	...

Зураг 3.3: Ажилтны жагсаалт

33

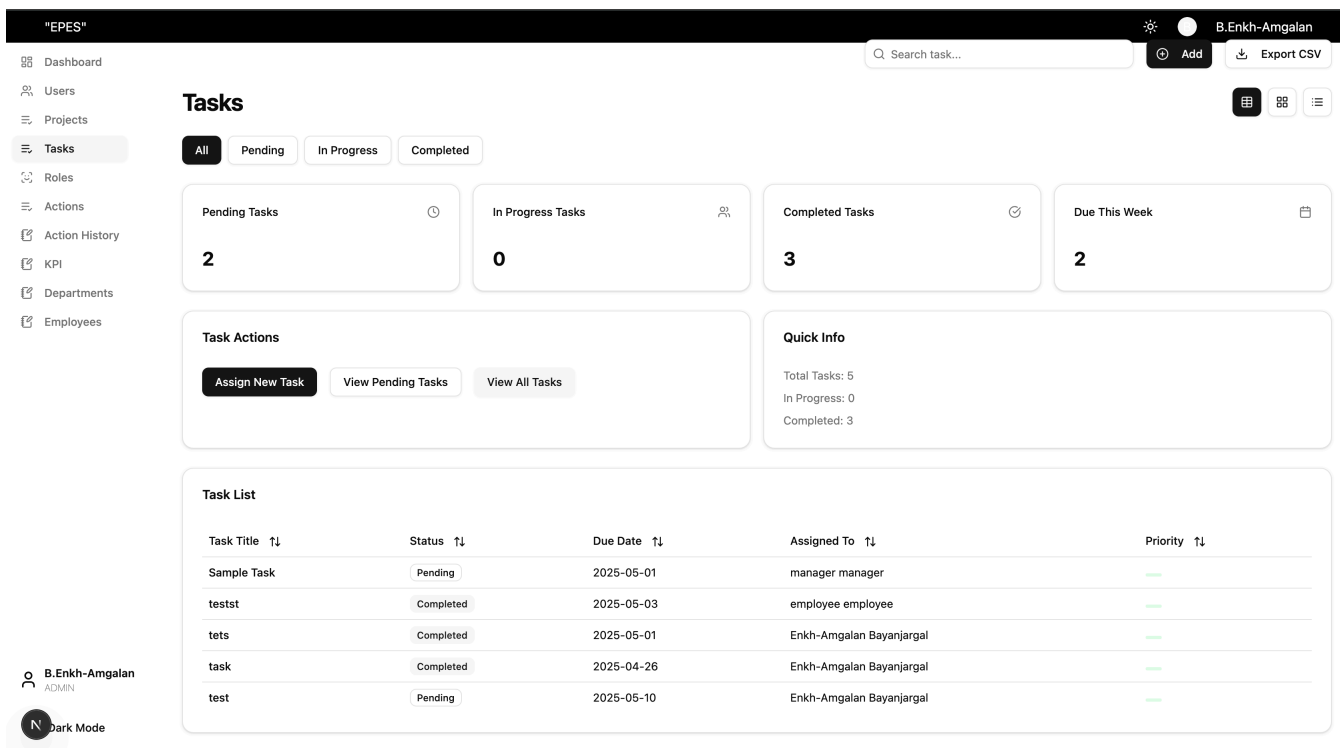
### 3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.4: Төслийн харагдац

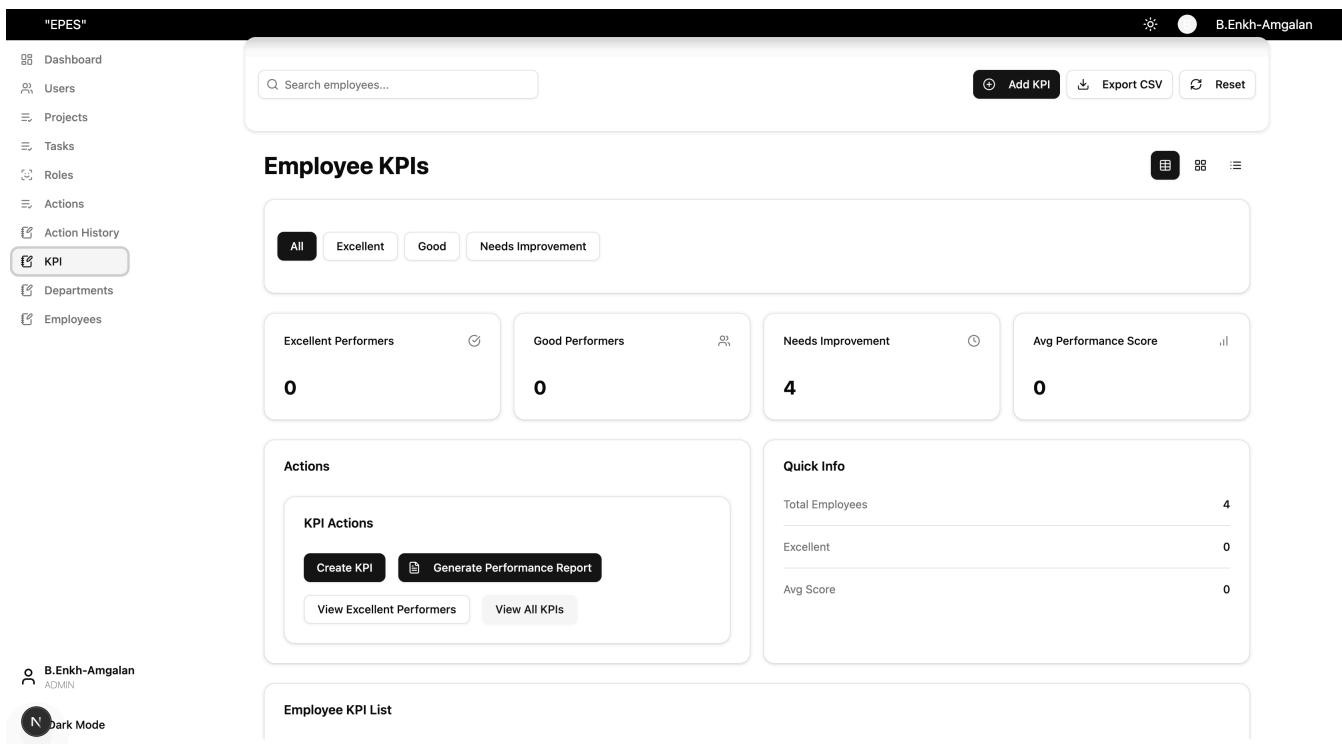
### 3.2. СИСТЕМИЙН ШААРДЛАГА

### БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.5: Даалгаварын харагдац

### 3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.6: Гүйцэтгэлийн үнэлгээний харагдац

×

### Generate Performance Report

Configure the performance evaluation report settings.

Employee

All Employees ▾

Time Period

All Time ▾

☒ Include Task Details

☒ Include Project Details

☐ Employee Performance Comments

Cancel

Generate PDF Report

Зураг 3.7: Тайлан гаргах компонентийн харагдац

### 3.2. СИСТЕМИЙН ШААРДЛАГА 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

"EPES"

⚙️

Dashboard

👤

Users

📁

Projects

📅

Tasks

👤

Roles

📄

Actions

📄

Action History

📄

KPI

📄

Departments

📄

Employees

⚙️

B.Enkh-Amgalan

ADMIN

🌙

Dark Mode

←

Back

User Profile

Personal Information

First Name

Enkh-Amgalan

Last Name

Bayanjargal

Login ID

test

Contact Information

Work Email

enkhmgalan.b@ondo.mn

Personal Email

amgalan.dbb@gmail.com

Work Phone

60096966

Personal Phone

88087316

Status

Active Status

Active

Work Duration

11/11/2024 - 11/11/2025

Position

Role

Role not found

Department

Department not found

Edit

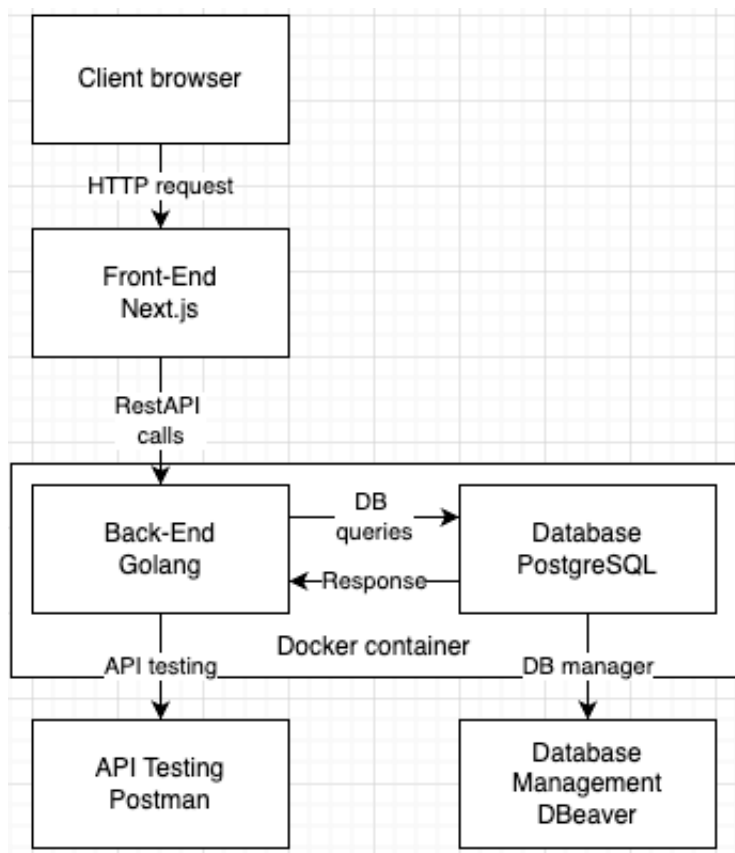
Зураг 3.8: Хэрэглэгчийн профайлын харагдац

38

### 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

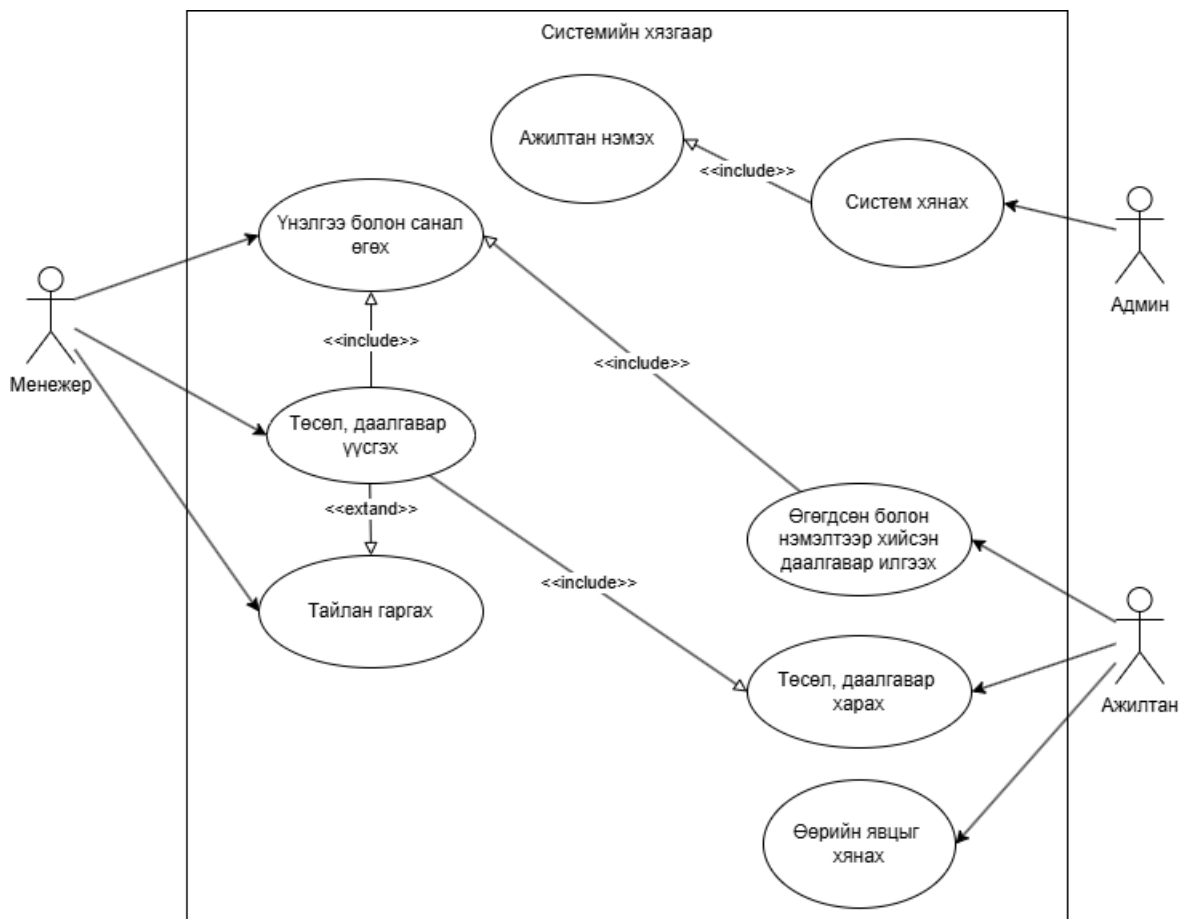
#### 3.3.2 Системийн зохиомж

##### 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

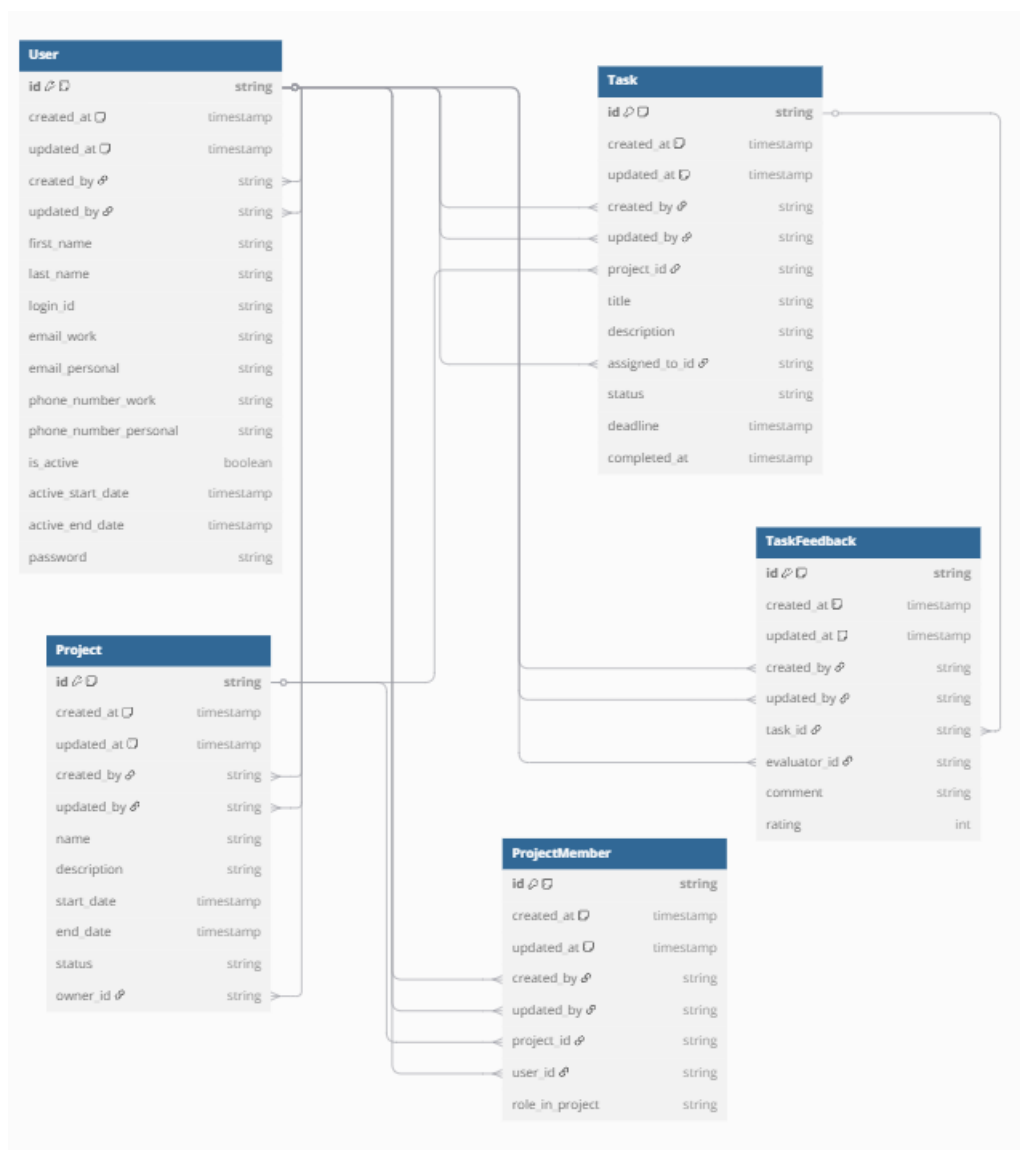
#### Системийн нэгж хоорондын харилцаа хамаарлын диаграмм

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

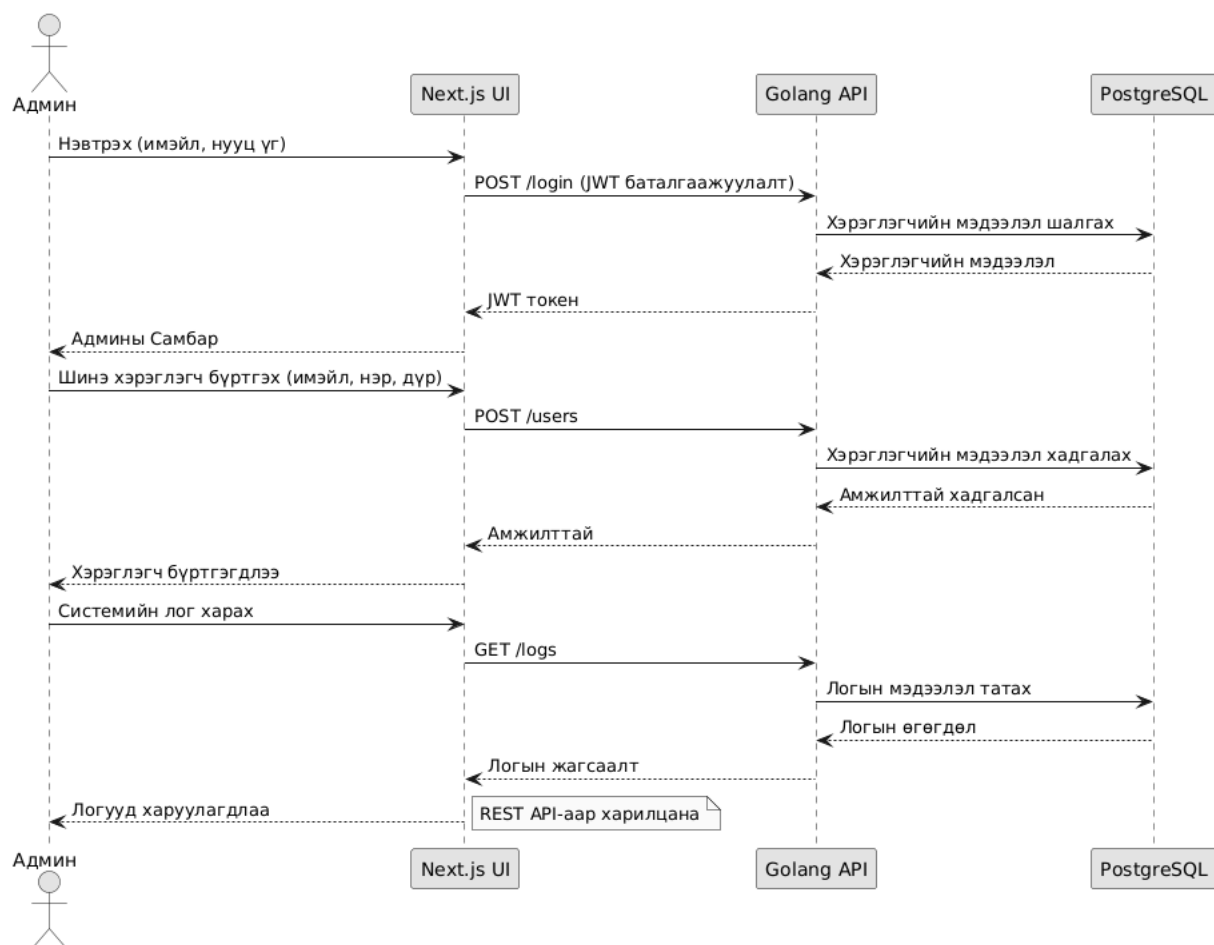


Зураг 3.13: Ажилтан болон түүний эрхийн харилцан хамаарлын диаграмм

### Дарааллын диаграмм

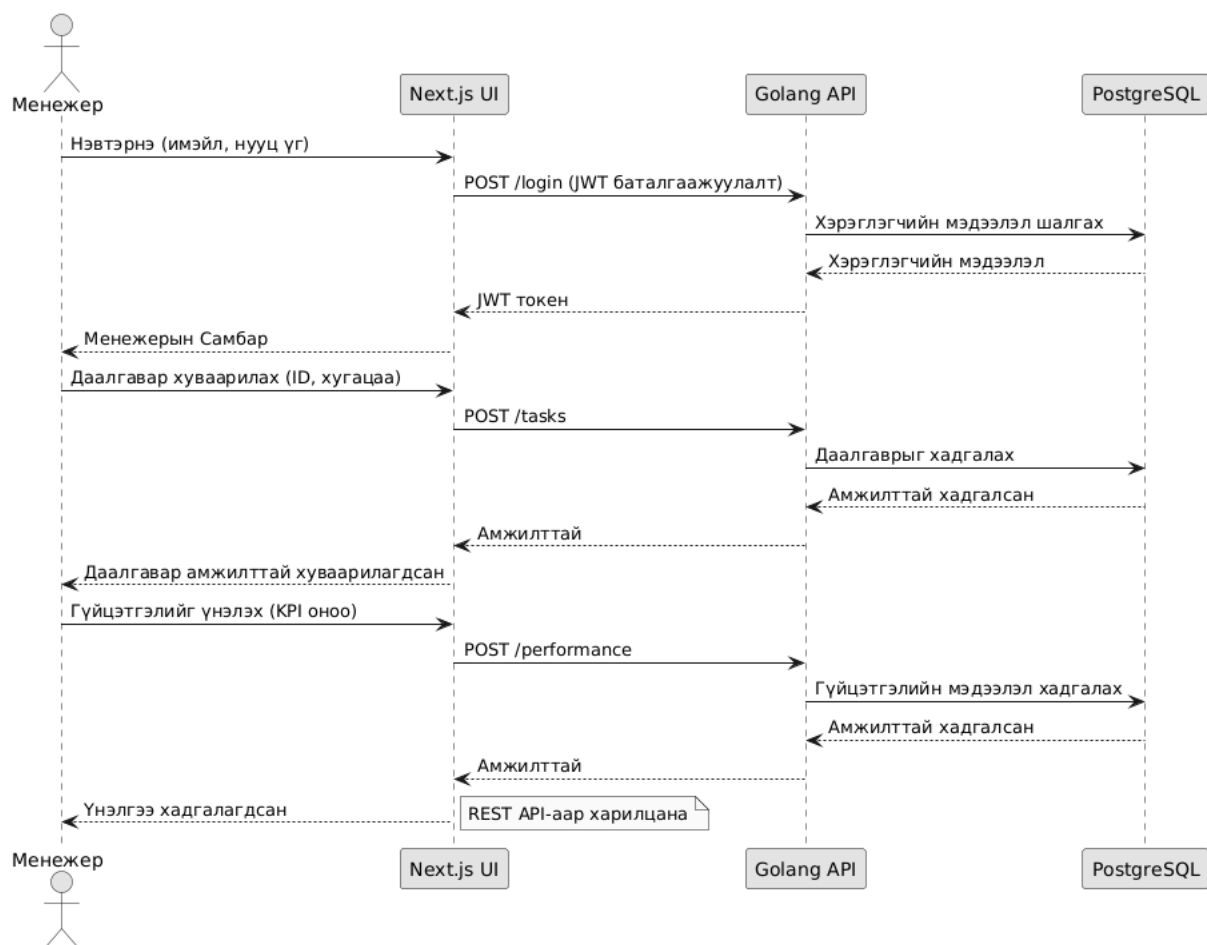
Дарааллын диаграмм нь системийн бүрэлдэхүүн хэсгүүдийн хоорондын харилцан үйлчлэлийг цаг хугацааны дарааллаар харуулдаг UML-ийн нэг төрлийн диаграмм юм. Энэ нь тодорхой үйлдэл, жишээ нь даалгаврын илгээлт гэх мэт процессын явцад объектуудын хоорондох мессежийн урсгалыг тодорхойлдог.

### 3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ

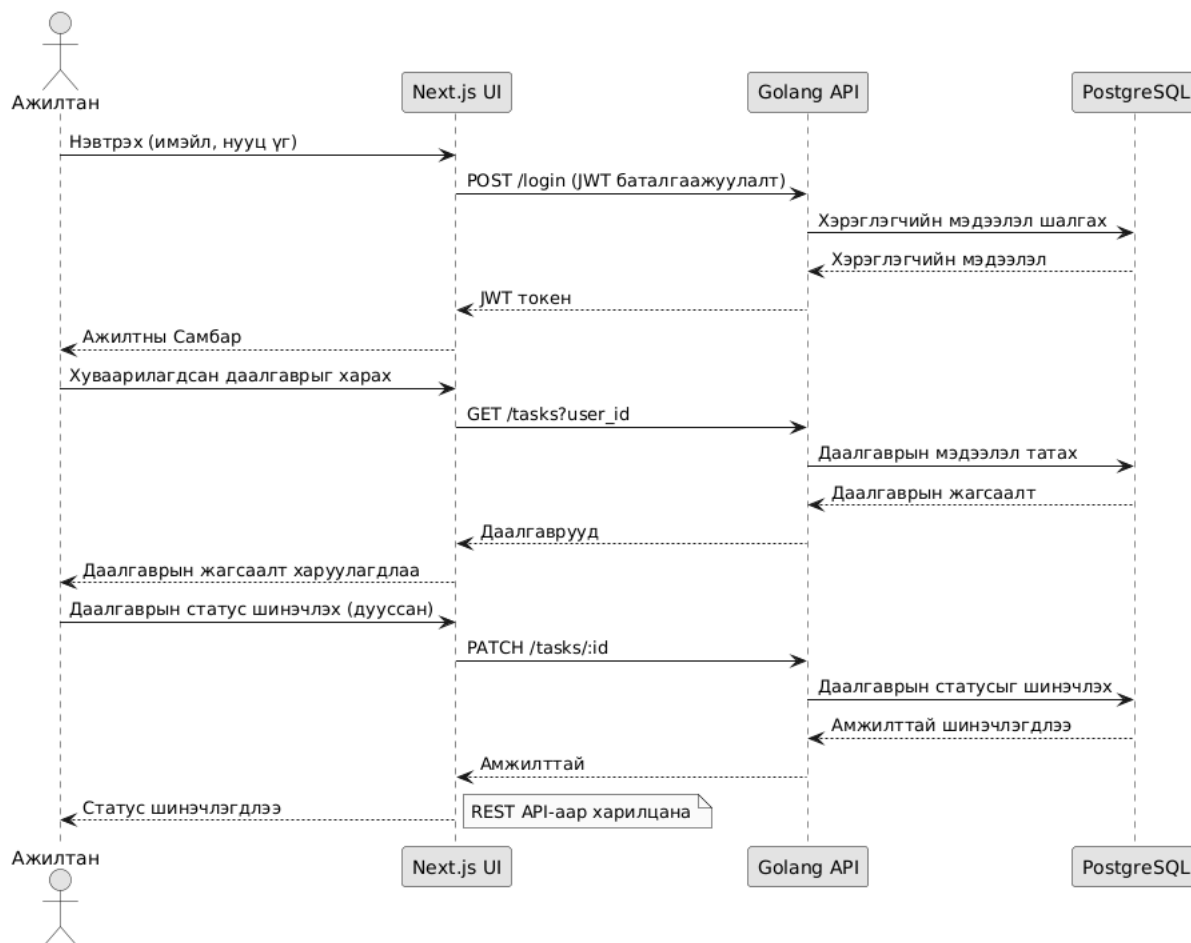


Зураг 3.14: Админ дарааллын диаграмм

### 3.3. СИСТЕМИЙН ЗАГВАР БҮЛЭГ 3. СИСТЕМИЙН ШИНЖИЛГЭЭ, ЗОХИОМЖ



Зураг 3.15: Менежер дарааллын диаграмм



Зураг 3.16: Ажилтан дарааллын диаграмм

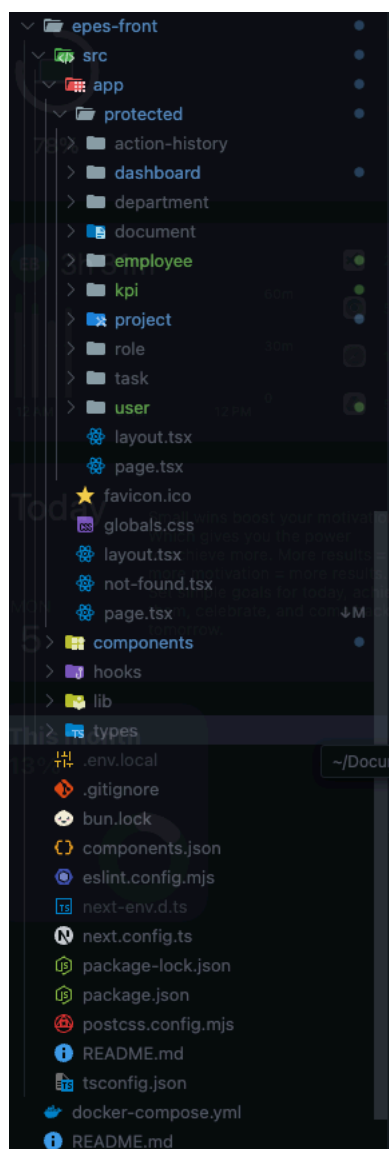
### 3.4 Бүлгийн дүгнэлт

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

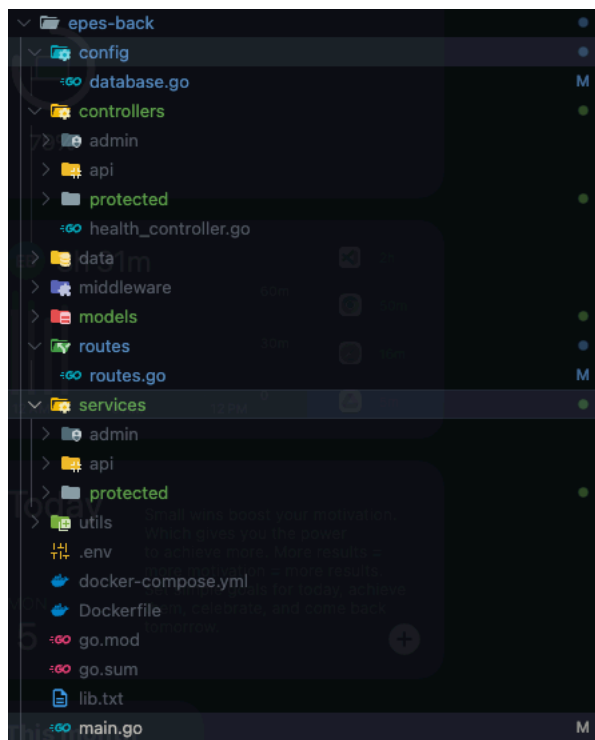
### 4.1 Файлын бүтэц

#### 4.1.1 *Front-end*



Зураг 4.1: Front-end файлын бүтэц

## 4.1.2 Back-end



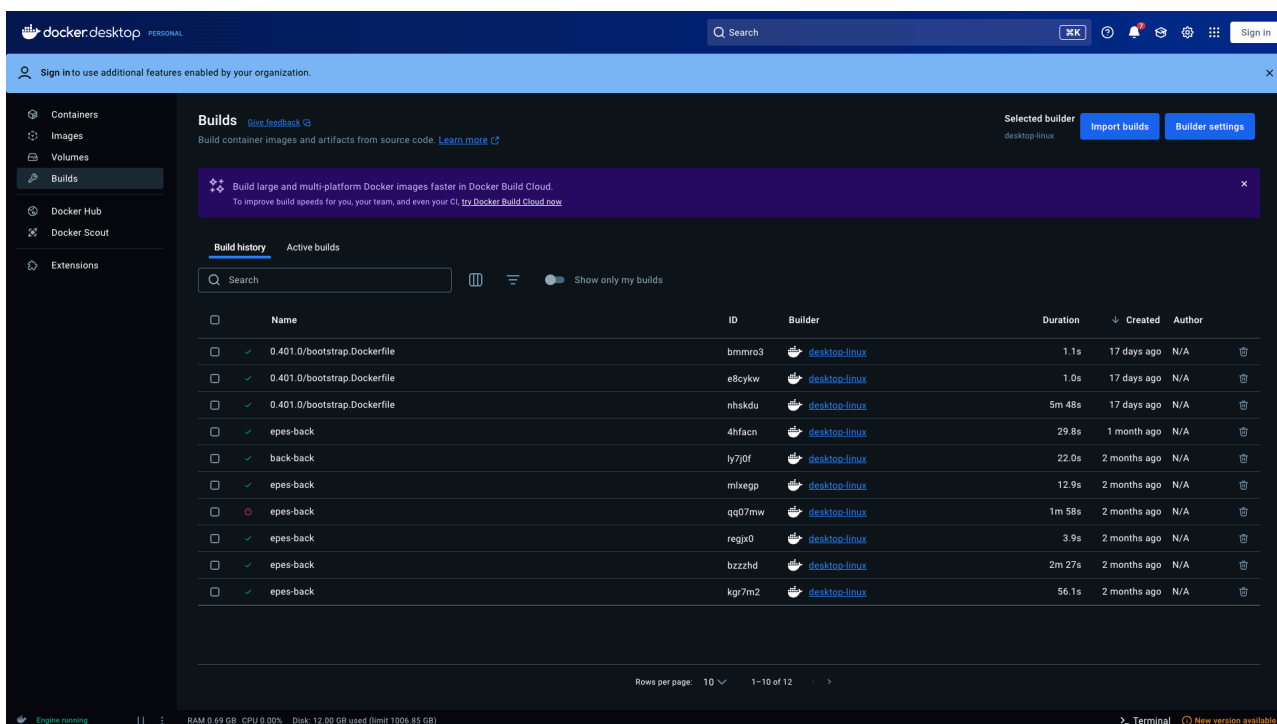
Зураг 4.2: Back-end файлын бүтэц



## 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.

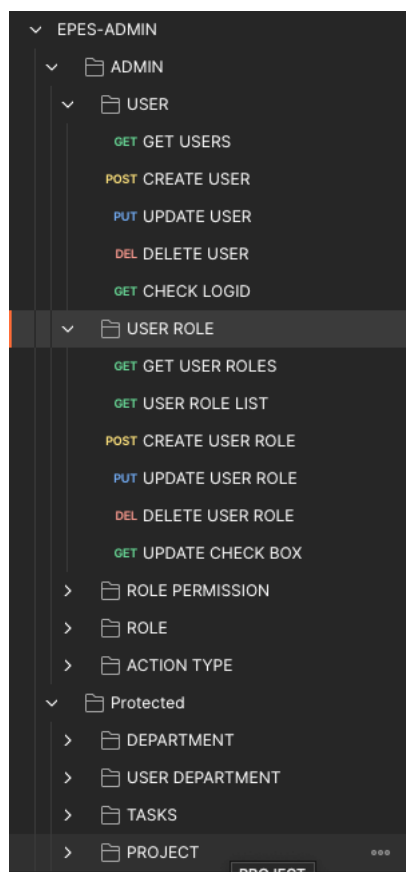
```
1 brew install go
2 brew install gin
```

Код 4.6: Golang суулгах

Хэрэгцээт технологиудаа суулгасны дараа шаардлагад тодорхойлсон архитектурын дагуу код бичнэ.

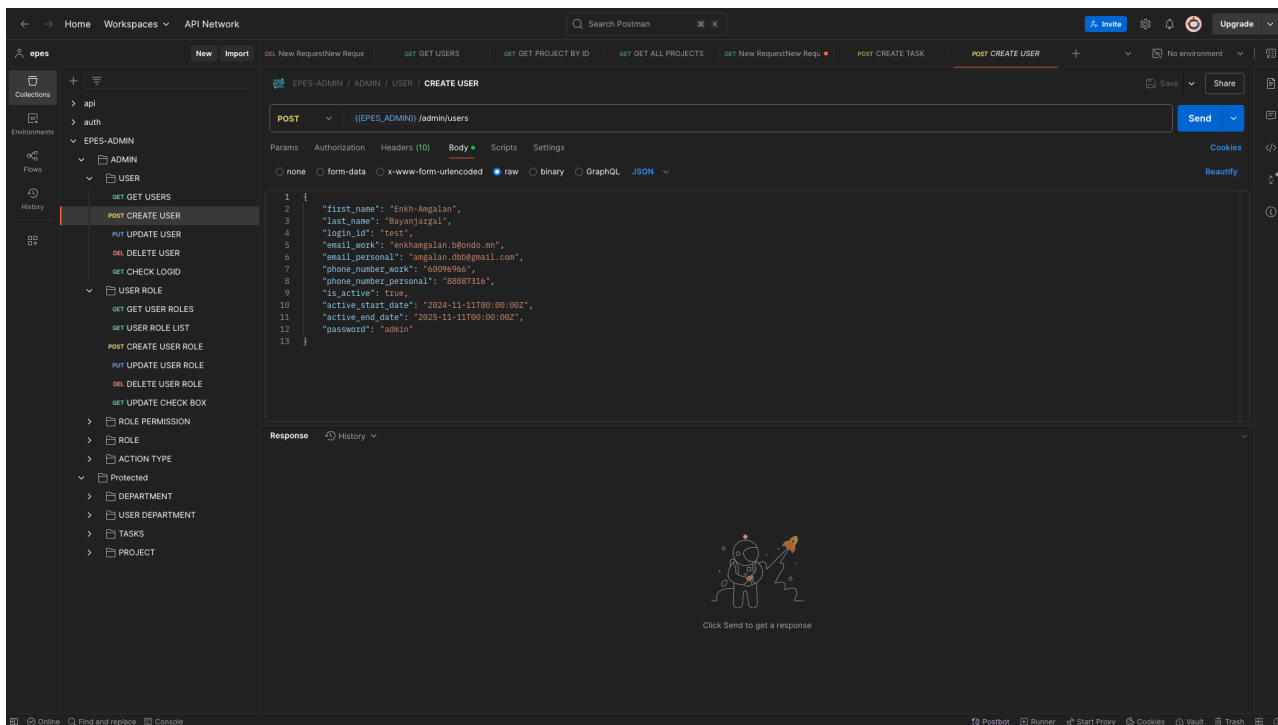
#### 4.2.5 *Test* орчин бүрдүүлэлт

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 Хэрэгжүүлэлт

Энэ хэсэгт ажилчны гүйцэтгэлийн үнэлгээний систем (EPES)-ийн хэрэгжүүлэлтийн үйл явцыг дэлгэрэнгүй тайлбарлаж, хөгжүүлэлтийн үе шатууд, ашигласан технологиудын хэрэглээ, тулгарсан асуудлууд болон тэдгээрийн шийдлийг тодорхойлно. Хэрэгжүүлэлтийн кодын хэсгүүдийг Хавсралт-д тусгасан бөгөөд энэ хэсэгт кодын үйл ажиллагаа, хөгжүүлэлтийн үр дүнтэй хамаарлыг оновчтой харуулна.

### 4.3.1 Хэрэгжүүлэлтийн үе шатууд

EPES системийн хөгжүүлэлт дараах үндсэн үе шатуудаас бүрдсэн:

- **Шаардлагын шинжилгээ:** Бизнесийн байгууллагуудын гүйцэтгэлийн үнэлгээний хэрэгцээг

судалж, бодит цагийн хяналт, 360 хэмжээний санал хүсэлт, OKR болон KPI-д суурилсан модулиудыг тодорхойлсон.

- **Технологийн сонголт:** Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), PostgreSQL, Docker зэрэг технологиудыг сонгож, тэдгээрийн өндөр гүйцэтгэл, хямд байдал, бизнесийн орчинд тохирсон байдлыг харгалзсан.
- **Системийн загварчлал:** REST API-д суурилсан микро үйлчилгээний архитектурыг бий болгож, front-end болон back-end харилцан ажиллагааг төлөвлөсөн.
- **Хөгжүүлэлт ба туршилт:** Системийн модулиудыг хөгжүүлж, Postman болон DBever ашиглан API болон өгөгдлийн сангийн гүйцэтгэлийг туршсан.
- **Нэвтрүүлэлт:** Docker-ийн тусламжтайгаар системийг тогтвортой орчинд нэвтрүүлсэн.

#### 4.3.2 Кодчиллын хэсгүүдийн хамаарал

Хэрэгжүүлэлтийн кодчиллын гол хэсгүүдийг Хавсралт-д оруулсан бөгөөд эдгээр нь системийн үндсэн модулиудын үйл ажиллагааг хангахад шууд хамааралтай юм:

- **REST API (Golang, Gin):** Хэрэглэгчийн баталгаажуулалт, гүйцэтгэлийн мэдээллийн асуулга, санал хүсэлтийн боловсруулалтыг хариуцдаг. Жишээлбэл, Хавсралт А-д оруулсан API endpoint нь KPI мэдээллийг бодит цагт авах боломжийг олгож, системийн хурд, найдвартай байдлыг хангана.
- **Өгөгдлийн сангийн удирдлага (GORM, PostgreSQL):** Гүйцэтгэлийн өгөгдөл, санал хүсэлтийн түүхийг хадгалах, хурдан хайлт хийхэд зориулагдсан. Хавсралт Б-д оруулсан загварууд нь өгөгдлийн бүтцийг тодорхойлж, 360 хэмжээний санал хүсэлтийн нийлмэл өгөгдлийг удирдахад тусална.
- **Front-end (Next.js, Tailwind CSS):** Хэрэглэгчдэд ээлтэй интерфэйсээр гүйцэтгэлийн тайлан, OKR хяналтын самбарыг харуулдаг. Хавсралт В-д оруулсан код нь хариу үйлдэлтэй

KPI самбарыг бий болгож, байгууллагуудын брэндийн онцлогт тохируулах боломжийг олгоно.

Эдгээр кодын хэсгүүд нь EPES системийн бодит цагийн хяналт, хэрэглэгчийн туршлага, өгөгдлийн найдвартай удирдлагын шаардлагыг хангаж, хөгжүүлэлтийн үр дүнд гүйцэтгэлийн үнэлгээний автоматжуулсан, хямд, local хэрэглээнд тохирсон системийг бий болгоход хувь нэмэр оруулсан.

#### 4.3.3 Тулгарсан асуудлууд ба шийдэл

Хэрэгжүүлэлтийн явцад дараах асуудлууд тулгарч, тэдгээрийг доорх аргаар шийдвэрлэсэн:

- **Асуудал 1: API удаашрал:** Олон хэрэглэгчийн зэрэг хүсэлтийг боловсруулахад Gin-ийн API-ийн хариу өгөх хугацаа удааширсан.

**Шийдэл:** PostgreSQL-ийн индексжүүлэлчийг оновчтой болгож, Gin-ийн middleware-д cache хийх логик нэмсэн. Жишээлбэл, KPI тайлангийн асуулгад Redis cache-ийг нэвтрүүлж, хариу өгөх хугацааг 40%-аар бууруулсан.

- **Асуудал 2: Front-end хариу үйлдлийн удаашрал:** Том хэмжээний гүйцэтгэлийн өгөгдлийг харуулахад Next.js-ийн rendering удааширсан.

**Шийдэл:** Next.js-ийн Incremental Static Regeneration (ISR)-ийг ашиглан статик хуудсуудыг урьдчилан үүсгэж, Tailwind CSS-ийн utility классуудыг тодорхой болгосон. Энэ нь хуудасны ачааллын хугацааг 30%-аар хурдасгасан.

- **Асуудал 3: Docker контейнерын нөөцийн хэт хэрэглээ:** Хөгжүүлэлтийн орчинд Docker контейнеруудын RAM болон CPU-ийн хэрэглээ өндөр байсан.

**Шийдэл:** Docker Compose тохиргоог оновчтой болгож, контейнерын нөөцийн хязгаарлалтыг тогтоосон. Мөн PostgreSQL-ийн тохиргоонд хэт ачааллыг бууруулахын тулд connection pooling нэмсэн.



**4.3.4 Хэрэгжүүлэлтийн үр дүнгийн дүгнэлт**

EPES системийн хэрэгжүүлэлт амжилттай хийгдэж, бодит цагийн гүйцэтгэлийн хяналт, 360 хэмжээний санал хүсэлт, OKR болон KPI-д суурилсан үнэлгээний модулиудыг багтаасан вэбд суурилсан систем бий болсон. Golang болон Next.js-ийн хослол нь хурдан, хэрэглэгчдэд ээлтэй платформыг хангасан бол PostgreSQL болон Docker нь өгөгдлийн найдвартай удирдлага, нэвтрүүлэлтийн тогтвортой байдлыг баталгаажуулсан. Тулгарсан асуудлуудыг оновчтой шийдвэрлэснээр системийн гүйцэтгэл, хүртээмжийг сайжруулж, жижиг, дунд байгууллагуудад хямд, local хэрэглээнд тохирсон шийдэл санал болгох боломжтой болсон. Гэсэн хэдий ч системийн том хэмжээний туршилт, аналитикийн модулийн өргөтгөл зэрэг нь цаашдын хөгжүүлэлтийн чухал чиглэл болно.

## 5. ДҮГНЭЛТ

### 5.1 Дүгнэлт

Энэхүү дипломын ажил нь бизнесийн байгууллагын ажилчны гүйцэтгэлийн үнэлгээг автоматжуулсан вэбд суурилсан систем (Employee Performance Evaluation System, EPES)-ийг хөгжүүлэх зорилготой байв. Судалгааны явцад гүйцэтгэлийн үнэлгээний орчин үеийн аргууд болох 360 хэмжээний санал хүсэлт, OKR (Objectives and Key Results), KPI (Key Performance Indicators)-ийг нарийвчлан судалж, эдгээрийг бизнесийн орчны онцлогт нийцүүлэн нэгтгэсэн цогц платформ бүтээсэн. Энэхүү ажил нь бизнесийн байгууллагуудын гүйцэтгэлийн удирдлагын процессыг автоматжуулж, ил тод байдал, үр ашгийг нэмэгдүүлэхэд чиглэсэн бөгөөд судалгааны үр дүнг доор нэгтгэн дүгнэв.

Судалгааны гол зорилго нь жижиг, дунд байгууллагуудад хямд, хэрэглэхэд хялбар, локал хэрэглээнд тохирсон гүйцэтгэлийн үнэлгээний системийг бий болгох байсан. EPES системийг **Golang (Gin, GORM, JWT), Next.js (Tailwind CSS), PostgreSQL, Docker** зэрэг нээлттэй эхийн технологиудыг ашиглан хөгжүүлсэн бөгөөд Админ, Менежер, Ажилтан гэсэн хэрэглэгчийн гурван эрхийн түвшинг дэмждэг. Систем нь жинлэсэн дундаж аргыг ашиглан гүйцэтгэлийн оноог тооцоолж, PDF болон CSV форматаар тайлан гаргах, даалгаврын удирдлага, бодит цагийн хяналтын модулиудыг багтаасан. Хэрэгжүүлэлтийн үр дүнд систем нь хэрэглэгчдэд ээлтэй интерфэйс, хурдан хариу өгөх хугацаа, өгөгдлийн бүрэн байдлыг хангасан бөгөөд тестийн үр дүнгээс харахад функциональ болон функциональ бус шаардлагыг бүрэн хангасан.

Судалгааны хүрээнд **Lattice** болон **BambooHR** зэрэг олон улсын гүйцэтгэлийн удирдлагын системүүдийг харьцуулан шинжилсэн. Харьцуулалтаас харахад EPES нь Lattice-ийн нарийвчилсан аналитик, 360 хэмжээний санал хүсэлтийн модулиудтай харьцуулахад илүү хямд, тохируулгын уян хатан байдлаараа онцлог бөгөөд BambooHR-ийн хялбаршуулсан интерфэйстэй харьцуулахад бодит цагийн хяналт, REST API-д суурилсан интеграцын боломжоороо давуу талтай. Нээлттэй

эхийн технологиудыг ашигласан нь лицензийн зардлыг хэмнэж, бизнесийн байгууллагуудын брэндийн онцлог, шаардлагад тохируулах боломжийг олгосон. Жишээлбэл, жинлэсэн дундаж аргыг ашигласан гүйцэтгэлийн тооцоолол нь үнэлгээний ил тод, шударга байдлыг хангаж, уламжлалт цаасан суурьтай үнэлгээний асуудлуудыг шийдвэрлэсэн.

Хөгжүүлэлтийн явцад тулгарсан гол сорилтуудыг, тухайлбал, API-ийн гүйцэтгэлийн удаашрал, фронтендийн рендерингийн хугацаа, 360 хэмжээний санал хүсэлтийн нийлмэл өгөгдлийн боловсруулалт, хэрэглээнд тохируулах бэрхшээлийг индексжүүлэлт, cache хийх, JSONB өгөгдлийн төрөл, хялбаршуулсан интерфэйсийн шийдлээр амжилттай даван гарсан. Эдгээр шийдлүүд нь системийн гүйцэтгэлийг 30-40%-иар сайжруулж, хэрэглэгчдийн хүлээн авах чадварыг нэмэгдүүлсэн. Хэрэгжүүлсэн кодын хэсгүүд (Хавсралт-ыг үзнэ үү) нь системийн үндсэн модулиудын үйл ажиллагааг хангаж, хөгжүүлэлтийн зорилгод хүрэхэд шууд хувь нэмэр оруулсан.

Судалгааны үр дүнд EPES систем нь жижиг, дунд байгууллагуудад хямд, хялбар хэрэглэгдэхүйц, гүйцэтгэлийн үнэлгээний автоматжуулсан шийдэл санал болгох боломжтойг харуулсан. Систем нь ажилчдын бүтээмж, сэтгэл ханамжийг дээшлүүлэх, байгууллагын удирдлагын процессыг ил тод болгоход хувь нэмэр оруулна. Мөн бизнесийн орчинд технологийн дэд бүтэц хязгаарлагдмал байдгийг харгалзан, хөнгөн, тогтвортой, нэвтрүүлэхэд хялбар шийдэл боловсруулсан нь практик ач холбогдолтой юм.

Ирээдүйн хөгжлийн чиглэлээр дараах боломжуудыг онцолж болно:

- **Хиймэл оюун ухааны интеграци:** Гүйцэтгэлийн өгөгдөлд суурилсан урьдчилсан таамаглах загваруудыг нэвтрүүлэх, жишээлбэл, ажилчдын гүйцэтгэлийн чиг хандлагыг урьдчилан таамаглах, автоматжуулсан санал болгох функцуудыг нэмэх.
- **360 хэмжээний санал хүсэлтийн өргөтгөл:** Илүү нарийвчилсан үнэлгээний загваруудыг (жишээ нь, хагас автоматжуулсан санал хүсэлтийн загвар) хөгжүүлж, хэрэглэгчдийн оролцоог нэмэгдүүлэх.
- **Мобайл апп хувилбар:** iOS болон Android платформд зориулсан мобайл апп хөгжүүлж, хэрэглэгчдийн хүртээмжийг өргөжүүлэх, ялангуяа алсын зайнаас ажиллах ажилчдад

тохиромжтой болгох.

- **Гуравдагч талын интеграци:** Slack, Microsoft Teams зэрэг платформтой шууд холбогдох API хөгжүүлж, системийн хэрэглээний уян хатан байдлыг нэмэгдүүлэх.

Эцэст нь, энэхүү судалгаа нь бизнесийн орчинд гүйцэтгэлийн үнэлгээний технологийн шийдлийн хэрэгцээ, боломжийг тодорхойлж, практик хэрэглээний загвар боловсруулсан нь шинжлэх ухаан, технологийн хувьд ач холбогдолтой юм. EPES систем нь уламжлалт үнэлгээний асуудлуудыг шийдвэрлэж, бизнесийн байгууллагуудын өрсөлдөх чадварыг дээшлүүлэхэд хувь нэмэр оруулна. Цаашид системийг бодит байгууллагуудад нэвтрүүлж, хэрэглэгчийн санал хүсэлтэд суурилсан сайжруулалт хийх, хиймэл оюун ухаан, тасралтгүй санал хүсэлтийн загваруудыг нэвтрүүлэх нь уг шийдлийг улам өрсөлдөх чадвартай, цогц платформ болгоно 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. REST API-ИЙН ЖИШЭЭ КОД (GOLANG, GIN)

```
1 package main
2
3 import (
4     "github.com/gin-gonic/gin"
5     "net/http"
6 )
7
8 // KPI endpoint
9 func getKPI(c *gin.Context)point {
10     employeeID := c.Param("id")
11     var kpi []KPIData
12     // PostgreSQL - KPI
13     if err := db.Where("employee_id=?", employeeID).Find(&kpi).Error;
14         err != nil {
15         c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed_
16             to_fetch_KPI"})
17         return
18     }
19     c.JSON(http.StatusOK, kpi)
20 }
21
22 func main() {
23     router := gin.Default()
24     router.GET("/kpi/:id", getKPI)
25     router.Run(":8080")
26 }
```

Код A.1: Routes

## V. 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 protectedRoutes := router.Group("/protected")
71 {
72     departmentRoutes := protectedRoutes.Group("/departments")
73     {
74         departmentRoutes.GET("/", protected.GetAllDepartments)
75         departmentRoutes.POST("/", protected.CreateDepartment)
76         departmentRoutes.PUT("/:id", protected.UpdateDepartment)
77         departmentRoutes.DELETE("/:id", protected.DeleteDepartment)
78     }
79
80     userDepartmentRoutes := protectedRoutes.Group("/user/
81         departments")
82     {
83         userDepartmentRoutes.GET("/", protected.
84             GetAllUserDepartments)
85         userDepartmentRoutes.POST("/", protected.
86             CreateUserDepartment)
87         userDepartmentRoutes.PUT("/:id", protected.
88             UpdateUserDepartment)
89         userDepartmentRoutes.DELETE("/:id", protected.
90             DeleteUserDepartment)
91         userDepartmentRoutes.GET("/list", protected.
92             UserDepartmentHandler)
93         userDepartmentRoutes.PUT("/update", protected.
94             UpdateUserDepartmentHandler)
95     }
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 B.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
62     Owner         User `json:"owner" gorm:"foreignKey:OwnerID
63                 ;references:ID"`
64     // Relationship to team members
65     TeamMembers []ProjectMember `json:"team_members" gorm:"foreignKey:
66                 ProjectID"`
67 }
68
69 type ProjectMember struct {
70     Model
71     ProjectID string `json:"project_id"`
72     Project    Project `json:"project" gorm:"foreignKey:ProjectID;
73                 references:ID"`
74     UserID     string `json:"user_id"`
75     User       User `json:"user" gorm:"foreignKey:UserID;
76                 references:ID"`
77     RoleInProject string `json:"role_in_project" // Optional: "
78                 Manager", "Developer", "QA", etc.
79 }
80
81 type EmployeeEvaluationReport struct {
82     Model
83     EmployeeID string `json:"employee_id"`
84     Employee    Employee `gorm:"foreignKey:EmployeeID"`
85     Period      string `json:"period"` // e.g., "Q1 2025"
86     ReportData  string `json:"report_data"` // JSON or HTML summary
87     GeneratedBy string `json:"generated_by"`
88     GeneratedAt time.Time `json:"generated_at"`
89 }

```

```

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 B.2: Models

## C. FRONT-END ЖИШЭЭ КОД (NEXT.JS)

```
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 "@/
11         components/ui/card";
12     import { Separator } from "@/components/ui/separator";
13     import {
14         BarChart,
15         CheckCircle,
16         CirclePlus,
17         Clock,
18         Download,
19         LayoutGrid,
20         List,
21         Table as TableIcon,
22         Users,
23         RefreshCw,
24     } from "lucide-react";
25     import { Search } from "lucide-react";
26     import { KPIStats } from "../components/KPIStats";
27     import { KPIActions } from "../components/KPIActions";
28     import { KPITable } from "../components/KPITable";
29     import { KPIGrid } from "../components/KPIGrid";
30     import { KPIList } from "../components/KPIList";
31     import { KPIReportDialog } from "../components/KPIReportDialog";
32     import { DeleteDialog } from "../components/DeleteDialog";
33     import { fetchEmployeeKPIs, deleteKPI } from "../services/kpiService";
34
35     import {
36         sortKPIs,
37         filterKPIs,
38         exportToCSV,
39         generatePerformanceReport,
40     } from "../utils/kpiUtils";
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;
```

```

47     return (...args: Parameters<F>) => {
48         clearTimeout(timeout);
49         timeout = setTimeout(() => func(...args), wait);
50     };
51 };
52
53 const KPIPage: React.FC = () => {
54     const { data: session, status } = useSession();
55     const router = useRouter();
56     const { toast } = useToast();
57     const [isLoading, setIsLoading] = useState(true);
58     const [viewMode, setViewMode] = useState<"table" | "grid" | "list"
59         ">("table");
60     const [searchTerm, setSearchTerm] = useState("");
61     const [searchInput, setSearchInput] = useState("");
62     const [kpis, setKPIs] = useState<EmployeeKPI[]>([]);
63     const [stats, setStats] = useState<KPIStat[]>([]);
64     const [sortField, setSortField] = useState<keyof EmployeeKPI |
65         null>(null);
66     const [sortDirection, setSortDirection] = useState<"asc" | "desc"
67         >("asc");
68     const [filterStatus, setFilterStatus] = useState<
69         "All" | "Excellent" | "Good" | "NeedsImprovement"
70         >("All");
71     const [deleteDialogOpen, setDeleteDialogOpen] = useState(false);
72     const [employeeToDelete, setEmployeeToDelete] = useState<string |
73         null>(null);
74     const [reportDialogOpen, setReportDialogOpen] = useState(false);
75     const [reportConfig, setReportConfig] = useState<ReportConfig>({
76         employeeId: "all",
77         period: "allTime",
78         includeTasks: true,
79         includeProjects: true,
80         includeComments: false,
81     });
82
83     // Centralized KPI metrics calculation
84     const calculateKPIMetrics = useCallback((kpiData: EmployeeKPI[])
85         => {
86         const excellentPerformers = kpiData.filter(
87             (k) => k.status === "Excellent"
88         ).length;
89         const goodPerformers = kpiData.filter((k) => k.status === "Good"
90             ).length;
91         const needsImprovement = kpiData.filter(
92             (k) => k.status === "NeedsImprovement"
93         ).length;
94         const avgPerformanceScore =
95             kpiData.length > 0
96                 ? Math.round(
97                     kpiData.reduce((sum, k) => sum + k.performanceScore, 0)
98                     /

```

```

92         kpiData.length
93     )
94     : 0;
95     const totalEmployees = kpiData.length;
96
97     return {
98         stats: [
99             {
100                 title: "Excellent Performers",
101                 value: excellentPerformers,
102                 icon: CheckCircle,
103             },
104             { title: "Good Performers", value: goodPerformers, icon:
105                 Users },
106             { title: "Needs Improvement", value: needsImprovement, icon
107                 : Clock },
108             {
109                 title: "Avg Performance Score",
110                 value: avgPerformanceScore,
111                 icon: BarChart,
112             },
113         ],
114         metrics: {
115             totalEmployees,
116             excellentPerformers,
117             avgPerformanceScore,
118         },
119     };
120 }, []);
121
122 // Debounced search handler
123 const debouncedSearch = useMemo(
124     () =>
125     debounce((value: string) => {
126         setSearchTerm(value);
127     }, 300),
128     []
129 );
130
131 // Handle search input change
132 const handleSearchChange = useCallback(
133     (e: React.ChangeEvent<HTMLInputElement>) => {
134         const value = e.target.value;
135         setSearchInput(value);
136         debouncedSearch(value);
137     },
138     [debouncedSearch]
139 );
140
141 // Reset all filters
142 const resetFilters = useCallback(() => {
143     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("/auth/signin");
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: CheckCircle },
182           { title: "Good Performers", value: 0, icon: Users },
183           { title: "Needs Improvement", value: 0, icon: Clock },
184           { title: "Avg Performance Score", value: 0, icon: BarChart },
185         ]);
186       } finally {
187         setIsLoading(false);
188       }
189     };

```



```

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

```

```

241         title: "Success",
242         description: "KPI_record_deleted_successfully.",
243     });
244     } catch (error: any) {
245         console.error("Failed to delete KPI:", error);
246         toast({
247             title: "Error",
248             description: "Failed to delete KPI:" + error.message,
249             variant: "destructive",
250         });
251     } finally {
252         setDeleteDialogOpen(false);
253         setEmployeeToDelete(null);
254     }
255 }, [employeeToDelete, session, toast, kpis, calculateKPIMetrics])
    ;

256
257     const sortedKPIs = useMemo(
258         () => sortKPIs(kpis, sortField, sortDirection),
259         [kpis, sortField, sortDirection]
260     );
261     const filteredKPIs = useMemo(
262         () => filterKPIs(sortedKPIs, searchTerm, filterStatus),
263         [sortedKPIs, searchTerm, filterStatus]
264     );
265
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("/auth/signin");
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       </* Search and Actions */>
304       <Card className="sticky top-0 z-30 border-b border-border
305         /50 bg-background/95 backdrop-blur supports-[backdrop-
306         filter]:bg-background/60">
307         <CardContent className="flex flex-col sm:flex-row items-
308           center gap-4 p-4">
309           <div className="relative w-full sm:w-auto flex-1">
310             <Search className="absolute left-2.5 top-2.5 h-4 w-4
311               text-muted-foreground" />
312             <Input
313               type="search"
314               value={searchInput}
315               onChange={handleSearchChange}
316               placeholder="Search employees..."
317               className="pl-8 w-full sm:w-[250px] lg:w-[350px]"
318             />
319           </div>
320           <div className="flex items-center gap-2 w-full sm:w-
321             auto justify-between sm:justify-end">
322             <Button onClick={() => router.push("/protected/kpi/
323               create")}>
324               <CirclePlus className="mr-2 h-4 w-4" />
325               Add KPI
326             </Button>
327             <Button
328               variant="outline"
329               onClick={() => exportToCSV(filteredKPIs)}
330               disabled={isLoading}>
331               >
332               <Download className="mr-2 h-4 w-4" />
333               Export CSV
334             </Button>
335             <Button variant="outline" onClick={resetFilters}>
336               <RefreshCw className="mr-2 h-4 w-4" />
337               Reset
338             </Button>
339           </div>
340         </CardContent>
341       </Card>

```

```

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

```

```

384         onClick={() => setFilterStatus("Good")}
385     >
386         Good
387     </Button>
388     <Button
389         variant={
390             filterStatus === "NeedsImprovement" ? "default"
391               : "outline"
392         }
393         onClick={() => setFilterStatus("NeedsImprovement")}
394     >
395         Needs Improvement
396     </Button>
397 </CardContent>
398 </Card>
399 <KPIStats stats={stats} isLoading={isLoading} />
400
401 <div className="grid_gap-4_md:grid-cols-2_mb-6">
402     <Card>
403         <CardHeader>
404             <CardTitle>Actions</CardTitle>
405         </CardHeader>
406         <CardContent>
407             <KPIActions
408                 isLoading={isLoading}
409                 onCreate={() => router.push("/protected/kpi/
410                   create")}
411                 onGenerateReport={() => setReportDialogOpen(true)}
412                 onViewExcellent={() =>
413                     router.push("/protected/kpi/excellent")
414                 }
415                 onViewAll={() => router.push("/protected/kpi/all")
416             }>
417         </CardContent>
418     </Card>
419     <Card>
420         <CardHeader>
421             <CardTitle>Quick Info</CardTitle>
422         </CardHeader>
423         <CardContent>
424             {isLoading ? (
425                 <Skeleton className="h-20_w-full" />
426             ) : (
427                 <div className="space-y-4">
428                     <div className="flex_justify-between_items-
429                       center">
430                         <span className="text-sm_text-muted-
431                           foreground">

```

```

429         Total Employees
430     </span>
431     <span className="text-sm font-medium">
432         {metrics.totalEmployees}
433     </span>
434 </div>
435 <Separator />
436 <div className="flex justify-between items-
437     center">
438     <span className="text-sm text-muted-
439         foreground">
440         Excellent
441     </span>
442     <span className="text-sm font-medium">
443         {metrics.excellentPerformers}
444     </span>
445 </div>
446 <Separator />
447 <div className="flex justify-between items-
448     center">
449     <span className="text-sm text-muted-
450         foreground">
451         Avg Score
452     </span>
453     <span className="text-sm font-medium">
454         {metrics.avgPerformanceScore}
455     </span>
456 </div>
457 </div>
458
459 </CardContent>
460 </Card>
461 </div>
462
463 <Card>
464     <CardHeader>
465         <CardTitle>Employee KPI List</CardTitle>
466     </CardHeader>
467     <CardContent>
468         {isLoading ? (
469             <div className="space-y-4">
470                 <Skeleton className="h-8 w-full" />
471                 <Skeleton className="h-32 w-full" />
472             </div>
473         ) : (
474             <>
475                 {viewMode === "table" && (
476                     <KPITable
477                         kpis={filteredKPIs}
478                         onSort={handleSort}
479                         onClick={handleKPIClick}
480                         onEdit={handleEditKPI}

```

```

477         onDelete={() => {
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={() => {
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={() => {
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 C.1: Routes