

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

Ганболдын Зандан-Очир

Дадлагын ажлын тайлан
(Internship report)

Мэдээллийн технологи (D061303)
Үйлдвэрлэлийн дадлагын тайлан

Улаанбаатар

2023 оны 09 сар

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

Дадлагын ажлын тайлан
(Internship report)

Мэдээллийн технологи (D061303)
Үйлдвэрлэлийн дадлагын тайлан

Удирдагч: _____ Др. Б.Сувдаа
Гүйцэтгэсэн: _____ Г.Зандан-Очир (20B1NUM1201)

Улаанбаатар
2023 оны 09 сар

Зохиогчийн баталгаа

Миний бие Ганболдын Зандан-Очир ”Дадлагын ажлын тайлан” 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
1. БАЙГУУЛЛАГЫН ТАНИЛЦУУЛГА	2
1.1 Байгууллагын ерөнхий танилцуулга	2
1.2 Үйл ажиллагаа	2
2. ДАДЛАГЫН АЖЛЫН ТӨЛӨВЛӨГӨӨ	3
3. ИЖИЛ СИСТЕМИЙН СУДАЛГАА	4
3.1 chewy.com	4
3.2 shoppy.mn	4
4. СИСТЕМИЙН ШААРДЛАГА	5
4.1 Хэрэглэгчийн шаардлага	5
5. АШИГЛАСАН ТЕХНОЛОГИ	6
5.1 Git/GitLab	6
5.2 React	6
5.3 Next.js	7
5.4 TypeScript	7
5.5 Mantine UI	8
5.6 i18next / next-i18next	9
6. ХЭРЭГЖҮҮЛЭЛТ	10
6.1 Төсөл эхлүүлэх	10
6.2 Нийтлэг component-ууд	11
6.3 Статик хуудаснууд	13
6.4 Динамик хуудаснууд	16
6.5 Бараа сагслах логик хийх	18
6.6 Хүргэлтийн форм	22
6.7 Төлбөр төлөлт	23

ГАРЧИГ

ГАРЧИГ

ДҮГНЭЛТ	27
НОМ ЗҮЙ	27
ХАВСРАЛТ	34
А. КОДЫН ХЭРЭГЖҮҮЛЭЛТ	35

ЗУРГИЙН ЖАГСААЛТ

6.1	Чиглүүлэгч component эцсийн байдлаар	13
6.2	Нүүр хуудас	28
6.3	Бидний тухай хуудас	29
6.4	Бүтээгдхүүний дэлгэрэнгүй хуудас	30
6.5	Ангилалын дэлгэрэнгүй хуудас	31
6.6	Case component эцсийн байдлаар	32
6.7	Хүргэлтийн форм эцсийн байдлаар	33

ХҮСНЭГТИЙН ЖАГСААЛТ

2.1	Дадлагын ажлын төлөвлөгөө	3
-----	---------------------------------	---

Кодын жагсаалт

5.1	TypeScript кодын жишээ (Алдаа өгнө)	8
5.2	JavaScript кодын жишээ (Төрөл шалгадаггүй учих алдаа өгөхгүй)	8
5.3	Англи хэлний орчуулгын файл	9
5.4	Монгол хэлний орчуулгын файл	9
5.5	Орчуулгаа DOM дотроо ашиглах нь	9
6.1	Вэбсайтны ашиглах 3 хэлийг нэмж өгсөн next-i18next санд зориулсан тохиргооны файл.	10
6.2	useRequest хүүкийг useApi нэрээр хэл солих үед туххйн хэлээр ахин дууддаг болгов	10
6.3	useApi хүүк ашиглан ангилалуудын датаг татав	11
6.4	Ангилалын датаг харуулж буй нь	12
6.5	Locale солих хэсэг	12
6.6	Хөдөлгөөний keyframe ууд	13
6.7	Keyframe ээ анимешн дотроо оруулж ирсэн нь	14
6.8	Youtube дээрх бичлэгийг вэб рүү embed хийн оруулж ирэх component	15
6.9	Ангилалын дэлгэргүйг SSR хийх нь	16
6.10	Сагстай холбоотой бүх үйлдэлийг хариуцах useCart хүүк	18
6.11	Ясны анимешн	20
6.12	Ясны анимешн CSS	21
6.13	Байгуулгын регистр шалгадаг component	22
6.14	Төлбөр төлөх хүсэлт явуулах код	23
6.15	Төлбөр төлөх хүсэлт явуулах код	25
A.1	Хэрэглэгч сагсанд бараа хийх хасах тоо хэмжээг өөрчилөх зэрэг сагстай холбоотой бүх үйлдэлийг хариуцах useCart хүүк	35
A.2	Бүтээгдэхүүний карт component	37

УДИРТГАЛ

Миний бие Г. Зандан-Очир нь үйлдвэрлэлийн дадлагыг систем хөгжүүлэлт, клауд консалтинг зэргээр үйл ажиллагаа явуулдаг **”Tech Partners ХХК”**-нд ахлах хөгжүүлэгч Д. Бат-Оргилын удирдлага дор веб фронт энд хөгжүүлэлт, эцсийн бүтээгдэхүүн хөгжүүлэн гаргах талаар суралцаж үүний дагуу **”Nomadic Pet Care LLC”** компанийн бараа бүтээгдэхүүн борлуулах Ecommerce системийн фронт-энд хөгжүүлэлтийг хийж гүйцэтгэлээ.

Зорилго: Фронт энд хөгжүүлэлт, хөгжүүлэлтийн арга барил мөн систем хөгжүүлэлтийн талаар бизнесийн түвшинд судлах

Зорилт: Удирдагчийн зааварчилгааны дагуу өгсөн үүрэг даалгаврыг цаг тухайд гүйцэтгэн хэрэгжүүлэлт хийх

1. БАЙГУУЛЛАГЫН ТАНИЛЦУУЛГА

1.1 Байгууллагын ерөнхий танилцуулга

Tech Partners ХХК нь 2020 оноос эхлэн Teso грүүпын хөрөнгө оруулалт аван **FIBO CLOUD** компаниас салбарлан Клауд консалтинг, Систем хөгжүүлэлтээр дагнан үйл ажиллагаа явуулж эхэлсэн. Tech Partners ХХК-ын хамт олон нь AWS, Kubernetes, Openstack, Private болон Hybrid cloud-н 40 гаран төсөл, мөн мобайл апп, үйлчилгээний газруудын захиалгын систем зэрэг платформыг хөгжүүлж амжилттай зах зээлд нэвтрүүлж байсан туршлагатай. Үүлэн технологийн чиглэлээр дагнаж зөвлөх үйлчилгээ, виртуалчлалын private болон public cloud бүтээгдэхүүн үйлчилгээ, сургалт, зөвлөгөөг мэргэжлийн өндөр түвшинд явуулдаг.

1.2 Үйл ажиллагаа

- **Клауд консалтинг үйлчилгээ**

AWS, Google Cloud, Azure зэрэг дэлхийн шилдэг клауд үйлчилгээг ашиглаж зардалаа багасган, найдвартай ажиллагааг нэмэгдүүлэх шийдлийг боловсруулж байгууллагын үйл ажиллагаанд нэвтрүүлэх.

- **Систем хөгжүүлэлт**

Харилцагчийн хүсэл шаардлагад нийцсэн шинэ бүтээгдэхүүн, систем, мобайл апп зэргийг орчин үеийн технологи, клауд шийдлүүд ашиглан богино хугацаанд, чанартай гүйцэтгэх

2. ДАДЛАГЫН АЖЛЫН ТӨЛӨВЛӨГӨӨ

Table 2.1: Дадлагын ажлын төлөвлөгөө

№	Гүйцэтгэх ажил	Хугацаа	Биелэлт	Дадлагын удирдагчийн үнэлгээ
1	Технологиудыг судлах	06/12		
2	Дизайнтай танилцаж, төсөлөө эхлүүлэх	06/13		
3	Гурван хэл дээр ажилдаг болгох	06/14 - 06/15		
4	Нүүр хуудас, Бидний тухай хуудас хийх	06/18 - 06/20		
5	Динамик бүтээгдхүүний дэлгэрэгүй, ангилалын дэлгэрэнгүй хуудас хийх	06/21 - 06/23		
6	Бараа сагслах логик хийх	06/25		
7	Хүргэлтийн мэдээлэл авах форм	06/26		
8	Төлбөр төлөх логик	06/27		
9	Тестенд өгөх, тестийн засвар, сайжруулалт хийх	06/28 - 06/30		

3. ИЖИЛ СИСТЕМИЙН СУДАЛГАА

Уг системийн фронт-эндийг хөгжүүлэхдээ React, Next.js зэрэг технологиудыг ашиглах чиглүүлэг авсан ба төстэй ажиллагаа болон технологи ашигласан дараах системүүдийг судлав.

3.1 chewy.com

Chewy.com бол гэрийн тэжээвэр амьтдын хоол болон бусад тэжээвэр амьтадтай холбоотой бүтээгдэхүүний чиглэлээр мэргэшсэн Америкийн онлайн жижиглэн худалдааны сайт юм. Уг сайт нь React, Next.js, Material UI ашиглан бүтээгдсэн интернэт хөтчийн санах ойг харахад мөн **i18n** сангаар орчуулгаа зохицуулдгийг харж болж байна. Ерөнхий бараа сагсагслах, бараа ангилах зэрэг төстэй үйлдлүүд хийгддэг тул хэрхэн бараа бүтээгдэхүүн харуулж байгаа сагсанд хийх үйлдэл зэрэгийг харж авав.

3.2 shoppy.mn

Захиалагч компани нь өмнө нь **shoppy.mn** дээр бараа бүтээгдэхүүнээ борлуулдаг байсан тул энэ системтэй төстэй байх хүлээлттэй байгаа гэж үзэн судлав. Технологийн хувьд React ашигласан мөн бараа бүтээгдэхүүн борлуулдаг Монголын томоохон худалдааны сайтуудын нэг юм. Хамгийн их анзаарсан зүйл нь вэбсайтны харилцан үйлдэл байлаа. Товч болгон hover хийх үед ямар нэгэн үйлдэл хийдэг, мөн өгөгдөл уншиж байх хооронд lazy loading хийж харуулдаг болохоор цэвэрхэн вэбсайтны хэсгүүд гэнэт хөдөлгөөн хийдээгүй болсон байна.

4. СИСТЕМИЙН ШААРДЛАГА

4.1 Хэрэглэгчийн шаардлага

4.1.1 Хэрэглэгчид

- Бараа бүтээгдэхүүн худалдан авах хэрэглэгчид

4.1.2 Функционал шаардлага

- Хэрэглэгч хэлээ солиж болдог байх
- Хэрэглэгч барааг төрлөөр жагсаалт хэлбэрээр хардаг байх
- Жагсаалтнаас бүтээгдхүүний нэр, богино тайлбар, үнэ харж болдог байх
- Бүтээгдэхүүний дэлгэрэнгүй харж болдог байх
- Хэрэглэгч сагсанд бараа хийх, хасах, тоо хэмжээг өөрчилж болдог байх
- Сагсанд хийсэн бараа дараа ороход байдаг байх
- Хүргэлтийн мэдээлэлээ хийж болдог байх
- Төлбөр төлөлт хийж болдог байх

4.1.3 Функционал бус шаардлага

- Сайт ямар ч төхөөрөмж дээр ажилдаг responsive байх
- Харилцан үйлдэл сайтай буюу хэрэглэгчид яг юу болж байгааг мэдээлэлдэг
- Ашиглахад ойлгомжтой байх
- Хүртээмжтэй байх

5. АШИГЛАСАН ТЕХНОЛОГИ

5.1 Git/GitLab

Git нь Version Control System буюу хувилбар удирдах систем ба ингэснээр хөгжүүлэлтийн явцад бүх өөрчлөлтөө хадгалах, багаар олон хүн нэг код base дээр ажиллах боломж бүрдүүлж өгдөг.

Git хостинг платформын тухайд **Tech Partners ХХК** нь нээлттэй эхийн GitLab-ийг өөрсдийн сервер дээр асаасан хэрэглэдэг ба Gitlab нь Git репозиторыг байршуулах, асуудлыг хянах, тасралтгүй нэгтгэх/тасралтгүй байршуулах (CI/CD) дамжуулах шугам, код шалгах хэрэгслүүд болон бусад олон зүйлсийг санал болгодог олон давуу талтай.

5.2 React

React¹ нь хэрэглэгчийн интерфейс бүтээхэд зориулагдсан нээлттэй эхийн JavaScript-ийн сан юм. React нь хэрэглэгчийн нарийн төвөгтэй интерфэйсийг бий болгохтой холбоотой хэд хэдэн асуудлыг шийдэж, дахин ашиглах боломжтой UI элементүүдийг бий болгох бүрэлдэхүүн хэсэгт суурилсан архитектурыг бий болгодог.

React-аар шийдсэн асуудлууд:

- **Бүрэлдэхүүн хэсгүүдэд суурилсан хөгжүүлэлт:** React нь бүрэлдэхүүн хэсэг дээр суурилсан архитектурыг дэмждэг бөгөөд UI элементүүдийг дахин ашиглах боломжтой, бүрэлдэхүүн хэсгүүдэд хуваадаг. Энэ нь том, нарийн төвөгтэй программуудыг жижиг, бие даасан хэсгүүдэд хуваах замаар удирдах, засвар үйлчилгээ хийхэд хялбар болгодог.
- **Virtual DOM:** React нь бодит DOM-ийн санах ойн дүрслэл болох Virtual DOM-той. Энэ нь DOM дотор зөвхөн өөрчлөгдсөн хэсгүүдийг шинэчлэх боломжийг олгодог бөгөөд

¹Reactjs official site <https://reactjs.org>

DOM-ийг бүтнээр бус хэсэгчилж шинчлэн хурд сайжруулдаг.

5.3 Next.js

Next.js² нь React дээр бүтээгдсэн фреймворк бөгөөд нэмэлт функцээр хангадаг ба веб программ бүтээхтэй холбоотой янз бүрийн асуудлыг шийддэг. Next.js-г React-тай ашиглах шалтгаанууд:

- **Сайжруулсан performance:** Next.js нь сервер талын дүрслэл (SSR) болон статик сайт үүсгэх (SSG) боломжтой болгож, программын эхний ачаалах хугацаа болон SEO-ийг сайжруулдаг.
- **Хялбаршуулсан routing:** Next.js нь энгийн бөгөөд ойлгомжтой routing системээр хангаснаар динамик болон үйлчлүүлэгч(client) талын навигациудыг хийхэд хялбар болгодог.
- **Автомат код хуваалт:** Next.js нь кодыг автоматаар жижиг багц болгон хувааж, зөвхөн тухайн хуудсанд шаардлагатай кодыг хүргэх замаар ачаалах хугацааг багасгадаг болгодог. Энэ нь илүү үр дүнтэй, хурдан ачаалагдах программыг бий болгодог.

5.4 TypeScript

TypeScript нь JavaScript-ийн статик хэлбэрээр бичигдсэн дээд багц бөгөөд хэд хэдэн давуу талыг санал болгодог.

- **Төрөл(Type) аюулгүй байдал:** TypeScript нь статик бичих үйлдлийг хэрэгжүүлдэг бөгөөд энэ нь та хувьсагчийн төрлүүд, функцийн параметрууд болон буцах утгуудыг зарлана гэсэн үг юм. Энэ нь хөрвүүлэх явцад төрөлтэй холбоотой алдааг олж илрүүлэхэд тусалж, ажиллах үеийн алдааны магадлалыг бууруулж, кодыг илүү бат бөх болгодог.
- **Хамтын ажиллагаа:** Олон хөгжүүлэгчидтэй төсөл дээр ажиллах үед TypeScript-ийн төрлийн тайлбарууд нь баримт бичгийн үүрэг гүйцэтгэдэг бөгөөд багийн гишүүд бие

²Next.js official site <https://nextjs.org>

биеийнхээ кодыг ойлгоход хялбар болгодог. Энэ нь илүү үр дүнтэй хамтын ажиллагаа болон дибаг хийх хугацааг багасгахад хүргэдэг. Мөн **TypeScript** нь сайн бүтэцтэй, маш сайн бичсэн API-г тодорхойлох боломжийг олгодог бөгөөд ингэснээр бусад хөгжүүлэгчид кодыг хэрхэн ашиглахыг ойлгоход хялбар болгож, буруугаар ашиглах эрсдэлийг бууруулдаг.

```
1 let num1: number = 5;
2 let num2: string = "10";
3
4 let result = num1 + num2;
```

Код 5.1: TypeScript кодын жишээ (Алдаа өгнө)

```
1 let num1 = 5;
2 let num2 = "10";
3
4 let result = num1 + num2;
```

Код 5.2: JavaScript кодын жишээ (Төрөл шалгадаггүй учих алдаа өгөхгүй)

5.5 Mantine UI

Mantine UI нь хүртээмжтэй байдал, хөгжүүлэгчийн туршлагад маш их ач холбогдол өгсөн нээлттэй эхийн React component сан юм. Үүнд 100 гаруй системдээ таруулан тохируулж болох component-той мөн веб сайтын хэрэглэгчийн интерфэйс бүтээх үйл ажиллагааг хялбарчлах use-disclosure, use-clipboard, use-local-storage, use-input-state, use-debounced-value гэх мэт React-ийн 50 гаруй хэрэгтэй хүүк агуулдаг.

Тэр дундаас use-local-storage буюу хэрэглэгчийн интернэт хөтчийн дотоод санах ойд өгөгдөл хадгалах хүүк нь хэрэглэгчийн сагсанд хийсэн бүтээгдэхүүн дараа орж ирэхэд байх гэдэг шаардлагыг биелүүлэхэд хэрэг болох юм.

5.6 i18next / next-i18next

i18next нь JavaScript хэл дээр бичигдсэн орчуулга удирдлагын систем юм. Орчуулгаа JSON файл хэлбэрээр хадгалж, тухайн хүний сонгосон хэлнээс хамаарч тохирсон орчуулгыг гаргаж ажилдаг ба маш хөнгөн бөгөөд бага кодоор орчуулгаа бүрэн шийдэх боломж олгодог.

next-i18next нь **i18next**-ийн миний ашиглах гэж буй **Next.js** санд тохируулан, уг сангийн router дотроо хэл хадгалах функц дээр тулгуурлан орчуулгаа удирддаг сан юм.

```
1 {  
2   "About Us": "About Us",  
3 }
```

Код 5.3: Англи хэлний орчуулгын файл

```
1 {  
2   "About Us": " ",  
3 }
```

Код 5.4: Монгол хэлний орчуулгын файл

```
1 import { useTranslation } from "react-i18next";  
2  
3 const AboutUsPage = () => {  
4   const { t } = useTranslation();  
5   return (  
6     <h1>  
7       {t("About Us")}  
8     </h1>  
9   )  
10 }
```

Код 5.5: Орчуулгаа DOM дотроо ашиглах нь

6. ХЭРЭГЖҮҮЛЭЛТ

6.1 Төсөл эхлүүлэх

Төслөө эхлүүлэхдээ хамгийн эхэнд хөгжүүлэлтийн орчноо бэлдэх зорилгоор **Next.js**-ийн **create-next-app** командаар шинэ төсөл эхлүүлж **Mantine UI**, **i18next**, **next-i18next** зэрэг сангуудыг нэмэн public фолдэр дотроо орчуулгын файлуудыг нэмэв.

Дараагаар системийн стайлыг хариуцах **Mantine UI** сангийн Provider дотор үндсэн өнгө, фонт зэргийг тохируулан, **next-i18next** сангийн Provider-ийг тохируулж, тохиргооны файлыг төсөлд нэмэв.

```
1 /** @type {import('next-i18next').UserConfig} */
2
3 module.exports = {
4   i18n: {
5     defaultLocale: "en",
6     locales: ["en", "mn", "cn"],
7   },
8 };

```

Код 6.1: Вэбсайтны ашиглах 3 хэлийг нэмж өгсөн next-i18next санд зориулсан тохиргооны файл.

Системийн өгөгдлийг REST API-аар авах ба хүсэлт дуудах явцыг ahooks сангийн useRequest хүүхийг ашиглан хийх тул төсөлд нэмж, уг хүүхийг хэл солигдох үед тухайн хэлийг хүсэлтийн толгойд хийн ахин хүсэлт явуулдаг болгон бичив.

```
1 import { useRequest } from "ahooks";
2 import { Options, Service } from "ahooks/lib/useRequest/src/types";
3 import { i18n } from "next-i18next";

```

```

4
5 function useApi<TData, TParams extends any[]>(
6   service: Service<TData, TParams>,
7   options?: Options<TData, TParams>
8 ) {
9   const api = useRequest(service, {
10     ...options,
11     refreshDeps: [i18n?.language, ...(options?.refreshDeps || [])],
12   });
13   return api;
14 }
15
16 export default useApi;

```

Код 6.2: useRequest хүүгийг useApi нэрээр хэл солих үед туххйн хэлээр ахин дууддаг болгов

6.2 Нийтлэг component-ууд

6.2.1 Чиглүүлэгч component

Бүх хуудасны толгой дээр байх чиглүүлэгч component дээр бүх ангилал меню болон орж ирэх ёстой.

```

1 const categoryListAPI = useApi(() => reference.getCategoryList({}), {
2   onError: (error) => {
3     Error(error.message);
4   },
5 });

```

Код 6.3: useApi хүүк ашиглан ангилалуудын датаг татав

```

1 const menus = categoryListAPI?.data
2   ?.sort((a, b) => {
3     return a.rank - b.rank;
4   })
5   ?.map((e) => {
6     return (
7       <Link
8         href={{
9           pathname: `/category/${e.id}`,
10          }}
11         onClick={() => closeDrawer()}
12         className={classes.link}
13         key={e.id}
14       >
15         {e.name}
16       </Link>
17     );
18   });

```

Код 6.4: Ангилалын датаг харуулж буй нь

Мөн хэл солих меню component-ийг хийв. Хэл солихдоо Next.js-ийн router дээр locale буюу хэл солих замаар ажиллана.

```

1 {data?.map((item) => (
2   <Menu.Item
3     key={item.value}
4     icon={
5       <Image src={item.image} width={32} height={18} alt={item.
        label} />

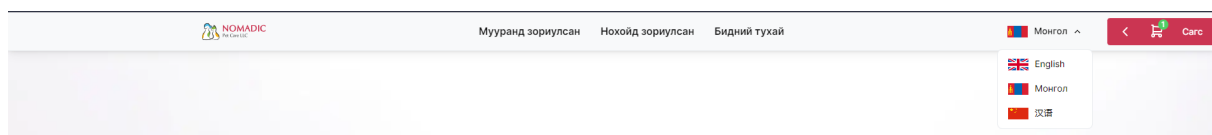
```

```

6      }
7      onClick={() => {
8          setSelected(item);
9          router.push(router.asPath, router.asPath, {
10             locale: item.value,
11          });
12      }}
13  >
14      {item.label}
15  </Menu.Item>
16  )))}

```

Код 6.5: Locale солих хэсэг



Зураг 6.1: Чиглүүлэгч component эцсийн байдлаар

6.3 Статик хуудаснууд

6.3.1 Нүүр хуудас

Нүүр хуудас дээр хамгийн эхэнд танилцуулгын хэсэг, дараа нь бүх ангилал түүний бүтэгдэхүүний жагсаалтуудыг харуулав.

Үүнд онцлох нь вэбсайтыг харилцан үйлдэл сайтай болгохын тулд нүүр хуудасны товчны icon дээр Mantine UI-ийн createStyles хүүк дээр CSS ашиглан дараах байдлаар хөдөлдөг анимешн хийв.

```

1  const jiggle = keyframes({

```

```

2      "0%": { rotate: "0deg" },
3      "5%": { rotate: "-10deg" },
4      "10%": { rotate: "10deg" },
5      "15%": { rotate: "0deg" },
6      "20%": { rotate: "10deg" },
7      "25%": { rotate: "-10deg" },
8      "30%": { rotate: "0deg" },
9      "35%": { rotate: "-10deg" },
10     "40%": { rotate: "10deg" },
11     "45%": { rotate: "0deg" },
12     "50%": { rotate: "-10deg" },
13     "60%": { rotate: "0deg" },
14     "70%": { rotate: "0deg" },
15     "80%": { rotate: "0deg" },
16     "90%": { rotate: "0deg" },
17     "100%": { rotate: "0deg" },
18   });

```

Код 6.6: Хөдөлгөөний keyframe ууд

```

1    ...
2
3    icon: {
4      transition: "transform 100ms ease",
5      rotate: "-45deg",
6
7      "&:hover": {
8        transform: "scale(1.2)",
9        animation: `${jiggle} 2s ease-in-out`,

```

```
10     },
11     },
12
13     ...
```

Код 6.7: Keyframe ээ анимешн дотроо оруулж ирсэн нь

6.3.2 Бидний тухай хуудас

Бидний тухай хуудас мөн адил back-end-ээс өгөгдлөө авчирч дизайны дагуу хийсэн ба онцлох зүйл нь Youtube дээрх бичлэгийг вэбсайт дээрээ харуулах хэсгийн дараах component-ээр шийдэв.

```
1 const YoutubeEmbed = ({ url }: { url: string }) => {
2   const id = extractVideoId(url);
3   return (
4     <div className="video-responsive">
5       <iframe
6         width="100%"
7         height="480"
8         src={`https://www.youtube.com/embed/${id}`}
9         frameBorder="0"
10        allow="accelerometer; autoplay; clipboard-write; encrypted-
11          media; gyroscope; picture-in-picture"
12        allowFullScreen
13        title="Embedded youtube"
14      />
15    </div>
16  );
17 };
```

```

17
18 function extractVideoId(url: string): string | null {
19     const regex = /(?:\/|\/%3D|v=|vi=)([0-9a-zA-Z_-]{11})(?:[%#?&]|$)/;
20     const match = url?.match(regex);
21     return match ? match[1] : null;
22 }
23
24 export default YoutubeEmbed;

```

Код 6.8: Youtube дээрх бичлэгийг вэб рүү embed хийн оруулж ирэх component

6.4 Динамик хуудаснууд

Өмнө дурдсанчлан Next.js санг ашиглах том шалтгаан нь Search Engine Optimization билээ. Хуудсаа хэрэглэгчийн интернэт хөтөч дээр биш сервер дээр дүрслэл(render) хийснээр хайлтын системийн бот crawl хийж хайлт сайжрах юм.

Дараах код нь бүтэгдэхүүний мэдээллийг болон орчуулгыг сервер дээрээ дуудан дүрслэл хийнэ. Бүтэгдэхүүний id-ийг дуудаж буй хуудаснаасаа буюу path-наас авах ба Next.js-ийн **GetServerSideProps** функц дээр сервер дээр хийгдэх бүх дата дуудалт мөн next - 118next сангийн **serverSideTranslations** функцээр серверийн орчуулгаа хийв.

```

1 ...
2
3 const CategoryDetailPage: FC<{ data: Category }> = ({ data }) => {
4     const { t } = useTranslation();
5     return (
6         <Page title={data?.name || t("Nomadic Pet Care LLC")}>
7             <CategoryDetail detail={data} />
8         </Page>

```



```

9   );
10  };
11
12  export const getServerSideProps: GetServerSideProps = async ({
13    query,
14    locale,
15  }) => {
16    const { id } = query as unknown as QueryProps;
17    try {
18      let all = await reference.getCategoryList({
19        locale: locale || "en",
20      });
21      let data = all.find((item) => item.id === Number(id));
22      if (!data) throw new Error("Category not found");
23      return {
24        props: {
25          data: data,
26          ...(await serverSideTranslations(locale || "en", ["common"])),
27        },
28      };
29    } catch (err) {
30      console.log("error", err);
31      return {
32        notFound: true,
33      };
34    }
35  };
36

```

```
37 export default CategoryDetailPage;
```

Код 6.9: Ангилалын дэлгэргүйг SSR хийх нь

6.5 Бараа сагслах логик хийх

6.5.1 *useCart* хүүк

Хэрэглэгч сагсанд бараа хийх, хасах, тоо хэмжээг өөрчлөх зэрэг сагстай холбоотой бүх үйлдлийг нэг аргаар, нэг сувгаар зохиулахын тулд дараах хүүкийг бичив.

Мөн хэрэглэгчийн сагсанд нь байгаа барааны мэдээллийг интернэт хөтчийн дотоод санах ойд хадгалснаар хуудас хооронд өгөгдөл дамжуулах, хэрэглэгч дараа хандахад бараанууд байх зэрэг хоёр асуудлыг шийдэж байгаа юм. Бүтэн кодыг хавсралт хэсгээс харна уу.

```
1  ...
2
3  export interface Item {
4    note?: string;
5    product_id: number;
6    qty: number;
7    product: Product;
8  }
9
10 const useCart = () => {
11   const { data: productList } = useApi(() => reference.getProductList
12     ({}));
13   const [cartItems, setCartItems] = useLocalStorage<Item[]>({
14     key: MyCartKey.items,
15     defaultValue: [],
16   });
```

```

16
17 const onQuantityChange = (id: number, qty: number) => {
18     const product = productList?.find((i) => i.id === id);
19     if (!product) {
20         notifications.show({
21             title: "Product not found",
22             message: "Please refresh the page",
23         });
24         return;
25     }
26     const item = cartItems.find((i) => i.product_id === id);
27     if (item) {
28         qty === 0
29             ? setCartItems(cartItems.filter((i) => i.product_id !== id))
30             : setCartItems(
31                 cartItems.map((i) => (i.product_id === id ? { ...i, qty } :
32                     i))
33             );
34     }
35     if (!item) {
36         setCartItems([...cartItems, { product_id: id, qty, product, note:
37             "" }]);
38     }
39     ...
40
41     return {

```

```

42     cartItems,
43     onQuantityChange,
44     getCartItem,
45     addToCart,
46     clearCart,
47     getItemCount,
48   };
49 };
50
51 export default useCart;

```

Код 6.10: Сагстай холбоотой бүх үйлдэлийг хариуцах useCart хүүк

6.5.2 Сагсны component

Дизайны дагуу сагсыг хажуу талаас гарч ирдэг байдлаар хийгээд useCart хүүкнээс одоо байгаа бараануудыг жагсаан харуулав. Бараа болгон өөр дээрээ тоог өөрчлөх, устгах цэстэй байгаа ба барааны богино мэдээлэл мөн үнийн дүнг харуулдаг болгов. Хамгийн доор нь нийт үнэ ба цааш үргэлжлүүлэх товч байна. Мөн барааны тоо өөрчлөгдөхөд хэрэглэгчийг урамшуулах үүднээс ясны icon-ууд санамсаргүй зүгт нисдэг анимешн хийж өгөв.

```

1 const renderFlyingBones = () => {
2   if (showFlyingBones) {
3     return (
4       <div className={classes.flyingBonesContainer}>
5         {Array.from({ length: 5 }, (_, index) => (
6           <IconBone
7             key={index}
8             style={{
9               position: "absolute",

```

```

10         top: 0,
11         left: 0,
12         animation: `fly-bone-${index} 2s linear`,
13         color: "#FBB71A",
14     }}
15     />
16     )))}
17 </div>
18 );
19 }
20 };

```

Код 6.11: Ясны анимешн

```

1 <style>
2     {Array.from(
3         { length: 5 },
4         (_, index) =>
5             `
6             @keyframes fly-bone-${index} {
7                 0% {
8                     transform: translate(0, 0);
9                 }
10                100% {
11                    transform: translate(${getRandomInt(
12                        -100,
13                        100
14                    })}px, ${getRandomInt(-100, 100)}px);
15                }

```

```

16         }
17         ,
18     )}
19 </style>

```

Код 6.12: Ясны анимешн CSS

6.6 Хүргэлтийн форм

Хүргэлтийн формыг ahooks сангийн **useForm** хүүкээр оролт баталгаажуулалтыг хийв. Онцлох зүйл нь хэрэглэгч баримтаа байгууллага дээр авахаар сонгосон тохиолдолд байгууллагын регистрээ хийх үед back-end луу тухайн байгууллага байдаг эсэхийг шалгадаг component-ийг дараах байдлаар шийдэв.

```

1  const orgAPI = useApi(reference.getOrg, {
2    manual: true,
3    onSuccess: (data) => {
4      if (data.found) {
5        setOrgName?.(data.name);
6        return;
7      }
8      setOrgName?.("Not Found");
9    },
10  });
11
12  useEffect(() => {
13    if (orgId && orgId.length === 7) {
14      orgAPI.run(orgId);
15    }
16    if (orgId && orgId.length !== 7) {

```

```

17     setOrgName?.("...");
18   }
19   // eslint-disable-next-line react-hooks/exhaustive-deps
20 }, [orgId]);
21
22 return (
23   <Input.Wrapper
24     ...
25   >
26     <Input
27       ...
28     />
29
30     {orgAPI.loading ? <Loader height={24} /> : <Text>{orgName}</
      Text>}
31
32   </Input.Wrapper>
33 );

```

Код 6.13: Байгуулгын регистр шалгадаг component

6.7 Төлбөр төлөлт

Хүргэлтийн форм бөглөж дууссаны дараа back-end луу төлбөр төлөх хүсэлт явуулах ба back-end дээр нэхэмжлэл үүсч хүсэлтэнд 200 буюу амжилттай гэсэн хариу ирвэл Social Pay үйлчилгээний төлбөр төлөх хуудасн руу хэрэглэгчийг чиглүүлнэ.

```

1     ...
2
3     const { run: create, loading: creating } = useApi(order.create, {

```

```

4      manual: true,
5      onSuccess: (data) => {
6          window.location.replace(
7              `https://ecommerce.golomtbank.com/socialpay/mn/${data.
8                  bank_invoice_id}`
9          );
10     },
11     onError: (error) => {
12         Error(error.message);
13     },
14
15     const onSubmit = (values: any) => {
16         create({
17             ...values,
18             district: districts?.find(
19                 (district) => district.id === Number(values.district)
20             )?.region_name,
21             khoroo: khoros?.find((khoros) => khoros.id === Number(
22                 values.khoroo))
23                 ?.region_name,
24             items: cartItems,
25             is_org: isOrg,
26             org_name: orgName,
27             phone: values.phone.toString(),
28         });
29     };
30
31     ...

```

Код 6.14: Төлбөр төлөх хүсэлт явуулах код

Хэрэглэгч төлбөрөө амжилттай төлбөл эсвэл амжилтгүй болвол уг сервис буцааж манай `order/detail` хуудас руу нэхэмжлэлийн дугаарыг араас залгуулан хэрэглэгчийг илгээх ба тухайн хуудас дээр төлбөр төлсөн үгүйг шалгаж амжилттай тохиолдолд нүүр хуудас руу чиглүүлнэ.

```
1  ...
2  const router = useRouter();
3  const { invoice } = router.query as { invoice: string };
4  const { clearCart } = useCart();
5  const {} = useApi(
6    () =>
7      order.check({
8        order_uuid: invoice,
9      }),
10   {
11     onSuccess: (data) => {
12       if (data.is_paid) {
13         clearCart();
14       }
15       router.push({
16         pathname: "/",
17         query: {
18           paid: data.is_paid,
19         },
20       });
21     },
22     ready: !!invoice,
```

```
23     }  
24   );  
25   ...
```

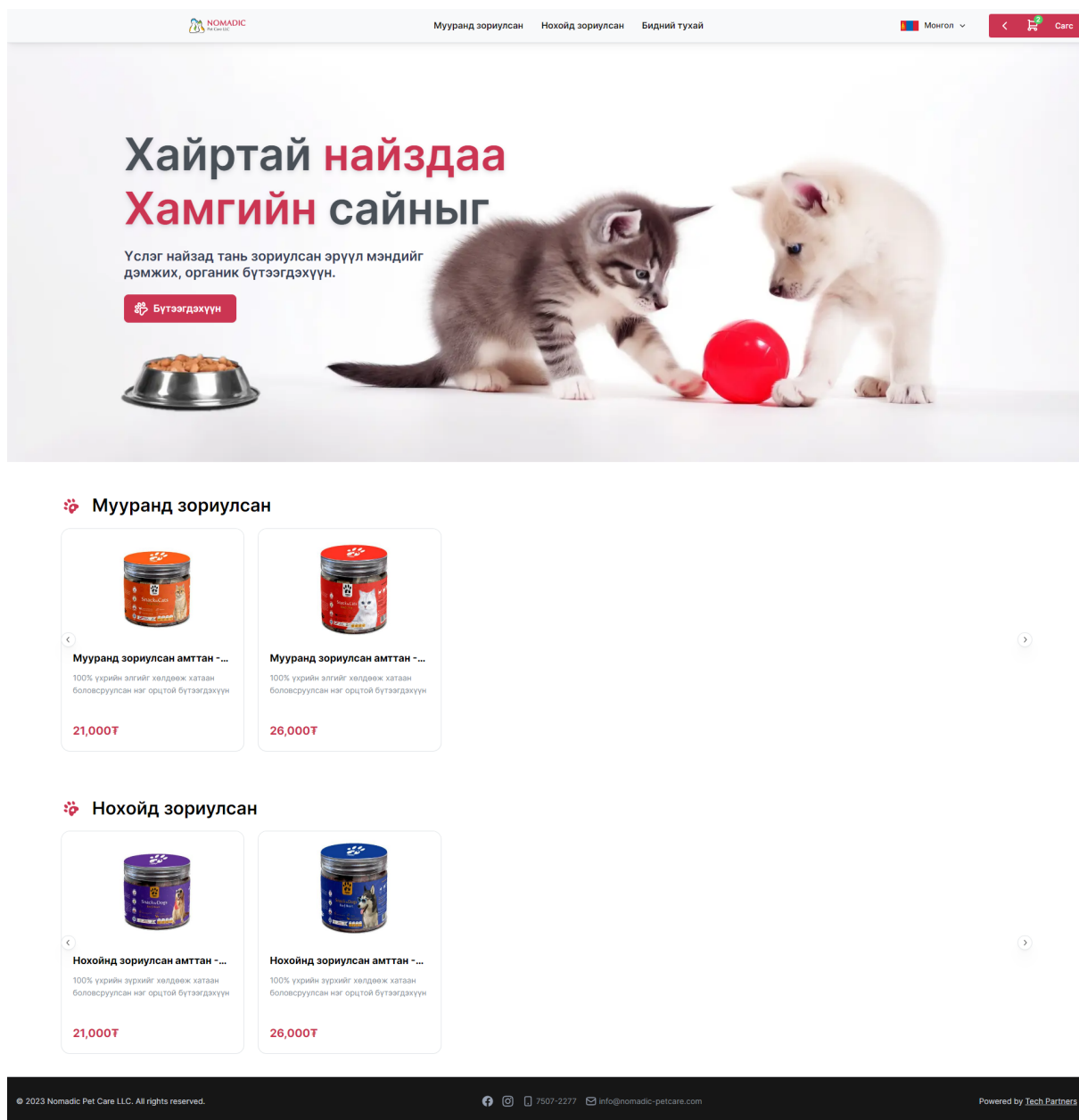
Код 6.15: Төлбөр төлөх хүсэлт явуулах код

Дүгнэлт

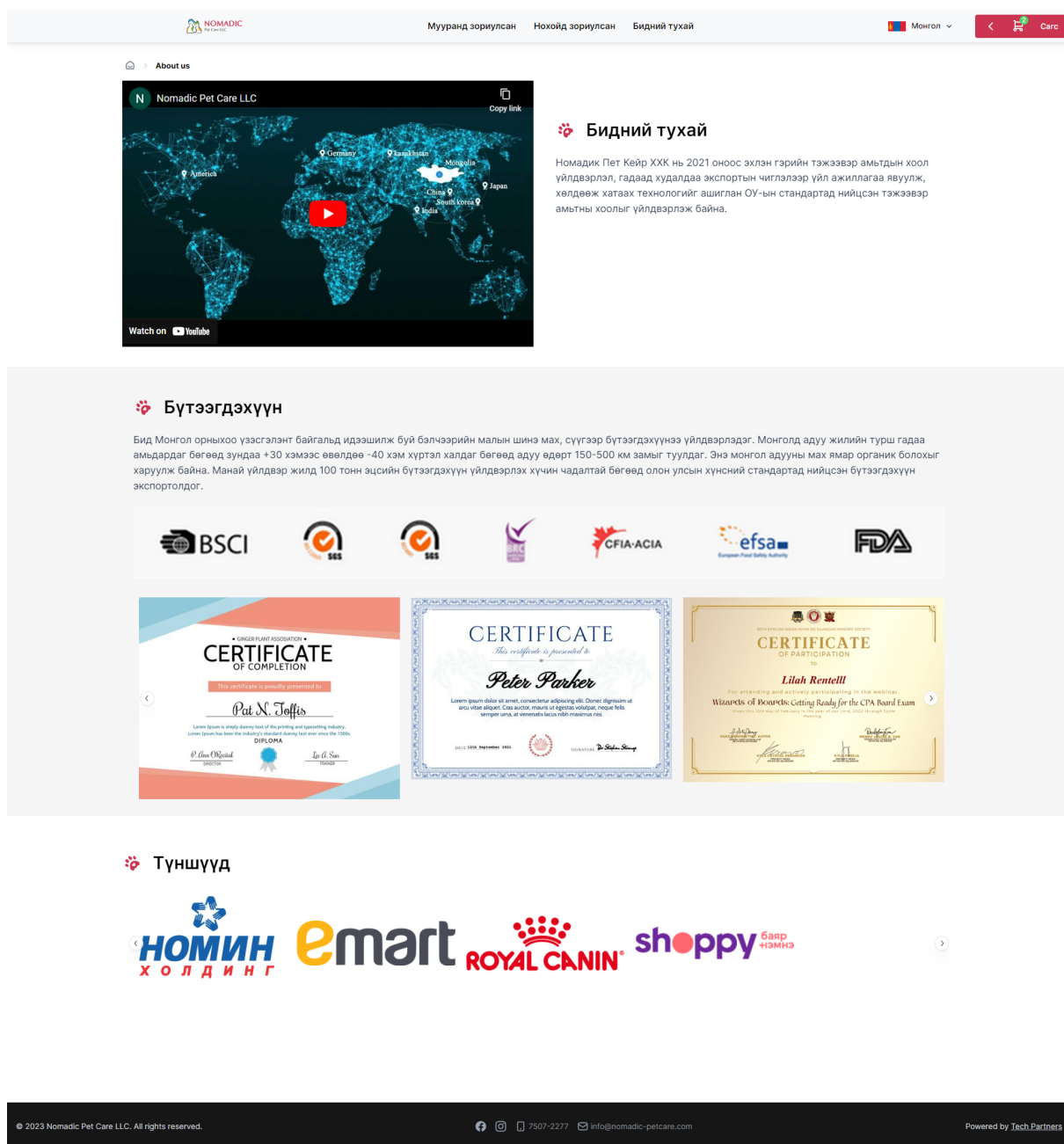
Миний бие Г. Зандан-Очир нь үйлдвэрлэлийн дадлагыг систем хөгжүүлэлт, клауд консалтинг зэргээр үйл ажиллагаа явуулдаг **”Tech Partners ХХК”**-нд ахлах хөгжүүлэгч Д. Бат-Оргилын удирдлага дор **”Nomadic Pet Care LLC”** компанийн бараа бүтээгдэхүүн борлуулах Ecommerce системийн фронт-энд хөгжүүлэлтийг хийж гүйцэтгэлээ. Дадлагын хугацаанд сервер дүрслэлт, динамик хуудас дүрслэл, цогц хүүк бичих, олон хэлээр API дуудах, JavaScript хэл дээр богино бөгөөд ойлгомжтой бичиглэл, CSS анимешн зэрэг Фронт-энд хөгжүүлэгчийн мэдэх шаардлагатай ур чадвар мэдлэгүүдийг олж авлаа. Мөн JavaScript хэлний эком систем дундаас хөгжүүлэгчийн туршлагыг сайжруулах, төвөгтэй үйлдлийг хөнгөвчлөх маш олон сан, фреймворкуудыг судалж хэрэглэж дадлаа.

Удирдагчаасаа хамгийн том сурсан зүйл нь асуудлыг инженер хүний нүдээр харж мэдэхгүй гэдэг дээр төвлөрөх биш асуудлаа ойлгож гарц гаргалгаа олдог, илүү сайн шийдэл судалж олж мэдэх, дэвшүүлдэг байх нь сайн инженерийг илтгэх шинж юм.

Цаашид фронт энд хөгжүүлэлтээ судлахаас гадна бусад back-end, Devops гээд хөгжүүлэлтийн бусад ойлголтуудыг судалж өгсөн үүрэг даалгавар, тулгарсан асуудал бүрийг шийдэж чадах инженер болох зорилготой байна.



Зураг 6.2: Нүүр хуудас



Зураг 6.3: Бидний тухай хуудас

Мууранд зориулсан

Нохойд зориулсан

Бидний тухай

Монгол

Care

Бүтээгдэхүүн

Нохойд зориулсан

Нохойнд зориулсан амттан - 60гр

21,000₮

1

Сагсанд нэмэх

Хоол боловсруулалтыг дэмжинэ

Үс уналтыг багасгана

Булчингийн үйл ажиллагааг дэмжин

Шүд, буйл бэхжүүлнэ

Монголд үйлдвэрлэв

Өөх Min	Эслэг Max	Чийг Max	Уураг Min	Калори 100g	Үс төрөгч Max
11.6%	4.98%	3.6%	72.2	433	23%

Орц найрлага:

Химийн нэмэлт орцгүй, 100% үхрийн зүхийг хөлдөөж хатаан боловсруулсан нэг орцтой бүтээгдэхүүн

Дэвүү тал:

- Үхрийн зүрх нь В витамин, магни, манган, кали г.м эрдсээр баялаг, өөх тосны агууламж багатай ба уургийн хэмжээ өндөртэй учир гэрийн тэжээвэр амьтны аминдам, уураг дутагдлыг нөхөж үс унах, шүд болон амны хөндийн өвчлөлийг бууруулж булчингийн сулралаас урьдчилан сэргийлнэ.
- Илүүдэл жинтэй, чихрийн шижин өвчтэй, харшилтай, ходоод гэдэсний өвчтэй, жирэмсэн болон дөнгөж төрсөн тэжээвэр амьтдад тохиромжтой.

Хэрэглэх заавар:

- 3 сараас дээш сартай амьтан хэрэглэхэд тохиромжтой

Нохойд зориулсан

Нохойнд зориулсан амттан...

100% үхрийн зүхийг хөлдөөж хатаан боловсруулсан нэг орцтой бүтээгдэхүүн

21,000₮

Нохойнд зориулсан амттан...

100% үхрийн зүхийг хөлдөөж хатаан боловсруулсан нэг орцтой бүтээгдэхүүн

26,000₮

© 2023 Nomadic Pet Care LLC. All rights reserved.

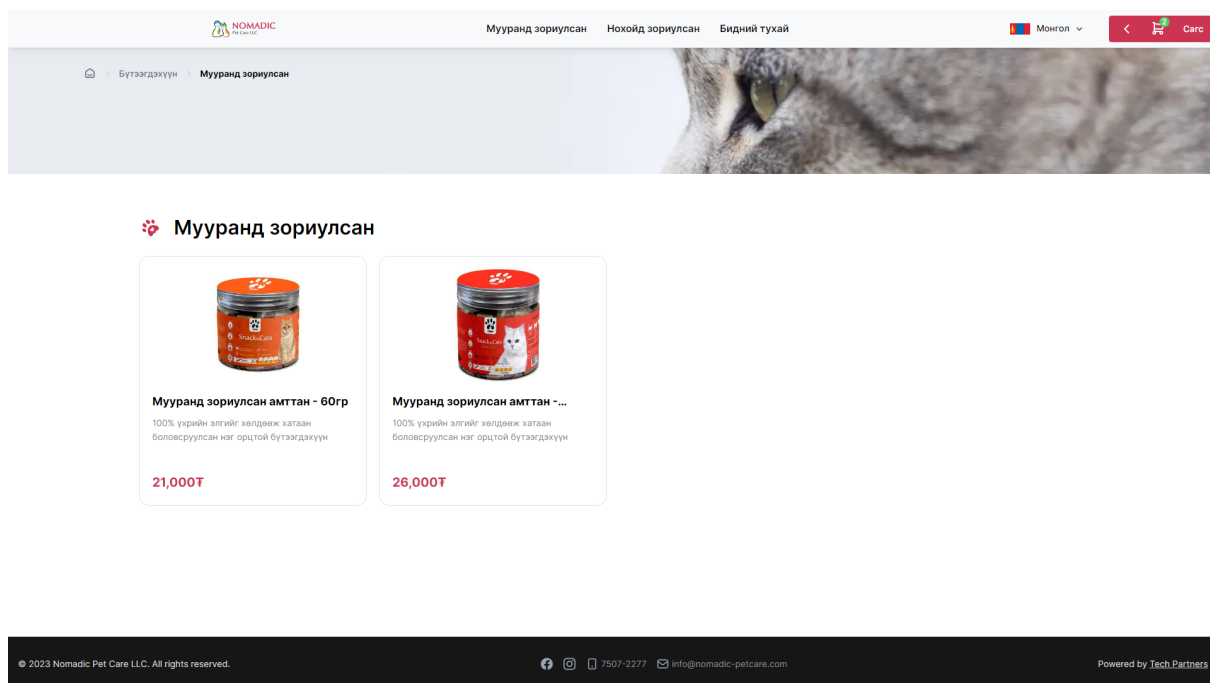
7507-2277

info@nomadic-petcare.com

Powered by TechPartners

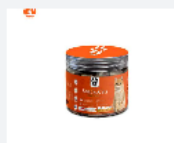
Зураг 6.4: Бүтээгдхүүний дэлгэрэнгүй хуудас

30



Зураг 6.5: Ангилалын дэлгэрэнгүй хуудас

Миний сагс



Snack for Cat - 60gram

100% organic fresh beef liver are processed by freeze-drying technology

- 1 +

Устгах

21,000₮



Нохойнд зориулсан амттан - 60гр

100% үхрийн зүрхийг хөлдөөж хатаан боловсруулсан нэг орцтой бүтээгдэхүүн

- 1 +

Устгах

21,000₮

Нийт

Activate Windows

42,000₮

Go to Settings to activate Windows.

Худалдан авах

Зураг 6.6: Cart component эцсийн байдлаар

< Хүргэлт

×

Дүүрэг
Pick one

Хороо
Pick one

Micro district
3, 4-р хороолол

Байр
6

Давхар
14

Тоот
6

Дэлгэрэнгүй хаяг
Address

Дугаар
99999999


Имайл
Enter email

VAT Type

Хувь хүн

Байгууллага

Payment Type

Social Pay
Голомт банк

Нийт

21,000₮

Баталгаажуулах

Зураг 6.7: Хүргэлтийн форм эцсийн байдлаар

Bibliography

- [1] React, <https://react.dev/>
- [2] Next.js, <https://nextjs.org/>
- [3] next-i18next, <https://github.com/i18next/next-i18next>

A. КОДЫН ХЭРЭГЖҮҮЛЭЛТ

```
1  ...
2
3  export interface Item {
4    note?: string;
5    product_id: number;
6    qty: number;
7    product: Product;
8  }
9
10 const useCart = () => {
11   const { data: productList } = useApi(() => reference.getProductList
12     ({}));
13   const [cartItems, setCartItems] = useLocalStorage<Item[]>({
14     key: MyCartKey.items,
15     defaultValue: [],
16   });
17
18   const onQuantityChange = (id: number, qty: number) => {
19     const product = productList?.find((i) => i.id === id);
20     if (!product) {
21       notifications.show({
22         title: "Product not found",
23         message: "Please refresh the page",
24       });
25       return;
26     }
27     const item = cartItems.find((i) => i.product_id === id);
28     if (item) {
29       qty === 0
30       ? setCartItems(cartItems.filter((i) => i.product_id !== id))
31       : setCartItems(
32         cartItems.map((i) => (i.product_id === id ? { ...i, qty } :
33           i))
34       );
35     }
36     if (!item) {
37       setCartItems([...cartItems, { product_id: id, qty, product, note:
38         "" }]);
39     }
40   };
41
42   const getCartItem = (id: number) => {
43     const product = productList?.find((i) => i.id === id);
44     if (!product) {
45       notifications.show({
46         title: "Product not found",
47         message: "Please refresh the page",
48       });
49     }
50   };
51 }
```

```

46     return;
47   }
48   const item = cartItems.find((i) => i.product_id === id);
49   if (item) {
50     return { ...item, product };
51   }
52   return undefined;
53 };
54
55 const addToCart = (id: number, qty: number) => {
56   const product = productList?.find((i) => i.id === id);
57   if (!product) {
58     notifications.show({
59       title: "Product not found",
60       message: "Please refresh the page",
61     });
62     return;
63   }
64   const item = cartItems.find((i) => i.product_id === id);
65   if (item) {
66     setCartItems(
67       cartItems.map((i) =>
68         i.product_id === id ? { ...i, qty: i.qty + qty } : i
69       )
70     );
71   }
72   if (!item) {
73     setCartItems([...cartItems, { product_id: id, qty, product, note:
74       "" }]);
75   }
76 };
77
78 const clearCart = () => {
79   setCartItems([]);
80 };
81
82 const getItemCount = () => {
83   return cartItems.reduce((acc, item) => acc + item.qty, 0);
84 };
85
86 return {
87   cartItems,
88   onQuantityChange,
89   getCartItem,
90   addToCart,
91   clearCart,
92   getItemCount,
93 };
94
95 export default useCart;

```

Код А.1: Хэрэглэгч сагсанд бараа хийх хасах тоо хэмжээг өөрчилөх зэрэг сагстай холбоотой бүх үйлдэлийг хариуцах useCart хүүк

```
1 import { Card, Image, Spoiler, Stack, Text, createStyles } from "  
  @mantine/core";  
2 import { useHover } from "@mantine/hooks";  
3 import Link from "next/link";  
4 import { FC } from "react";  
5 import { getImagePath, moneyFormat } from "../../utils";  
6  
7 const useStyles = createStyles((theme) => ({  
8  
9   card: {  
10     textDecoratation: "none",  
11     minWidth: "280px",  
12     "@media (max-width: 436px)": {  
13       minWidth: "100%",  
14       width: "100%",  
15  
16     },  
17  
18     transition: "box-shadow 150ms ease, transform 100ms ease",  
19     height: "100%",  
20  
21     "&:hover": {  
22       boxShadow: theme.shadows.md,  
23       transform: "scale(1.02)",  
24     },  
25   },  
26   description: {  
27     maxWidth: "100%",  
28     display: "-webkit-box",  
29     WebkitBoxOrient: "vertical",  
30     WebkitLineClamp: 3,  
31     overflow: "hidden",  
32     textOverflow: "ellipsis",  
33     "@media (max-width: 768px)": {  
34       fontSize: "11px",  
35       WebkitLineClamp: 2,  
36     },  
37   },  
38  
39   imageContainer: {  
40     height: "200px",  
41     overflow: "hidden",  
42     "@media (max-width: 768px)": {  
43       height: "180px",  
44     },  
45   },  
46   zoomImage: {
```

```

47     transform: "scale(1.1)",
48     transition: "transform 200ms ease",
49 },
50 title: {
51     display: "-webkit-box",
52     WebkitBoxOrient: "vertical",
53     WebkitLineClamp: 1,
54     overflow: "hidden",
55     textOverflow: "ellipsis",
56     fontSize: theme.fontSizes.lg,
57     "@media (max-width: 768px)": {
58         fontSize: "12px",
59         WebkitLineClamp: 1,
60     },
61 },
62 money: {
63     fontSize: theme.fontSizes.xl,
64     "@media (max-width: 768px)": {
65         fontSize: "14px",
66     },
67 },
68 }));
69
70 interface ProductCardProps {
71     title: string;
72     description: string;
73     price: number;
74     image: string;
75     url: string;
76 }
77
78 const ProductCard: FC<ProductCardProps> = ({
79     title,
80     description,
81     price,
82     image,
83     url,
84 }) => {
85     const { classes } = useStyles();
86     const { hovered, ref } = useHover();
87     return (
88         <Link
89             href={url}
90             style={{
91                 textDecoration: "none",
92             }}
93         >
94             <Card
95                 ref={ref}
96                 padding="lg"
97                 radius="lg"
98                 withBorder

```

```

99     className={classes.card}
100     onClick={(e) => {
101         e.stopPropagation();
102     }}
103 >
104     <Card.Section className={classes.imageContainer}>
105         <Image
106             src={getImagePath(image)}
107             height={200}
108             alt="cover"
109             className={hovered ? classes.zoomImage : undefined}
110         />
111     </Card.Section>
112
113     <Stack mt="md" justify="justify-between" spacing={0}>
114         <Text weight={700} mb="xs" className={classes.title}>
115             {title}
116         </Text>
117         <Spoiler
118             showLabel=""
119             hideLabel=""
120             maxHeight={90}
121             sx={{
122                 height: "90px",
123                 "@media (max-width: 768px)": {
124                     height: "50px",
125                 },
126             }}
127         >
128             <Text
129                 size="sm"
130                 color="dimmed"
131                 className={classes.description}
132                 dangerouslySetInnerHTML={{
133                     __html: description,
134                 }}
135             ></Text>
136         </Spoiler>
137         <Text color="#CB3552" weight={700} className={classes.money}>
138             {moneyFormat(price)}
139         </Text>
140     </Stack>
141 </Card>
142 </Link>
143 );
144 };
145
146 export default ProductCard;

```

Код А.2: Бүтээгдэхүүний карт component