

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

ҮЙЛДВЭРИЙН ДАДЛАГЫН АЖЛЫН ТАЙЛАН

Үйлдвэрлэлийн дадлага
хийсэн байгууллагын нэр:

”Зочил Технологи” ХХК

Шалгасан:

Үйлдвэрлэлийн дадлага
удирдсан ажилтны нэр,
албан тушаал:

Т.Билгүүн, Фронт-энд хөгжүүлэгч

Гүйцэтгэсэн:

Програм хангамж
IV ангийн оюутан

Э.Жавхлан (20B1NUM0649)

Улаанбаатар хот

2024 он

Зохиогчийн баталгаа

Миний бие Энхбаярын Жавхлан үйлдвэрийн дадлагын ажлын тайланг хийж гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг тайлангийн ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

ГАРЧИГ

УДИРТГАЛ	1
1. БАЙГУУЛАГЫН ТАНИЛЦУУЛГА	3
1.1 Товч танилцуулга	3
1.2 Ямар үйлчилгээ үзүүлдэг вэ?	3
2. ИЖИЛ СИСТЕМИЙН СУДАЛГАА	4
2.1 Shopify	4
2.2 Saleor	5
3. СИСТЕМИЙН ШААРДЛАГА	6
3.1 Танилцуулга	6
3.2 Функционал шаардлагууд	6
3.3 Функционал бус шаардлагууд	7
3.4 Use-case диаграм	8
4. АШИГЛАХ ТЕХНОЛОГИУД	9
4.1 Git болон GitHub	9
4.2 Javascript	10
4.3 React	10
4.4 React Hook Form	11
4.5 Bootstrap	12
4.6 GitBook	12
5. ХЭРЭГЖҮҮЛЭЛТ	13
5.1 Динамик баннерийн CRUD форм	13
5.2 Арга хэмжээ үүсгэх CRUD форм	23
5.3 Үйлчилгээний төлбөрийн саналын хуудас	31
5.4 Мод бүтэцтэй бүтээгдэхүүний категори	36
5.5 Zochil платформын гарын авлагыг шинэчлэх	41

ГАРЧИГ

ГАРЧИГ

6. ДҮГНЭЛТ	43
6.1 Yр дүнгийн тайлан	43
НОМ ЗҮЙ	44
ХАВСРАЛТ	45
А. УДИРДАГЧИЙН ҮНЭЛГЭЭ ДҮГНЭЛТ	45

ЗУРГИЙН ЖАГСААЛТ

3.1	Use-case диаграм	8
5.1	Баннерийн жагсаалт хэсэг	22
5.2	Баннерийн форм хэсэг	22
5.3	Баннерийн жагсаалт хэсэг	29
5.4	Баннерийн форм хэсэг	30
5.5	Үйлчилгээний төлбөрийн саналын компонент	36
5.6	Категоруудыг нэмэх компонент	41
5.7	Категоруудыг нэмэх компонент	42
A.1	Удирдагчийн үнэлгээ дүгнэлт	45

ХҮСНЭГТИЙН ЖАГСААЛТ

1	Дадлагын ажлын төлөвлөгөө	2
---	---------------------------------	---

Кодын жагсаалт

4.1	JSX ашиглаж ”container” кластай html элемент буцаах компонент	11
5.1	Баннерийн CRUD хийх custom hook	13
5.2	Баннерийн форм validation	15
5.3	Баннерийн форм компонент	16
5.4	Баннерийн жагсаалт компонент	20
5.5	Арга хэмжээн CRUD хийх custom hook	23
5.6	Арга хэмжээн форм validation	24
5.7	Арга хэмжээн форм компонент	24
5.8	Арга хэмжээн жагсаалт компонент	28
5.9	Үйлчилгээний төлбөрийн саналын компонент	31
5.10	Категоруудыг авах custom hook	36
5.11	Категоруудыг нэмэх компонент	37
5.12	Категоруудыг нэмэх компонент	39

УДИРТГАЛ

Миний бие Энхбаярын Жавхлан үйлдвэрийн дадлагын ажлыг “Зочил технологи” ХХК компани дээр гүйцэтгэсэн. Энэхүү үйлдвэрийн дадлагын хүрээнд хичээлээс болон бие даан эзэмшсэн мэдлэг, чадвараа ашиглан бодит ажлын орчинд бүтээгдэхүүн хөгжүүлж бодит хэрэглэгчдийн гарпт хүргэх процесст суралцан Админ веб удирдлагын систем төслийн front-end хөгжүүлэлтэнд оролцсон.

Зорилго: React, Bootstrap технологиудын талаар судалж бүтээгдэхүүн хөгжүүлэх, компанийн хөгжүүлэлтийн арга барилтай танилцах.

Зорилт: Удирдагчийн зааварчилгааны дагуу алхам алхмаар судалгаа хийж өгсөн шаардлагын хүрээнд хэрэгжүүлэлт хийх. Хурал уулзалтуудад хамрагдаж, бодит бүтээгдэхүүн хөгжүүлэлтийн процесст оролцох. Back-end хөгжүүлэлтийн багтай багаар хамтран ажиллах.

Table 1: Дадлагын ажлын төлөвлөгөө

№	Гүйцэтгэх ажил	Хугацаа	Биелэлт	Дадлагын удирдагчийн үнэлгээ
1	Zochil платформын Admin веб болон гар утасны Zochil IO апп-г судлах	12/29 - 01/02		
2	Admin веб дээр бараа бүтээгдэхүүн болон захиалгуудын жагсаалтын responsive засах	01/02 - 01/04		
3	React-hook-form-г судлах	01/04 - 01/05		
4	Олон төрлийн динамик баннерийн CRUD форм хийх	01/05 - 01/07		
5	Арга хэмжээ үүсгэх CRUD форм хийх	01/07 - 01/09		
6	Zochil платформын үйлчилгээний төлбөрийн саналын хуудас хийх	01/09 - 01/11		
8	Мод бүтэцтэй бүтээгдэхүүний категори нэмэх компонент хийх	01/11 - 01/14		
9	Zochil платформын гарын авлагыг GitBook дээр шинэчлэн хийх	01/14 - 01/20		
11	Admin вебийн зарим хуудсын responsive засах	01/20- 01/29		

1. БАЙГУУЛАГЫН ТАНИЛЦУУЛГА

1.1 Товч танилцуулга

“Зочил Технологи” ХХК нь мэдээлэл технологийн чиглэлээр 2019 онд үүсгэн байгуулагдсан. Мэдээлэл технологи, борлуулалт, маркетингийн чиглэлээр 10 гаруй жилийн туршлагатай баг, хамт олны бүрэлдэхүүнтэй ажиллаж буй онлайн худалдааг хөгжүүлэх эрхэм зорилготой компани юм.

1.2 Ямар үйлчилгээ үзүүлдэг вэ?

Уг компани нь онлайн худалдааны бизнес эрхлэгчдэд зориулан илүү бага зардлаар, илүү хялбар технологийг ашиглаж, захиалга, хургэлт, худалдааг автоматжуулах үйлчилгээг онлайн худалдааны Zochil платформоор дамжуулан үзүүлдэг.

2. ИЖИЛ СИСТЕМИЙН СУДАЛГАА

Технологийн хөгжил болон ковид-19 цар тахлаас үүдэн дэлхийн хэмжээнд онлайн худалдааны сайтуудын борлуулалт, худалдан авагч болон онлайн дэлгүүрүүдийн тоо өндөр өсөлттэй байна. Үүнийгээ дагаад маш олон тооны интернэтээр бараа бүтээгдэхүүн борлуулдаг (e-commerce) төрлийн сайтууд үүссэн ба эдгээрээс манай бүтээгдэхүүнтэй ижил төстэй 2 системийг авч танилцууллаа.

2.1 Shopify

Shopify бол бизнес эрхлэгчдэд онлайн дэлгүүр үүсгэх, удирдах боломжийг олгодог цахим худалдааны платформ юм. Энэ нь хувь хүмүүс болон бизнесүүдэд техникийн өргөн мэдлэг шаардлагагүйгээр өөрсдийн онлайн дэлгүүрээ байгуулж, ажиллуулахад туслах олон төрлийн хэрэгсэл, үйлчилгээгээр хангадаг. Shopify-ийн гол онцлогууд нь:

- **Дэлгүүрийн дизайн:** Хэрэглэгчид дэлгүүрийнхээ өнгө үзэмжийг өөрийн брэндэд тохируулан өөрчлөх боломжтой.
- **Бүтээгдэхүүний менежмент:** Борлуулагчид бүтээгдэхүүнээ хялбархан нэмж, тайлбар, үнэ, бараа материалын түвшин зэрэг дэлгэрэнгүй мэдээллийг оруулах боломжтой.
- **Төлбөрийн боловсруулалт:** Shopify нь төлбөрийн олон шийдлийг дэмждэг тул үйлчлүүлэгчдэд төлбөрийн янз бүрийн аргыг ашиглан худалдан авалт хийхэд хялбар болгодог.
- **Аюулгүй байдал:** Shopify нь хостинг, төлбөр хийх, хэрэглэгчийн мэдээллийг хамгаалах зэрэг аюулгүй байдлын асуудлыг хариуцдаг.
- **Захиалгын менежмент:** Энэхүү платформ нь захиалгыг удирдах, бараа материалыг хянах, тээвэрлэлт, гүйцэтгэлийг зохицуулахад тусалдаг.

- **Аналитик:** Бизнесүүдэд туслах зорилгоор борлуулалт, хэрэглэгчийн зан төлөв болон бусад чухал хэмжүүрүүдийг хянах аналитик хэрэгслээр хангадаг.

2.2 Saleor

Saleor нь бизнес эрхлэгчдэд онлайн дэлгүүр байгуулах, удирдах боломжийг олгодог нээлттэй эхийн цахим худалдааны платформ юм. Уян хатан байдал, өргөтгөх боломжтой, ашиглахад хялбар болгох үүднээс эдгээр технологийг ашигладаг. Saleor нь өөрсдийн цахим худалдааны вэбсайтыг бий болгохыг хүсэж буй хөгжүүлэгчид болон бизнесүүдэд зориулсан.

Ашигласан технологийн хувьд яг ижил ба үүнд React, GraphQL, болон Django технологиудыг ашиглан хөгжүүлжээ. Мөн Merchant вебийнхээ интерфейс дээр Material-ui хэрэглэсэн байна. Хэрэглэгч буюу дэлгүүрийн админ өөрийн хүссэн үедээ шинээр дэлгүүр нээх боломжтой. Ингэснээр зөвхөн танай дэлгүүрт зориулагдсан e-commerce веб, дэлгүүрээ удирдах мерчант вебийг бэлдэн гаргаж өгөх болно. Манай Oneline төсөл дэлгүүр хэсгээ зөвхөн нэг гар утасны апп дээр шийдсэн бол уг систем веб болон PWA ашиглан дэлгүүр бүрд шинэ бүтээгдэхүүн өгдөг байдлаараа ялгаатай байна.

3. СИСТЕМИЙН ШААРДЛАГА

3.1 Танилцуулга

Миний дадлагын хугацаанд ажилсан систем нь бизнес эрхлэгчдэд өөрийн онлайн дэлгүүрийг удирдах Zochil платформын Админ веб удирдлагын систем юм. Энэхүү системээр бизнес эрхлэгчид өөрийн онлайн дэлгүүрийн бараа бүтээгдэхүүний каталог удирдах, гүйцэтгэл болон бүтээгдэхүүний захиалга хянах, олон нийтийн мэдээллийн хэрэгсэл, зах зээл болон бусад онлайн платформууд зэрэг янз бүрийн сувгаар бүтээгдэхүүн борлуулах болон нэмэлт үйлчилгээ авах боломжийг олгодог вебд сууриссан систем юм.

3.2 Функционал шаардлагууд

- Хэрэглэгчийн бүртгэл:** Вэбсайт нь хэрэглэгчдэд утасны дугаараа ашиглан бүртгүүлэх боломжтой байх.
- Хэрэглэгчийн нэвтрэлт:** Бүртгэгдсэн хэрэглэгчид бүртгүүлсэн утасны дугаар болон нууц үгээ ашиглан эсвэл нэг удаагийн нууц үгээр нэвтрэх боломжтой байх.
- Бүтээгдэхүүний каталогийн менежмент:** Бүтээгдэхүүний тодорхойлолт, зураг, үнэ зэргийг багтаасан онлайн бүтээгдэхүүний каталогийг үүсгэх, засварлах, удирддаг байх.
- Захиалгын менежмент:** Үйлчлүүлэгчийн захиалгыг харах, боловсруулах, биелүүлэх, захиалгын байдал, тээвэрлэлтийн мэдээллийг удирдах функциүүдтэй байх.
- Олон сувгийн борлуулалтын менежмент:** Олон нийтийн мэдээллийн хэрэгсэл, зах зээл болон бусад онлайн платформууд гэх мэт янз бүрийн сувгаар бүтээгдэхүүн борлуулах дэмжлэг үзүүлдэг байх.

3.3. ФУНКЦИОНАЛ БУС ШААРДЛАГУУД БҮЛЭГ 3. СИСТЕМИЙН ШААРДЛАГА

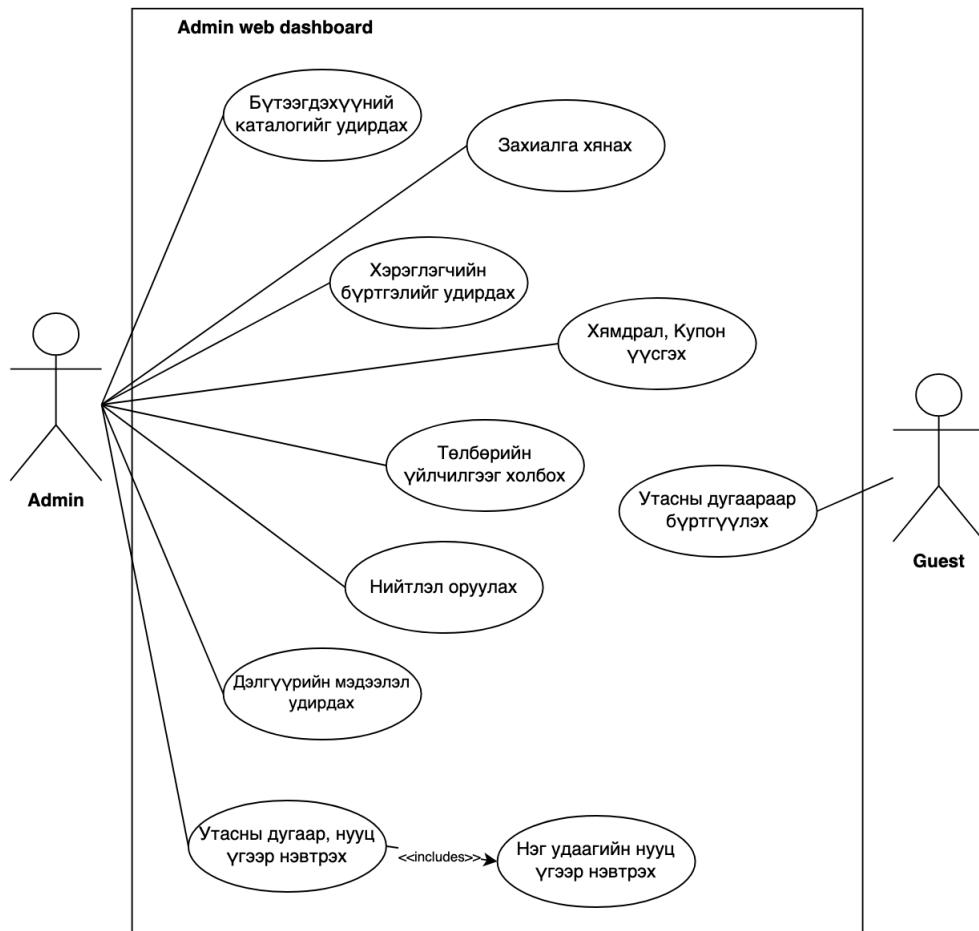
- **Төлбөрийн үйлчилгээний интеграци:** Онлайн гүйлгээг аюулгүй болгох үүднээс төлбөрийн үйлчилгээ үзүүлэгчтэй хамтран ажиллах боломжтой байх.
- **Маркетинг ба сурталчилгаа:** Маркетингийн кампанит ажил явуулах, хөнгөлөлт үзүүлэх, сурталчилгааны үйл ажиллагааг удирдах хэрэгслүүдтэй байх.
- **Аналитик:** Өгөгдөл тулгуурлан бизнесийн шийдвэр гаргахад дэмжлэг үзүүлэх борлуулалтын тайлан, гүйцэтгэлийн хэмжүүрийг харж боломжтой байх.
- **Хэрэглэгчийн удирдлага:** Админ хяналтын самбар доторх хэрэглэгчийн бүртгэл, үүрэг, зөвшөөрлийг удирдах функцтэй байх.
- **Мобайл болон веб хандалт:** Онлайн дэлгүүрийг удирдахын тулд гар утасны апликацийн болон вэб хяналтын самбараар дамжуулан ашиглах боломжтой байх.

3.3 Функционал бус шаардлагууд

- **Гүйцэтгэл:** Админ хяналтын самбарын үйлдлүүд болон хуудас ачаалах хооронд хамгийн бага хоцрогдолтой, хурдан бөгөөд хариу үйлдэл үзүүлэх ёстой.
- **Аюулгүй байдал:** Вэбсайт нь хэрэглэгчийн мэдээллийг хамгаалах, зөвшөөрөлгүй нэвтрэхээс урьдчилан сэргийлэх хатуу арга хэмжээ авах ёстой.
- **Өргөтгөх чадвар:** Вэбсайт нь бизнес өсөхийн хэрээр нэмэгдэж буй траффик болон өгөгдлийг зохицуулах чадвартай байх ёстой.
- **Ашиглах боломж:** Вэбсайт нь ашиглах, удирдахад хялбар байх бөгөөд хэрэглэгчдэд хэрэгтэй мэдээллээ хурдан олох боломжийг олгодог, ээлтэй интерфейстэй, ойлгомжтой зааварчилгаа, хялбар навигацитай байх ёстой.
- **Найдвартай байдал:** Вэбсайт нь найдвартай байх ёстой бөгөөд хамгийн бага сул зогсолт, бизнесийн үйл ажиллагаанд саад учруулахгүй байх ёстой.

- **Хүртээмжтэй байдал:** Вэбсайт нь өөр өөр хөтөч (Chrome, Firefox, Safari гэх мэт) болон төхөөрөмжүүдтэй (ширээний компьютер, гар утас, таблет) нийцтэй байх ёстай.
- **Maintainability:** Вэбсайт болон түүний үндсэн код нь засвар үйлчилгээ хийх, шинэчлэхэд хялбар байх ёстай.
- **Дагаж мөрдөх:** Вэбсайт нь өгөгдөл хамгаалах хууль, дүрэм журамд нийцсэн байх ёстай.

3.4 Use-case диаграм



Зураг 3.1: Use-case диаграм

4. АШИГЛАХ ТЕХНОЛОГИУД

4.1 Git болон GitHub

Linus Torvalds буюу Linux Kernel-г хөгжүүлсэн хүн Kernel-ийнхээ эх кодыг удирдах зорилгоор уг технологийг анх санаачилж, хэрэгжүүлсэн байдаг. Гол зорилго нь Version Control System буюу хувилбар удирдах системийг бүтээх ба ингэснээр хөгжүүлэлтийн явцад бүх өөрчлөлтөө хадгалах, багаар ажиллах боломжийг бүрдүүлж өгсөн юм.

Үндсэн ажиллагааг нь тайлбарлавал Git ашиглаж буй хөгжүүлэгч/хэрэглэгч бүр өөрийн төхөөрөмж дээр үндсэн repository-тай яг ижил repository-г үүсгэнэ. Ингэснээр сүлжээнд холбогдсон эсэхээс үл хамааран дурын өөрчлөлт, хөгжүүлэлтээ хийх боломжтой ба уг ажил дууссан үед бичсэн өөрчлөлтуүдээ commit хийж, remote repository руу push үйлдлийг хийнэ.

GitHub бол Git-г ашигладаг веб платформ юм. Бусадтай хамтарч ажиллах, төслөө хадгалах, проект менежмент, continuous integration, болоод бусдын төслийг харах, судлах, хуулбарлах гэх мэт маш олон үйлчилгээг үзүүлдэг. Гол онцлогуудаас дурдвал,

- Гит репозиториудыг хадгалж, тэдгээртэй харьцаад хялбар UI.
- Багаар ажиллаж байхдаа нэг нэгнийхээ кодыг хянах байдлаар хийсэн өөрчлөлтуүдийг бусад гишүүдээр эсвэл хариуцсан мэргэжилтнээр нь хянуулж бусад мөчиртэй нэгтгэх хүсэлт илгээж болдог.
- Forks: Төсөлд ямар ч нөлөө үзүүлэхгүйгээр хуулбарлаж хөгжүүлэлт хийх боломж.
- Issues: Алдаа, bug болон сайжруулалтыг хянах нэг арга.
- Actions: Хөгжүүлэх, тестлэх, байршуулах ажлын урсгалыг автоматжуулдаг.

4.2 Javascript

JavaScript нь прототайпад суурилсан скриптийн хэл бөгөөд динамик, өгөгдлийн сүл төрөлжүүлэлттэй, нэгдүгээр зэргийн функцүүдтэй хэл юм. Энэ нь объект хандлагат, исператив, функциональ програмчлалын хэв маягуудыг дэмждэг олон парадигмт хэл юм.

Ажиллагааны хувьд товчхондоо JavaScript engine нь JS code-г хөрвүүлээд ажиллуулдаг. Анх веб хөтөч дээр ажиллахад зориулж гарч ирсэн тул олон янзын веб хөтчүүдэд ижилхэн ажиллах стандарт хэрэгтэй болсон нь ECMAScript болж л дээ. Бидний түгээмэл хэрэглэдэг стандарт нь ES6 (ECMAScript 6) буюу 2015 оны хувилбар нь.

JavaScript-г ашиглаад дан ганц веб биш мобайл апп, тоглоом эсвэл BackEnd сервер ч бас хийж болдог. React.js, Vue.js ашиглан FrontEnd, Node.js суурьтай Loopback, Nest.js ашиглан BackEnd эсвэл Ionic, React Native ашиглаад гар утасны апп хийж болно.

4.3 React

Фейсбүүк компани дотооддоо ашиглаж байсан технологио 2013 онд танилцуулсан нь Javascript хэлийг ашиглаж хийсэн Front-end library болох React технологи юм. Declarative UI хөгжүүлэлтийн аргыг хамгийн анх дэлгэрүүлж, өргөн хэрэглээнд нэвтрүүлж чадсан тул Declarative UI-н гол төлөөлөгч гэж явдаг. Уг технологийг ашиглахын тулд үндсэн хэдэн ойлголтууд авах хэрэгтэй. Үүнд component ба түүний lifecycle, javascript-н өргөжүүлсэн хувилбар болох jsx, мөн хамгийн чухал зүйл болох Virtual DOM нар багтана.

Declarative UI гэдэг нь хэрэглэгчийн интерфейсийн кодыг бичихдээ юу зурагдах буюу generator хийх үеийн интерфейсийг бүгдийг урьдчилан тодорхойлдог. Imperative програмчлалаас ялгаатай нь хязгаартай нөхцөлд яг юу хийхийг хатуугаар зааж өгөхгүйгээр тухайн state-с хамааруулж хэрэглэгчийн хүссэн зүйлийг гаргаж өгөх боломжтой.

React нь component-based буюу DOM дээр хэвлэж байгаа бүх зүйлс component байна гэсэн дурмийг баримталдаг. Component үүсгэж бичихийн давуу тал нь нэг бичсэн кодоо олон дахин бичигдэхээс зайлсхийж, дахин ашиглах боломжийг олгодог. Тус бур өөрсдийн гэсэн дотоод

төлөвтэй мөн гаднаас утга хүлээн авах чадвартай. Үүнийг бид Props гэж нэрлэдэг. Мөн component нь stateless, stateful гэж хоёр хуваагддаг ба stateful component нь өөрийн гэсэн төлөвтэй, түүнийгээ удирддаг, class болон hook ашигласан функцууд байна. React-н давуу тал нь state эсвэл props-н өөрчлөлтийг үргэлж хянаж байдаг тул өөрчлөлт орж ирэхэд бүтэн хуудсыг зурах бус зөвхөн тухайн өөрчлөгдсөн component-г л дахин зурдаг. Ингэснээр энгийн вебүүдээс илүү хурдтай ажилладаг.

JSX нь Javascript Extended гэсэн үгний товчлол бөгөөд энгийнээр javascript дотор HTML-н тагуудыг бичиж өгөх мөн кодыг илүү богино болгож хүссэн үр дүндээ хүрэх боломжийг олгодог. Үүний цаана Babel гэсэн transcompiler-г ашиглаж дундын хөрвүүлэлтийг хийдэг ба хэдийгээр HTML таг бичиж байгаа харагддаг ч код дунд цэвэр HTML-г огтоос бичиж өгдөггүй гэсэн үг юм.

```

1  export default function Container({ children }) {
2
3      return (
4          <div className ="container">
5              { children }
6          </div >
7      )
}

```

Код 4.1: JSX ашиглаж ”container” класстай html элемент буцаах компонент

Жинхэнэ DOM дээр богино хугацаанд олон өөрчлөлт хийхэд удах асуудал гарсан тул React маань Virtual DOM гэсэн abstraction давхарга үүсгэж өөрчлөлтүүдээ Virtual DOM дээрээ хадгалаад нэгдсэн нэг өөрчлөлтийг жинхэнэ DOM руугаа дамжуулдаг.

4.4 React Hook Form

React Hook Form нь React апликашнд дахь форм удирдах сан юм. Энэ нь функциональ бүрэлдэхүүн хэсгүүдийн төлөв байдал болон гаж нөлөөг удирдах илүү хялбар арга замаар

React 16.8-д нэвтрүүлсэн React hook дээр бүтээгдсэн. React Hook Form нь тусгайлан useForm болон useField hook ашиглан форм хөгжүүлэлтийг илүү үр ашигтай, уян хатан болгодог.

4.5 Bootstrap

2010 онд Twitter-т ажилладаг 2 залуу нийлж, тухайн үед жинхэнэ толгойны өвчин болоод байсан вебийн интерфейс хөгжүүлэх үйл явцыг хөнгөвчлөх зорилготой бяцхан төсөл эхлүүлсэн нь өдгөө дэлхий дээрх хамгийн олон хэрэглэгчидтэй, хамгийн өргөн ашиглагддаг CSS фрэймворк болох Bootstrap байсан юм.

Өнөөдөр тэрхүү бяцхан төсөл цар хүрээгээрээ асар том болж хөгжжээ. Анх Twitter Blueprint нэртэй байсан Bootstrap нь дэлхий дээрх хамгийн анхны CSS фрэймворкуудын нэг билээ. Бусдаас ялгарах гол давуу тал нь илүү найдвартай ажиллагаа, тасралтгүй хөгжүүлэлт байв. Одоо ч гэсэн хамгийн их хөгжүүлэлт хийгддэг, support сайтай фрэймворк хэвээрээ байна.

BuiltWith сайтын судалгаанаас харахад интернэтэд одоо ашиглагдаж буй нийт веб сайтуудын 20.6 хувийг, Javascript library-бүхий вебүүдийн 26.9 хувийг Bootstrap ашиглан хийсэн байдаг бөгөөд нийт фрэймворкийн зах зээлийн 72 хувийг дангаараа эзэлсээр. Мянган мянган алдартай компаниуд энэхүү фрэймворкийг ашигладаг бөгөөд тэдний тоонд Linkedin, Twitter, Spotify, Snapchat, Udemy, Baidu, Zoom, Netflix компаниуд ч багтаж байна.

4.6 GitBook

GitBook нь код хэлбэрээр бичиг баримт боловсруулах, мэдлэг солилцох ажлыг хялбаршуулдаг мэдлэгийн удирдлагын хэрэгсэл юм. Git болон Markdown ашиглан баримт бичиг бүтээх боломжийг олгодог бөгөөд мөн GitHub болон GitLab-тай интеграци хийх боломжийг олгодог. Энэхүү платформыг Adobe, Netflix, Apple, Snyk, Google зэрэг компаниуд зэрэг 2 сая гаруй хэрэглэгчид, олон мянган баг ашигладаг.

5. ХЭРЭГЖҮҮЛЭЛТ

5.1 Динамик баннерийн CRUD форм

5.1.1 Өгөгдлийн сан руу баннерийн CRUD хийх custom hook

Database-с баннерийн өгөгдлийг унших, өөрчлөх, устгах болон шинээр үүсгэхэд хялбарчлах үүднээс custom hook бичсэн. Custom hook-н нэр нь ”use” гэж эхэлсэн дахин ашиглах боломжтой функц бөгөөд бусад react hook-үүдийг дуудаж ажиллуулах боломжтой байдаг.

```
1  export function useBannerFetcher() {
2    const { status, error, response, fetcher } = useFetcher();
3    return {
4      status,
5      error,
6      response,
7      list: ({ filter, options }) =>
8        fetcher({
9          filter,
10         options,
11         method: "get",
12         url: `/admin/banners/list`,
13       }),
14      detail: (id) =>
15        fetcher({
16          method: "get",
17          url: `/admin/banners/detail/${id}`,
18          params: { id },
19        }),
20    };
21  }
22
23  export function useBannerUpdater() {
24    const { status, error, response, fetcher } = useFetcher();
25    const defaults = { status: "enabled", type: "slider", position: "home_top" };
26    return {
27      status,
28      error,
29      createBanner: ({ banner }) =>
30        fetcher({
31          method: "post",
32          body: {
33            content_banner: false,
34            is_additional: false,
35            ...banner,
```

```

36     ...defaults,
37   },
38   url: `/admin/banners/create`,
39 },
40 createBannerAdditional: ({ banner }) =>
41   fetcher({
42     method: "post",
43     body: {
44       content_banner: false,
45       is_additional: true,
46       ...banner,
47       ...defaults,
48     },
49     url: `/admin/banners/create`,
50   }),
51 createBannerContent: ({ banner }) =>
52   fetcher({
53     method: "post",
54     body: {
55       content_banner: true,
56       is_additional: false,
57       ...banner,
58       ...defaults,
59     },
60     url: `/admin/banners/create`,
61   }),
62 updateBanner: ({ id, banner }) =>
63   fetcher({
64     method: "post",
65     url: `/admin/banners/update`,
66     body: { id, ...banner, ...defaults },
67   }),
68 removeBanner: ({ id }) =>
69   fetcher({
70     method: "post",
71     url: `/admin/banners/remove`,
72     body: { id },
73   }),
74 };
75 }

```

Код 5.1: Баннерийн CRUD хийх custom hook

5.1.2 Баннерийн форм validation хийх

Энэхүү системд Yup хэмээх форм validation хялбарчлах зорилготой нэмэлт package-г ашигладаг.

Yup-ийн тусламжтайгаар та өөрийн өгөгдлийн схемийг тодорхойлж, дараа нь өгөгдөл тухайн схемд нийцэж байгаа эсэхийг шалгадаг. Форм validation нь вэб хөгжүүлэлтийн чухал хэсэг

бөгөөд хэд хэдэн чухал зорилготой:

- Формоор илгээсэн өгөгдөл хүлээгдэж буй шалгуур, чанарын стандартад нийцэж байгаа эсэхийг баталгаажуулах. Энэ нь буруу эсвэл бүрэн бус өгөгдлийг өгөгдлийн санд хадгалах эсвэл сервер рүү илгээхээс урьдчилан сэргийлэхэд тусалдаг.
- Хэрэглэгчдэд формоо илгээхээсээ өмнө буруу оруулсан мэдээллээ засч залруулахад тусалдаг.
- Бталгаажуулалт нь таны програмыг SQL injection эсвэл сайт хоорондын скрипт (XSS) гэх мэт хортой халдлагаас хамгаалахад тусалдаг.
- Серверийн ачааллыг бууруулах, шаардлагагүй хүсэлтийг бууруулдаг.

```

1 export const bannerSchema = yup.object().shape({
2   banner_type: yup.string().required().oneOf(["", " ", " ", " "]),
3   title: yup.string().required(" " " " ."),
4   description: yup.string(),
5   url: yup.string().url().nullable(),
6   button_title: yup.string(),
7   image: yup.string().url().required(" " " " ."),
8   video_url: yup.string().when("banner_type", {
9     is: " ",
10    then: yup
11      .string()
12      .required()
13      .matches(
14        /^(https?:\/\/)?((www\.)?youtube\.com|youtu\.be)\//.+$/,
15        "Youtube " " " .",
16      ),
17    otherwise: yup.string().nullable().notRequired(),
18  }),
19});

```

Код 5.2: Баннерийн форм validation

5.1.3 Фронтэнд хөгжүүлэлт

Орчин цагийн Фронтэнд хөгжүүлэлтэнд хамгийн түгээмэл ашиглагддаг филсофи бол бүрэлдэхүүн хэсгүүдийг аль болох олон жижиг хэсгүүдэд хувааж дараа нь тэдгээрийг ахин ашиглах явдал юм.

Хэрэглэгч үндсэн, нэмэлт, контент гэсэн 3 төрлийн баннер үүсгэх боломжтой ба үндсэн болон нэмэлт хэдэн ч баннер үүсгэх боломжтой, харин ганц л контент баннер үүсгэх боломжтой байна.

Хэрэглэгч үүсгэсэн байгаа баннеруудаа харах боломжтой байх үүднээс баннерийн жагсаалт компонентыг бичиж өгсөн. Мөн хэрэглэгч баннер шинээр үүсгэх эсвэл шинэчлэх үед баннерийн форм компонентийг дуудна. Баннерийн формийг дуудах үед URL-н параметрт id байвал тухайн id-тай баннер мэдээллийг авч формийн оролтуудад бичнэ. Үүний дараагаар хэрэглэгч өөрийн хүссэн мэдээллээ шинэчлэх боломжтой болно.

Баннерийн форм компонент

```

1  export default function BannerForm() {
2    const history = useHistory();
3    const { id } = useParams();
4    const fetcher = useBannerFetcher();
5    const updater = useBannerUpdater();
6    const shopId = getItem("current_merchant");
7
8    const [filter, setFilter] = useState({});
9    const [options, setOptions] = useState({});
10   const [hasContentBanner, setHasContentBanner] = useState(false);
11
12   const {
13     register,
14     control,
15     handleSubmit,
16     setValue,
17     unregister,
18     watch,
19     formState: { errors },
20   } = useForm({
21     resolver: yupResolver(bannerSchema),
22     defaultValues: {
23       banner_type: BannerType.MAIN,
24     },
25   });
26
27   const watchBannerType = watch("banner_type");
28
29   useEffect(() => {
30     if (id) {
31       fetcher.detail(id);
32     } else {
33       fetcher.list({
34         filter,
```

```

35     options,
36   );
37 }
38 }, [id, filter]);
39
40 useEffect(() => {
41   if (fetcher.response?.data) {
42     if (id) {
43       const banner = fetcher.response.data;
44       if (banner.content_banner) {
45         setValue("banner_type", BannerType.CONTENT);
46       } else if (banner.is_additional) {
47         setValue("banner_type", BannerType.ADDITIONAL);
48       } else {
49         setValue("banner_type", BannerType.MAIN);
50       }
51       for (const field of [
52         "title",
53         "description",
54         "image",
55         "url",
56         "button_title",
57         "video_url",
58       ]) {
59         setValue(field, banner[field]);
60       }
61     } else {
62       const banners = fetcher.response.data;
63       banners.map((b) => {
64         if (b.content_banner) {
65           setHasContentBanner(true);
66         }
67       });
68     }
69   }
70 }, [fetcher.response]);
71
72 useEffect(() => {
73   const status = updater.status;
74   const error = updater.error;
75   if (updater.status === "done") {
76     if (!error) {
77       history.push("/banners");
78       toast.success("      □      !");
79     } else {
80       toast.error(error.err_message || "      □      .□□□      □      .");
81     }
82   }
83 }, [updater.error, updater.status, updater.response]);
84
85 const onSubmit = (banner) => {
86   if (id) {
87     updater.updateBanner({ id, banner });
88     analytics.track("UPDATE_BANNER", {

```

```

89     merchant_id: Number(shopId) || {}.id,
90   });
91 } else {
92   if (watchBannerType === BannerType.ADDITIONAL) {
93     updater.createBannerAdditional({ banner });
94   } else if (watchBannerType === BannerType.CONTENT) {
95     updater.createBannerContent({ banner });
96   } else {
97     updater.createBanner({ banner });
98   }
99
100 analytics.track("CREATE_BANNER", {
101   merchant_id: Number(shopId) || {}.id,
102 });
103 }
104 };
105
106 if (fetcher.status === "pending") {
107   return <Loader />;
108 }
109
110 return (
111   <>
112     <ScreenHeader back_url='/banners' title={ `${id ? " " : " "}`}>
113       />
114     <Form loading={updater.status === "pending"} onSubmit={handleSubmit(
115       onSubmit)}>
116       <FormGroup label=' ' >
117         <select
118           defaultValue={watchBannerType}
119           onChange={(e) => setValue("banner_type", e.target.value)}
120           className='custom-select'
121           disabled={updater.status === "pending" || id}
122           data-toggle='select'
123           {...register("banner_type")}>
124             <option value={BannerType.MAIN} > </option>
125             <option value={BannerType.ADDITIONAL} > </option>
126             <option value={BannerType.CONTENT} disabled={hasContentBanner}>
127               </option>
128             </select>
129       </FormGroup>
130       {watchBannerType === BannerType.CONTENT && (
131         <p className='invalid-text-justify mx-1'>
132           .
133         </p>
134       )}
135       <FormGroup label=' ' error={errors.title?.message}>
136         <input
137           type='text'
138           disabled={updater.status === "pending"}
139           placeholder=' ' >
```

```

140     {...register("title")}
141     className={classNames("form-control", {
142       "is-invalid": !!errors.title,
143     })}
144   />
145 </FormGroup>
146 <FormGroup label='    ' error={errors.image?.message}>
147   <Controller
148     name='image'
149     control={control}
150     disabled={updater.status === "pending"}
151     render={({ field }) => (
152       <ImageUploader
153         field={field}
154         is_valid={errors.image}
155         message='    ' ' ' ' ' ' ' .'
156       />
157     )}
158   />
159 </FormGroup>
160 {! (watchBannerType === BannerType.CONTENT) ? (
161   <>
162     <FormGroup label='    ' error={errors.description?.message}>
163       <input
164         type='text'
165         defaultValue={}
166         disabled={updater.status === "pending"}
167         placeholder='    ' ' '
168         {...register("description")}
169         className={classNames("form-control", {
170           "is-invalid": !!errors.description,
171         })}
172       />
173     </FormGroup>
174
175     <p className='text-muted mb-4 text-justify mx-1'>
176       4:1 . / 1200x300px /
177     </p>
178     <FormGroup label='    ' error={errors.url?.message}>
179       <input
180         type='text'
181         disabled={updater.status === "pending"}
182         placeholder='    ' ' '
183         className={classNames("form-control", {
184           "is-invalid": !!errors.url,
185         })}
186         {...register("url")}
187       />
188     </FormGroup>
189     <FormGroup label='    ' error={errors.button_title?.message}>
190       <input
191         type='text'
192         disabled={updater.status === "pending"}
193         placeholder='    ' ' '

```

```

194         className={classNames("form-control", {
195             "is-invalid": !!errors.button_title,
196         })}
197         {...register("button_title")})
198     />
199     </FormGroup>
200   ) : (
201   <>
202     <FormGroup label='Youtube' error={errors.video_url?.message}>
203       <input
204         type='text'
205         disabled={updater.status === "pending"}
206         placeholder='Youtube URL' .'
207         {...register("video_url")})
208         className={classNames("form-control", {
209             "is-invalid": !!errors.video_url,
210         })}
211       />
212       </FormGroup>
213     </>
214   )
215   </Form>
216 </>
217 );
218 }
219 }
```

Код 5.3: Баннерийн форм компонент

Баннерийн жагсаалт компонент

```

1 export default function BannerList() {
2   const [filter, setFilter] = useState({});
3   const [options, setOptions] = useState({});
4   const { status, response, list } = useBannerFetcher();
5   const { status: removeStatus, removeBanner } = useBannerUpdater();
6
7   useEffect(() => {
8     list({
9       filter,
10      options,
11    });
12   }, [filter]);
13
14   useEffect(() => {
15     if (removeStatus === "done") {
16       list({
17         filter,
18         options,
19       });
20     }
21   }, [removeStatus]);
```

```

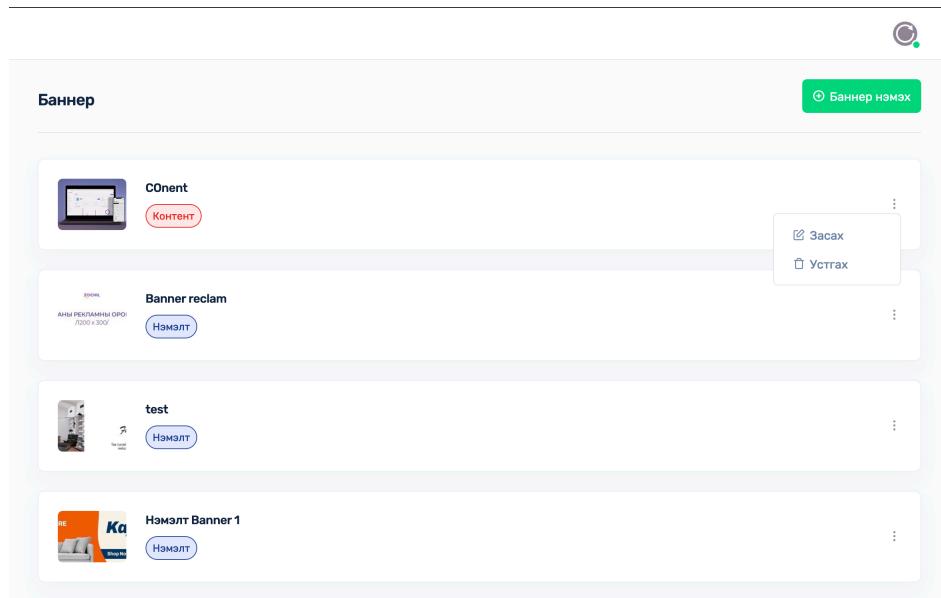
22
23 const { data: banners = [] } = response || {};
24
25 if (status === "pending" || removeStatus === "pending") {
26   return <Loader />;
27 }
28
29 return (
30   <>
31     <ScreenHeader new_url="/banners/new" title=" " />
32     {!banners[0] ? (
33       <div className='list-row mb-5 justify-content-center'>
34         <div className='card mt-5 justify-content-center'>
35           <div className='card-body pt-5 text-center'>
36             <div className='row justify-content-center pt-5'>
37               <div className='col-12 col-xl-12'>
38                 <img
39                   src='/img/illustrations/faq.svg'
40                   alt='...'
41                   className='img-fluid mt-n5 mb-4'
42                   style={{ height: "50vh" }}>
43               />
44               <h2 className='mb-2'> .</h2>
45               <p className='text-muted'> .</p>
46             </div>
47           </div>
48         </div>
49       </div>
50     ) : (
51       <div>
52         {banners.map((banner) => (
53           <ListItem
54             key={banner.id}
55             banner={banner}
56             onRemove={(id) => removeBanner({ id })}
57           />
58         )));
59       </div>
60     )
61   </>
62 );
63
64 }

```

Код 5.4: Баннерийн жагсаалт компонент

5.1.4 Ур дүн

Хэрэгжүүлэлтийн үр дүн:



Зураг 5.1: Баннерийн жагсаалт хэсэг

Баннер нэмэх

Баннерын төрөл
Ундсан баннер

Баннерын нэр
Баннерын гарчигаа оруулна уу

Баннер зураг
Баннерын зурага орлога энд дарж эсвэл зөөж оруулна уу.

Баннерын текст
Баннерын гарчигаа оруулна уу

Та 4:1 харьцаатай зураг оруулна уу. / 1200x300px /

Баннерын линк
Баннерын линкээ оруулна уу

Товчийн текст
Товчийн текстээ оруулна уу

Хадгалах

Зураг 5.2: Баннерийн форм хэсэг

5.2 Арга хэмжээ үүсгэх CRUD форм

5.2.1 Өгөгдлийн сан руу арга хэмжээг CRUD хийх custom hook

```

1  export function useEventFetcher() {
2    const { status, error, response, fetcher } = useFetcher();
3    return {
4      status,
5      error,
6      response,
7      list: ({ filter, options }) =>
8        fetcher({
9          filter,
10         options,
11         method: "get",
12         url: `/admin/events`,
13       }),
14      detail: (id) =>
15        fetcher({
16          method: "get",
17          url: `/admin/events/${id}`,
18        }),
19    };
20  }
21
22  export function useEventUpdater() {
23    const { status, error, response, fetcher } = useFetcher();
24    const defaults = {};
25    return {
26      status,
27      error,
28      createEvent: ({ event }) =>
29        fetcher({
30          method: "post",
31          url: `/admin/events`,
32          body: { ...event, ...defaults },
33        }),
34      updateEvent: ({ id, event }) =>
35        fetcher({
36          method: "put",
37          url: `/admin/events/${id}`,
38          body: { ...event, ...defaults },
39        }),
40      removeEvent: ({ id }) =>
41        fetcher({
42          method: "delete",
43          url: `/admin/events/${id}`,
44        }),
45    };
46  }

```

Код 5.5: Арга хэмжээн CRUD хийх custom hook

5.2.2 Арга хэмжээн форм validation хийх

```

1  export const eventSchema = yup.object().shape({
2    name: yup.string().required(),
3    start_at: yup.string().required(),
4    end_at: yup.string().required(),
5    address: yup.string().required(),
6    youtube_url: yup
7      .string()
8      .nullable()
9      .matches(/^https?:\/\/((www\.)?youtube\.com|youtu\.be)\/.+$/, {
10        message: "Youtube URL is invalid",
11        excludeEmptyString: true,
12      }),
13  });

```

Код 5.6: Арга хэмжээн форм validation

5.2.3 Фронтэнд хөгжүүлэлт

Арга хэмжээн форм компонент

```

1  export default function EventForm() {
2    const history = useHistory();
3    const { id } = useParams();
4    const fetcher = useEventFetcher();
5    const updater = useEventUpdater();
6    const {
7      register,
8      control,
9      handleSubmit,
10     setValue,
11     formState: { errors },
12   } = useForm({
13     resolver: yupResolver(eventSchema),
14     defaultValues: {
15       images: [],
16     },
17   });
18   useEffect(() => {
19     if (id) {
20       fetcher.detail(id);
21     }
22   }, [id]);
23   useEffect(() => {
24     if (fetcher.response?.data) {
25       const event = fetcher.response.data;
26       for (const field of [
27         "name",
28         "description",
29         "start_at",

```

```

30     "images",
31     "end_at",
32     "address",
33     "youtube_url",
34   ]) {
35     setValue(field, event[field]);
36   }
37 }
38 }, [fetcher.response]);
39 useEffect(() => {
40   const error = updater.error;
41   if (updater.status === "done") {
42     if (!error) {
43       history.push("/events");
44     } else {
45       toast.error(error.err_message || " " . " " " " . ");
46     }
47   }
48 }, [updater.error, updater.status, updater.response]);
49 const onSubmit = (event) => {
50   if (id) {
51     updater.updateEvent({ id, event });
52   } else {
53     updater.createEvent({ event });
54   }
55 };
56 if (fetcher.status === "pending") {
57   return <Loader />;
58 }
59 return (
60   <>
61     <ScreenHeader back_url='/events' title={` ${id ? " " : " "}`}
62       />
63     <Form loading={updater.status === "pending"} onSubmit={handleSubmit(
64       onSubmit)}>
65       <FormGroup label=' ' error={errors.name?.message}>
66         <input
67           type='text'
68           disabled={updater.status === "pending"}
69           placeholder=' ' ' ' ' .
70           {...register("name")}
71           className={classNames("form-control", {
72             "is-invalid": !errors.name,
73           })}
74         />
75       </FormGroup>
76       <FormGroup label=' ' error={errors.description?.message}>
77         <textarea
78           autoComplete={0}
79           autoCorrect={0}
80           disabled={updater.status === "pending"}
81           spellCheck={false}
82           placeholder=' ' ' ' ' .
83           {...register("description")}>

```

```

82     className={classNames("form-control pl-2", {
83       "is-invalid": !!errors.description,
84     })}
85   />
86 </FormGroup>
87 <div className='row'>
88   <div className='col-12 col-md-6'>
89     <FormGroup label='    ' error={errors.start_at?.message}>
90       <Controller
91         name='start_at'
92         control={control}
93         render={({ onChange, value, field }) => (
94           <DatePicker
95             selected={Date.parse(field.value)}
96             dateFormat='yyyy/MM/dd hh:mm aa'
97             showTimeSelect
98             timeFormat='hh:mm'
99             timeIntervals={15}
100            onChange={(value) => field.onChange(value.toISOString())}
101            placeholderText='    '
102            className={classNames("form-control", {
103              "is-invalid": !!errors.start_at,
104            })}
105          />
106        )}
107      />
108    </FormGroup>
109  </div>
110 <div className='col-12 col-lg-6'>
111   <FormGroup label='    ' error={errors.end_at?.message}>
112     <Controller
113       name='end_at'
114       control={control}
115       render={({ onChange, value, field }) => (
116         <DatePicker
117           selected={Date.parse(field.value)}
118           dateFormat='yyyy/MM/dd hh:mm aa'
119           showTimeSelect
120           timeFormat='hh:mm'
121           timeIntervals={15}
122           onChange={(value) => field.onChange(value.toISOString())}
123           placeholderText='    '
124           className={classNames("form-control", {
125             "is-invalid": !!errors.end_at,
126           })}
127         />
128       )}
129     />
130   </FormGroup>
131 </div>
132 <div>
133   <FormGroup label='    ' error={errors.images?.message}>
134     <Controller
135       name='images'

```

```

136     control={control}
137     disabled={updater.status === "pending"}
138     render={({ field }) => (
139       <ImageUploader
140         field={field}
141         is_valid={errors.images}
142         message='□ □ □ □ □ □ .'
143       />
144     )}
145   />
146 </FormGroup>
147 <FormGroup label=' ' error={errors.address?.message}>
148   <textarea
149     autoComplete={0}
150     autoCorrect={0}
151     disabled={updater.status === "pending"}
152     spellCheck={false}
153     placeholder='□ □ □ □ .'
154     {...register("address")}
155     className={classNames("form-control", {
156       "is-invalid": !!errors.address,
157     })}
158   />
159 </FormGroup>
160 <FormGroup label='Youtube' error={errors.youtube_url?.message}>
161   <div
162     className={classNames("input-group", {
163       "is-invalid": !!errors.event_video_url,
164     })}
165   >
166     <input
167       type='url'
168       disabled={updater.status === "pending"}
169       placeholder='Youtube .'
170       className={classNames("form-control", {
171         "is-invalid": !!errors.youtube_url,
172       })}
173       {...register("youtube_url")}
174     />
175     <div className='input-group-append'>
176       <span className='input-group-text' id='basic-addon2'>
177         <i className='fe fe-youtube' />
178       </span>
179     </div>
180   </div>
181 </FormGroup>
182 </Form>
183 </>
184 );
185 }

```

Код 5.7: Арга хэмжээн форм компонент

Арга хэмжээн жагсаалт компонент

```

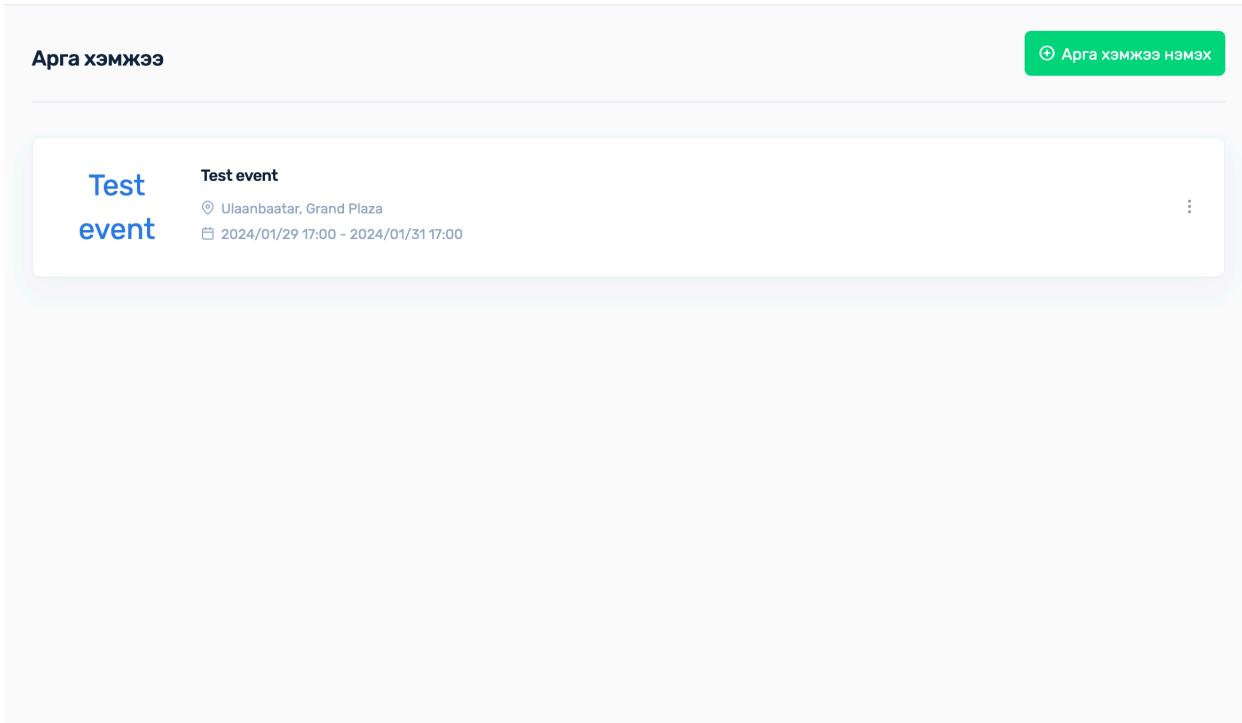
1  export default function EventList() {
2    const [filter, setFilter] = useState({});
3    const [options, setOptions] = useState({});
4    const { status, response, list } = useEventFetcher();
5    const { status: removeStatus, removeEvent } = useEventUpdater();
6    useEffect(() => {
7      list({
8        filter,
9        options,
10      });
11    }, [filter, options]);
12    useEffect(() => {
13      if (removeStatus === "done") {
14        list({
15          filter,
16          options,
17        });
18      }
19    }, [removeStatus]);
20    const { data: events = [] } = response || {};
21    if (status === "pending" || removeStatus === "pending") {
22      return <Loader />;
23    }
24    return (
25      <>
26        <ScreenHeader new_url='/events/new' title=' ' />
27        {!events[0] ? (
28          <div className='list_row_mb-5_justify-content-center'>
29            <div className='card_mt-5_justify-content-center'>
30              <div className='card-body_pt-5_text-center'>
31                <div className='row_justify-content-center_pt-5'>
32                  <div className='col-12_col-xl-12'>
33                    <img
34                      src='/img/illustrations/faq.svg'
35                      alt='...'
36                      className='img-fluid_mt-n5_mb-4'
37                      style={{ height: "50vh" }}>
38                  />
39                  <h2 className='mb-2'> .</h2>
40                  <p className='text-muted'>
41                    .
42                    </p>
43                  </div>
44                </div>
45              </div>
46            </div>
47          ) : (
48            <div>
49              {events.map((event) => (
50                <ListItem key={event.id} event={event} onRemove={(id) =>
51                  removeEvent({ id })} />

```

```
52     })})  
53   </div>  
54 }  
55 </>  
56 );  
57 }
```

Код 5.8: Арга хэмжээн жагсаалт компонент

5.2.4 Yp дүн



Зураг 5.3: Баннерийн жагсаалт хэсэг

Арга хэмжээ нэмэх

Нэр
Арга хэмжээний нэрээ оруулна уу.

Танилцуулга
Арга хэмжээний танилцуулгаа оруулна уу.

Эхлэх огноо **Дуусах огноо**

Арга хэмжээ дуусах огноо Арга хэмжээ дуусах огноо

Зураг
Арга хэмжээний зургаа энд дарж эсвэл зөөж оруулна уу.

Хаяг
Арга хэмжээний хаягийг оруулна уу.

Youtube линк
Youtube видео линк.

Хадгалах

Зураг 5.4: Баннерийн форм хэсэг

5.3 Үйлчилгээний төлбөрийн саналын хуудас

Zochil платформ шинээр үйлчилгээний төлбөрийн багц гаргаж байгаа бөгөөд Админ веб системд багц худалдаж аваагүй хэрэглэгчдэд саналын шинээр хуудас үүсгэх шаардлагатай болсон.

```

1  export default function PricingPlans() {
2    const [show, setShow] = useState(false);
3    const [showMonth, setShowMonth] = useState(false);
4    const handleClose = () => setShow(false);
5    const handleMonthClose = () => setShowMonth(false);
6    const shop_id = getItem("current_merchant");
7    const [qrCode, setQrCode] = useState("");
8    const [price, setPrice] = useState(66000);
9    const [month, setMonth] = useState(12);
10   const [planUid, setPlanUid] = useState("basic");
11   const [contID, setContId] = useState(0);
12   const { status, response, createInvoice } = useBillUpdater();
13   const { status: cStatus, response: cResponse, detail } = useBillFetcher();
14   const callPaymentService = (amountPMonth) => {
15     setShowMonth(false);
16     let detailInvoice = {
17       merchant_id: shop_id,
18       quantity: month,
19       plan_uid: planUid,
20     };
21     setPrice(amountPMonth);
22     createInvoice({ detailInvoice });
23     setShow(true);
24   };
25   const choosePlan = (planIndex) => {
26     setPrice(tiers[planIndex].price);
27     setPlanUid(tiers[planIndex].uid);
28     setShowMonth(true);
29   };
30
31   const checkInvoicePaid = () => {
32     detail(contID);
33   };
34
35   useEffect(() => {
36     setQrCode(response?.qrcode);
37     if (response?.id != null) {
38       setContId(response.id);
39     }
40   }, [response]);
41
42   useEffect(() => {}, [month]);
43
44   useEffect(() => {
45     if (cStatus === "done") {

```

5.3. ҮЙЛЧИЛГЭЭНИЙ ТӨЛБӨРИЙН САНАЛЫН ХУУГДАЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
46     if (cResponse) {
47       if (cResponse?.success) {
48         toast.success("    □    □    .");
49         document.location.href = "/";
50       } else {
51         toast.error("    □    □    □    .");
52       }
53     }
54   }, [cResponse, cStatus]);
55
56   return (
57     <div className='pricing-container mt-4'>
58       <Modal show={showMonth} onHide={handleMonthClose}>
59         <Modal.Header closeButton>
60           <Modal.Title>    </Modal.Title>
61         </Modal.Header>
62         <div className='d-flex flex-column bg-'>
63           <div
64             className='d-flex justify-content-between align-items-center my-3
65               mx-5 px-3 py-3 my-2'
66             style={{ background: "#f9fbfd", borderRadius: 16, cursor: "
67               pointer" }}>
68             onClick={() => setMonth(12)}
69           >
70             <label
71               className=''
72               style={{ margin: "0 0 0 12px", fontSize: 20 }}>
73               htmlFor='month12'
74             >
75               12
76             </label>
77             <input
78               type='checkbox'
79               id='month12'
80               name='month12'
81               style={{
82                 width: "18px",
83                 height: "18px",
84               }}
85               checked={month == 12}
86             />
87           </div>
88           <div className='d-flex justify-content-between align-items-center
89             mx-5 px-3 py-3 my-2'>
90             <div>    :</div>
91             <div
92               style={
93                 month == 12
94                   ? { textDecoration: "line-through" }
95                   : { textDecoration: "none" }
96               }
97             >
98               {numeral(month * price).format("0,0") }
99           </div>
100         </div>
101       </Modal>
102     </div>
103   );
104 }
```

5.3. ҮЙЛЧИЛГЭЭНИЙ ТӨЛБӨРИЙН САНАЛЫН ХУУГДААГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
97      </div>
98  </div>
99  <div className='d-flex justify-content-between align-items-center' style={{margin: '5px 3px 3px 2px'}}>
100    <div> </div>
101    <div>{month == 12 ? "10%" : "0%"}</div>
102  </div>
103
104  <div class='dropdown-divider'></div>
105  <div className='d-flex justify-content-between align-items-center' style={{margin: '5px 3px 3px 2px'}}>
106    <div> </div>
107    <div>
108      {numeral(month == 12 ? month * price * 0.9 : month * price).format("0,0")}</div>
109    </div>
110  </div>
111  </div>
112  <Modal.Footer>
113    <Button className='btn btn-light' variant='secondary' onClick={handleMonthClose}>
114
115    </Button>
116    <Button
117      type='submit'
118      className='btn btn-success'
119      variant='primary'
120      onClick={() => callPaymentService(price)}>
121    >
122
123    </Button>
124  </Modal.Footer>
125</Modal>
126<Modal show={show} onHide={handleClose}>
127  <Modal.Header closeButton>
128    <Modal.Title> </Modal.Title>
129  </Modal.Header>
130  <div className='row justify-content-center'>
131    <div className='d-flex justify-content-center col-12 col-md-6'>
132      {status === "done" && (
133        <QRCode
134          value={`${qrCode}`}
135          style={{margin: 50, height: 150, width: 150}}
136        >
137      )}
138    </div>
139  </div>
140  {status === "pending" && (
141    <div style={{margin: 50}}>
142      <SkeletonTheme>
143        <Skeleton height={150} width={150} />
144      </SkeletonTheme>
145    </div>
146  )}
```

5.3. ҮЙЛЧИЛГЭЭНИЙ ТӨЛБӨРИЙН САНАЛЫН ХУУДААГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
147         </div>
148     )
149   </div>
150   <div
151     className='col-12 col-md-6 d-flex justify-content-center'
152     style={{ marginTop: 50 }}
153   >
154     <div className='list-group list-group-flush my-n3'>
155       <div className='list-group-item'>
156         <small className='text-muted justify-content-center'> </
157           small>
158         <h4>
159           <p> </p>
160           </h4>
161         </div>
162         <div className='list-group-item'>
163           <small className='text-muted justify-content-center'></
164             small>
165           <h4>
166             <p>
167               {numeral(month == 12 ? month * price * 0.9 : month * price
168                 ).format(
169                   "0,0",
170                 )}
171
172           </p>
173           </h4>
174         </div>
175       </div>
176     <Modal.Footer>
177       <Button className='btn btn-light' variant='secondary' onClick={
178         handleClose}>
179
180       </Button>
181       <Button
182         type='submit'
183         className='btn btn-success'
184         variant='primary'
185         onClick={checkInvoicePaid}
186         disabled={cStatus === "pending"}
187       >
188         {cStatus === "pending" && (
189           <span className='spinner-border spinner-border-sm mr-1' />
190         )}
191
192     </Modal.Footer>
193   </Modal>
194   <div className='mb-5'>
195     <h1 className='header-title'> </h1>
196     <h3 className='header-subtitle'> </h3>
197   </div>
```

5.3. ҮЙЛЧИЛГЭЭНИЙ ТӨЛБӨРИЙН САНАЛЫН ХУУДАА ЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
197 |     <div className='timeline'>
198 |       <div className='timeline-progress' />
199 |       <div className='timeline-item'>
200 |         <div className='timeline-contain'>{`1600`}</div>
201 |       </div>
202 |     </div>
203 |     <div className='tiers-grid'>
204 |       {tiers.map((tier) => (
205 |         <div
206 |           key={tier.uid}
207 |           className={classNames("card", {
208 |             "tier-popular": tier.isRecommended,
209 |           })}>
210 |           >
211 |             <div className='card-body'>
212 |               <div className='tier-header'>
213 |                 <div className='tier-wrapper'>
214 |                   <h1 className='tier-title'>{tier.name}</h1>
215 |                   {tier.isRecommended && <p className='tier-tag'> </p>}
216 |                 </div>
217 |                 <p className='tier-price'>
218 |                   <span>{tier.priceMonthly}</span>
219 |                   /
220 |                 </p>
221 |                 <div className='tier-description'>{tier.description}</div>
222 |                 <button onClick={() => choosePlan(tier.uid)} className='tier-button'>
223 |
224 |                   </button>
225 |                 </div>
226 |               <div>
227 |                 <p className='tier-label'></p>
228 |                 <div>
229 |                   {tier.features.map((feature, i) => (
230 |                     <div key={i} className='tier-feature'>
231 |                       <div className='feature-wrapper'>
232 |                         <div className='feature-text'>
233 |                           <i className='fe fe-check-circle' />
234 |                           <p>{feature}</p>
235 |                         </div>
236 |                       </div>
237 |                       {tier.features.length - 1 !== i && <hr />}
238 |                     </div>
239 |                   )));
240 |                 </div>
241 |               </div>
242 |             </div>
243 |           )));
244 |         </div>
245 |       </div>
246 |     );
247 |   };
248 | }
```

5.4. МОД БҮТЭЦТЭЙ БҮТЭЭГДЭХҮҮНИЙ КАТЕГОРИБҮЛЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

Код 5.9: Үйлчилгээний төлбөрийн саналын компонент

5.3.1 Yp дүн

The screenshot shows the Yp service's pricing page. At the top, it says "1600 Харилцагчид". Below that are four plan options:

- Starter**: **66,000₮/сарын**. Description: Онлайн дэлгүүр үүсгэх үндсэн багц. Buttons: Сонгох, УЙЛЧИЛГЭЭ, Бүтээгдэхүүний мэдээллийн удирдлага, Контент удирдлага.
- Basic**: **88,000₮/сарын**. Description: Төлбөрийн хэрэгсэл багтаасан багц. Buttons: Сонгох, УЙЛЧИЛГЭЭ, Бүтээгдэхүүний мэдээллийн удирдлага, Контент удирдлага, Домайн холболт, Төлбөр тооцоо автоматжуулах, Хүргэлтийн үйлчилгээ холбох.
- Standart**: **110,000₮/сарын**. Description: Бүх төлбөрийн хэрэгсэлийн хялбар шийдэл. Buttons: Сонгох, УЙЛЧИЛГЭЭ, Бүтээгдэхүүний мэдээллийн удирдлага, Контент удирдлага, Урсдаг зар, Нэмэлт баннерууд, Pop Up сурталчилгаа, Домайн холболт. A "Санал болгох" button is also present.
- Advanced**: **150,000₮/сарын**. Description: Бүгдийг багтаасан. Buttons: Сонгох, УЙЛЧИЛГЭЭ, Бүтээгдэхүүний мэдээллийн удирдлага, Контент удирдлага, Урсдаг зар, Нэмэлт баннерууд, Pop Up сурталчилгаа, Домайн холболт.

Зураг 5.5: Үйлчилгээний төлбөрийн саналын компонент

5.4 Мод бүтэцтэй бүтээгдэхүүний категори

5.4.1 Өгөгдлийн сангаас категоруудыг авах custom hook

```
1 export function useMasterCategoryFetcher() {
2   const shopId = getItem("current_merchant");
3   const { status, error, response, fetcher } = useFetcher();
4
5   return {
```

5.4. МОД БҮТЭЦТЭЙ БҮТЭЭГДЭХҮҮНИЙ КАТЕГОРИБҮЛЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
6   status,
7   error,
8   response,
9   list: ({ filter, options }) =>
10    fetcher({
11      filter,
12      options,
13      method: "get",
14      url: `/admin/product-main-categories/list`,
15    }),
16   listAll: () =>
17    fetcher({
18      method: "get",
19      url: `/admin/product-main-categories/list-all`,
20    }),
21   searchName: ({ name }) =>
22    fetcher({
23      method: "get",
24      url: `/admin/product-main-categories/search?name=${name}`,
25    }),
26  };
27 }
```

Код 5.10: Категоруудыг авах custom hook

5.4.2 Фронтэнд хөгжүүлэлт

Категоруудыг нэмэх компонент

```
1 export default function MasterCategory() {
2   const {
3     status: masterStatus,
4     response: masterResponse,
5     list: masterList,
6     listAll: masterListAll,
7   } = useMasterCategoryFetcher();
8
9   const [showMasterCateg, setShowMasterCateg] = useState(false);
10  const [checkedMasterCateg, setCheckedMasterCateg] = useState([]);
11  const masterCategRef = useRef();
12
13  useEffect(() => {
14    const masterCategHandler = (e) => {
15      if (masterCategRef.current && !masterCategRef.current.contains(e.target))
16        setShowMasterCateg(false);
17    }
18  });
19
20  document.addEventListener("mousedown", masterCategHandler);
21  return () => {
```

5.4. МОД БҮТЭЦТЭЙ БҮТЭЭГДЭХҮҮНИЙ КАТЕГОРИБҮЛЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
22     document.removeEventListener("mousedown", masterCategHandler);
23   };
24 }, []);
25
26 function masterCategoryAdd(id, name) {
27   setCheckedMasterCateg((checkedMasterCateg) => [...checkedMasterCateg, {
28     id, name }]);
29 }
30
31 function masterCategoryRemove(id) {
32   for (const master of checkedMasterCateg) {
33     if (id === master.id) {
34       checkedMasterCateg.splice(checkedMasterCateg.indexOf(master), 1);
35       break;
36     }
37   }
38   setCheckedMasterCateg([...checkedMasterCateg]);
39 }
40
41 useEffect(() => {
42   masterListAll({});
```

42 }, []);

```
43
44 const { data: masterCategories = [] } = masterResponse || {};
45
46 useEffect(() => {
47   if (fetcher.response?.data) {
48     setCheckedCategories(fetcher.response?.data?.categories || []);
49   }
50 }, [fetcher.response]);
51
52 useEffect(() => {
53   setValue("main_category_ids", checkedMasterCateg);
54 }, [checkedMasterCateg]);
55
56 return (
57   <>
58   <Col sm={4}>
59     <div className='card'>
60       <div className='card-header'>
61         <h5 className='header-title'></h5>
62       </div>
63       <div className='card-body'>
64         <div className='row_justify-content-center'>
65           {checkedMasterCateg.map((masterCateg) => (
66             <div className='checked-master-list' key={masterCateg.id}>
67               <span className=''{>{masterCateg.name}</span>
68               <a
69                 type='button'
70                 className='text-sm_text-danger'
71                 onClick={() => masterCategoryRemove(masterCateg.id)}
72               >
73                 <i className='fe_fe-trash' />
74             </div>
75           ))
76         </div>
77       </div>
78     </div>
79   </Col>
80 </>
81 <div style={{ display: 'flex', justify-content: 'space-around' }}>
82   <button style={{ width: '150px' }}>САЛГАА</button>
83   <button style={{ width: '150px' }}>ХАМГИУУЛСАА</button>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 </div>
132 </div>
133 </div>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </div>
142 </div>
143 </div>
144 </div>
145 </div>
146 </div>
147 </div>
148 </div>
149 </div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
```

5.4. МОД БҮТЭЦТЭЙ БҮТЭЭГДЭХҮҮНИЙ КАТЕГОРИБҮЛЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

```
75      </div>
76    ))
77    <button
78      type='button'
79      className='btn btn-secondary btn-xl w-100'
80      onClick={() => setShowMasterCateg(!showMasterCateg)}
81    >
82      <i className='fe fe-plus text-white font-weight-bold mr-2' />
83    </button>
84  </div>
85 </div>
86 </div>
87 </Col>
88 <div
89   ref={masterCategRef}
90   className={classNames("right-slide", {
91     "show-right-slide": showMasterCateg,
92     "hide-right-slide": !showMasterCateg,
93   })}
94 >
95   <div className='right-slide-header'>
96     <button
97       className='btn btn-secondary btn-sm'
98       onClick={() => setShowMasterCateg(false)}
99     >
100       <i className='fe fe-x' />
101     </button>
102   </div>
103   <div className='right-slide-body'>
104     <p className='text-muted'> :</p>
105     {masterCategories.map((masterCategory) => (
106       <MasterCategoryItem
107         key={masterCategory.id}
108         category={masterCategory}
109         checkedCategories={checkedMasterCateg}
110         categoryAdd={masterCategoryAdd}
111         categoryRemove={masterCategoryRemove}
112       />
113     )));
114   </div>
115 </div>
116 </>
117 );
118 }
```

Код 5.11: Категоруудыг нэмэх компонент

Категоруудын жагсаалтын item компонент

Категорууд нь мод бүтэцтэй учир нэг категори дотор хэдэн ч категори байж болно.

```
1 export default function MasterCategoryItem({
```

5.4. МОД БҮТЭЦТЭЙ БҮТЭЭГДЭХҮҮНИЙ КАТЕГОРИБҮЛЭГ 5. ХЭРЭГЖҮҮЛЭЛТ

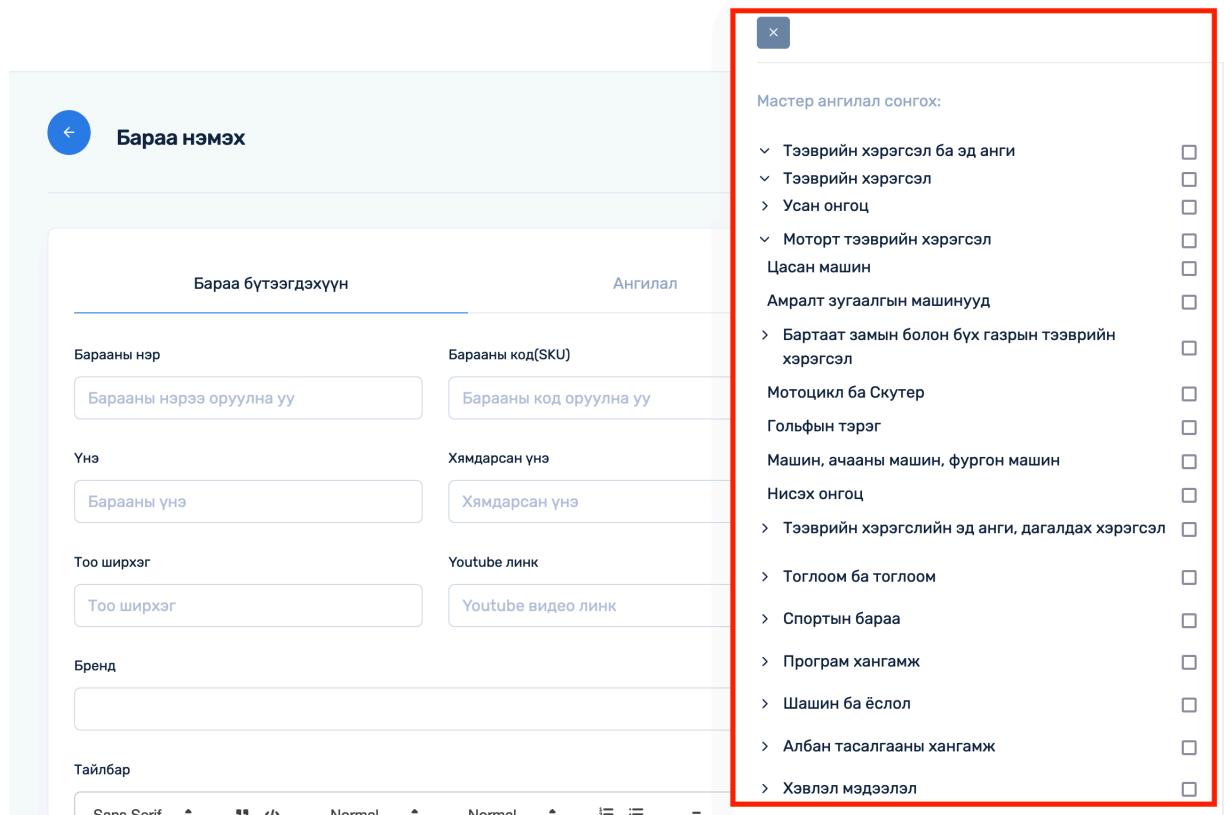
```
2 category,
3 checkedCategories,
4 categoryAdd,
5 categoryRemove,
6 } {
7   const [categItem, setCategItem] = useState(false);
8   const [isOpen, setIsOpen] = useState(false);
9
10 useEffect(() => {
11   const isCategoryChecked = checkedCategories.some((categ) => categ.id ===
12     category.id);
13   setCategItem(isCategoryChecked);
14 }, [checkedCategories, category.id]);
15
16 function categoryStatus() {
17   if (!categItem) {
18     setCategItem(true);
19     categoryAdd(category.id, category.name);
20   } else {
21     setCategItem(false);
22     categoryRemove(category.id);
23   }
24 }
25
26 const nestedCategory = (category.children || []).map((childCategory) => {
27   return (
28     <MasterCategoryItem
29       key={childCategory.id}
30       category={childCategory}
31       checkedCategories={checkedCategories}
32       categoryAdd={categoryAdd}
33       categoryRemove={categoryRemove}
34     />
35   );
36 });
37
38 return (
39   <div className='mb-2'>
40     <div className={`master-select-item`} style={{ marginLeft: `0.5 * ${
41       category.level}rem` }}>
42       <div className='master-item-name' onClick={() => setIsOpen(!isOpen)}>
43         {category.children && (
44           <i
45             className={classNames("fe fe-chevron-right dropdown-arrow", {
46               "rotate-90": !isOpen,
47             })}
48           />
49         )
50         <div type='button' className='master-item-title'>
51           {category.name}
52         </div>
53       </div>
54       <input className='mr-4' type='checkbox' checked={categItem} onChange
55         ={categoryStatus} />
```

5.5. ZOCHIL ПЛАТФОРМЫН ГАРЫН АВЛАГЫГ ШИНЭЧЛЭХ. ХЭРЭГЖҮҮЛЭЛТ

```
53     </div>
54     <Collapse isOpen={isOpen}>{nestedCategory}</Collapse>
55   );
56 }
57 }
```

Код 5.12: Категоруудыг нэмэх компонент

5.4.3 Yp дүн



Зураг 5.6: Категоруудыг нэмэх компонент

5.5 Zochil платформын гарын авлагыг шинэчлэх

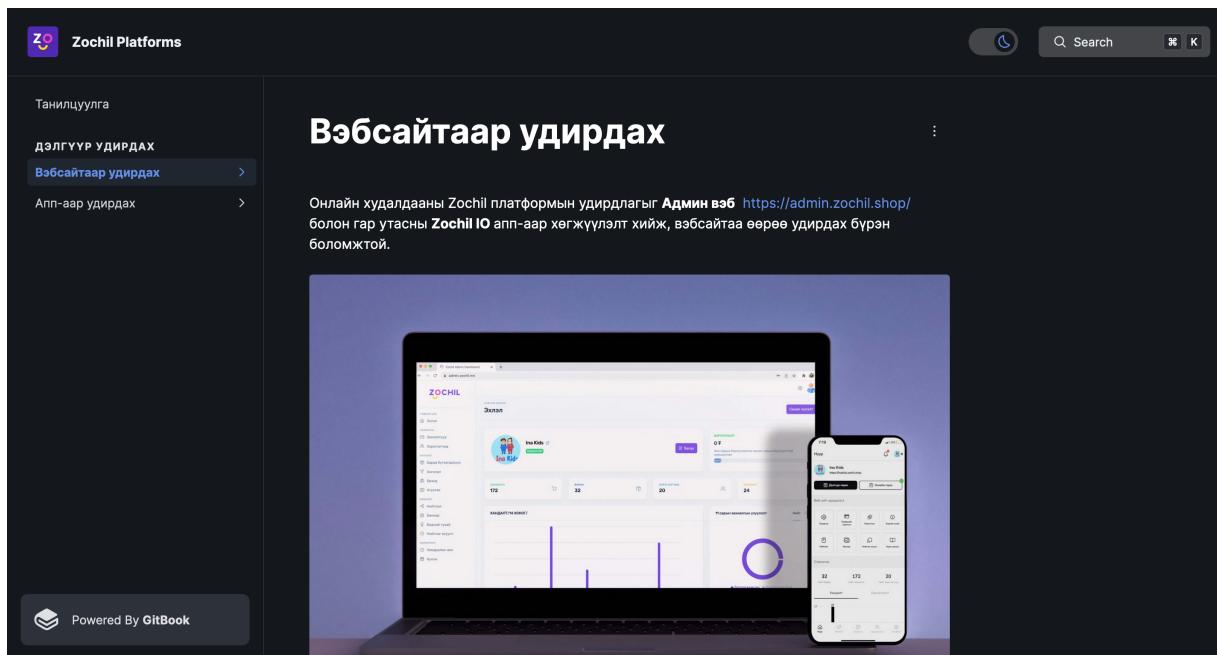
Одоогийн Zochil платформын гарын авлага нь хуучирсан бөгөөд Админ веб болон гар утасны Zochil IO апп-н шинэ хувилбарт зориулж хэрэглэгчдэд гарын авлагыг шинэчлэх шаардлагатай болсон. Гарын авлагыг GitBook платформыг ашиглан хийсэн. Zochil платформын гарын

5.5. ZOCHIL ПЛАТФОРМЫН ГАРЫН АВЛАГЫГ ШИНЭЧЛЭХ. ХЭРЭГЖҮҮЛЭЛТ

авлага нь бизнес эрхлэгч цахим дэлгүүрээ хэрхэн Админ веб болон гар утасны Zochil IO апп-аар хэрхэн удирдахыг зураг болон текстэн мэдээллийн тусlamжтай алхам алхмаар харуулсан.

Шинэчлэгдсэн Zochil платформын гарын авлагын холбоос: <https://zochil-tech-2024.gitbook.io/zochil-docs/>

5.5.1 Yr дүн



Зураг 5.7: Категоруудыг нэмэх компонент

6. ДҮГНЭЛТ

6.1 Үр дүнгийн тайлан

Миний бие Энхбаярын Жавхлан нэг сарын хугацаанд Зочил Технологи компанид мэргэжлийн дадлагыг амжилттай гүйцэтгэж дуусгалаа. Уг хугацаанд хичээлийн хүрээнд үзсэн онолын ойлголтуудыг практик дээр туршиж, хэрэгжүүлсэн ба хөгжүүлэлт голчилсон технологийн компанийн ерөнхий үйл ажиллагаа, баг хооронд зохицон ажиллах чадвар, хөгжүүлэлтийн шинэ арга барилуудыг амжилттай эзэмшсэн гэж дүгнэж байна.

Back-end, Front-end-ийн хамгийн орчин үеийн технологиудтай танилцаж бодит төсөл дээр ажиллаа. Мөн бие даан асуудлыг шийдвэхийн тулд алхам дэс дараатай асуудлыг шийдвэрлэх мөн ашиглаж буй сан, технологийнхоо гарын авлага буюу documentation-тай илүү сайн танилцаж уг технологийнхоо цаана нь буй концепцийг хялбараар ойлгох гэх мэт чадваруудыг эзэмшсэн.

Энэхүү дадлагын хугацаанд өөрийн ямар чиглэлд сонирхолтой, цаашид аль чиглэлрүү гүнзгийрэх талаараа үнэтэй туршлага цуглууллаа. Мөн жижиг стартап компанийн ажлын орчин, арга барил, болон зах зээлийн бодит амьдралыг биеэрээ мэдэрч суралцаж цаашдаа энэ замналаар нийгэмд хэрэгцээтэй шийдвэл санаа хөгжүүлэх сэдлийг төрүүллээ.

Bibliography

- [1] React Documentation,
<https://react.dev/>
- [2] Declarative програмчлал болон Imperative програмчлалын ялгаа
<https://codeburst.io/declarative-vs-imperative-programming-a8a7c93d9ad2>
- [3] Bootstrap Documentation,
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- [4] React Hook Form API Documentation,
<https://react-hook-form.com/docs>
- [5] About Git,
<https://git-scm.com/about/>
- [6] GitBook Documentation,
<https://docs.gitbook.com/>

А. УДИРДАГЧИЙН ҮНЭЛГЭЭ ДУГНЭЛТ

МУИС, МЭДЭЭЛЛИЙН ТЕХНОЛОГИИ, ЭЛЕКТРОНИКИЙН СУРГУУЛЬ МЭДЭЭЛЭЛ, КОМПЬЮТЕРЫН УХАНЫ ТЭНХИМ

Програм хамгийн Ангийн
оюутан Э. Тоболан -ийн

ҮЙЛДВЭРЛЭЛИЙН ДАДЛАГЫН АЖЛЫН УДИРДАГЧИЙН ТОДОРХОЙЛОЛТ

2023 оны 02 сарын 01

Үйлдвэрлэлийн дадлагын хариуцсан удирдагч М. Батчуул / Батчуул

Зураг A.1: Удирдагчийн үнэлгээ дүгнэлт