ISSN 2181-3187

FOYDALANUVCHI INTERFEYSLARINI LOYIHALASHDA CSS PREPROCESSORLARINING (SASS, LESS) ROLI

To‘xtayeva Robiya Ravshanbek qizi
Andijon davlat universiteti, talaba
tukhtayeva_robiya@adu.uz

Anotatsiya: Ushbu maqola foydalanuvchi interfeyslarini loyihalashda CSS preprocessorlari, xususan Sass va LESS-ning rolini o‘rganishga bag‘ishlangan. CSS preprocessorlari nafaqat kodni strukturaviy va modular holga keltirishni, balki qayta foydalanish imkoniyatlarini oshirish, o‘zgaruvchilar, mixinlar va nesting kabi funksiyalar orqali murakkab dizaynlarni soddalashtirishni ta’minlaydi. Maqolada preprocessorlardan foydalanishning afzalliklari, ularning UX/UI dizaynga ta’siri hamda ishlab chiqish jarayonida vaqt va resurslarni tejashda o‘rni tahlil qilinadi. Zamonaviy frontend rivojlanish tendensiyalari kontekstida qanday qulayliklar yaratishi yoritiladi.

Kalit so‘zlar: Foydalanuvchi interfeysi dizayni, Stil kodini optimallashtirish, Kodni qayta foydalanish, Dynamic CSS, UX/UI dizayn, componentlar, web-sahifa

Foydalanuvchi interfeysi (UI) dizayni bugungi kunda veb-saytlar va mobil ilovalar muvaffaqiyatining asosiy omillaridan biri hisoblanadi. UI dizaynida CSS preprocessorlari, xususan Sass va LESS, samaradorlikni oshirish va murakkab dizaynlarni soddalashtirishda muhim rol o‘ynaydi. Ushbu maqolada CSS preprocessorlari haqida umumiy ma’lumot beriladi, ularning foydalanuvchilarga taqdim etadigan afzalliklari va dizayndagi o‘rni tahlil qilinadi.

CSS preprocessorlari - bu kengaytirilgan funksiyalar va qulayliklar taqdim etadigan vositalar bo‘lib, oddiy CSS kodni yanada samarali yozishga imkon beradi. Ular ishlab chiquvchilarga o‘zgaruvchilar, mixinlar, nesting (ichma-ich stil yozish), va boshqa ko‘plab funksiyalarni qo‘llash imkonini beradi. Natijada, murakkab interfeyslar uchun kod soddaroq, qayta foydalanishga yaroqli va o‘qishga qulay

ISSN 2181-3187

bo‘ladi. Shuningdek, preprocessorlardan foydalanish CSS yozish jarayonini avtomatlashtirishni ta’minlaydi. Ular orqali murakkab kodni qisqartirish va uni oson boshqarish mumkin. Bu esa, ayniqsa, katta hajmdagi loyihalarda ishlab chiquvchilarning vaqtini tejaydi.

Sass va LESS yordamida dizaynlarni modullarga bo‘lib, ular orasida tartibni saqlash va murakkab loyihalarda tartibni yo‘qotmaslikka erishiladi. Bu, ayniqsa, bir nechta ishlab chiquvchi birgalikda ishlaydigan loyihalarda juda muhimdir. Har bir komponent uchun alohida fayl yaratish orqali dizaynning barcha qismlari nazorat ostida bo‘ladi. O‘zgaruvchilar va mixinlar yordamida kodni bir marta yozib, uni bir necha joyda qo‘llash mumkin. Masalan, ranglar yoki shrift o‘lchamlari uchun o‘zgaruvchilar belgilab qo‘yiladi. Bu nafaqat vaqtni tejaydi, balki dizaynning yaxlitligini ham saqlaydi. Bir o‘zgaruvchini o‘zgartirish orqali butun loyiha dizayniga ta’sir ko‘rsatish imkonini beradi. Preprocessorlar avtomatlashtirilgan kodni yozish imkonini berib, ishlab chiquvchilarning vaqtini sezilarli darajada tejaydi. Har safar takrorlanadigan kod yozish o‘rniga, ishlab chiquvchi bir marta mixin yoki funksiyani yaratib, uni qayta-qayta ishlatishi mumkin.

Responsive dizayn yaratishda Sass va LESS media so‘rovlari bilan ishlashni osonlashtiradi. Stil kodlari tartibga solinadi va har bir ekran o‘lchami uchun alohida ko‘rinish yaratiladi. Misol uchun, ekran o‘lchamlariga qarab shrift o‘lchamlari yoki elementlarning joylashuvini osongina boshqarish mumkin. LESS va SASS yordamida brauzerlarga mos keladigan avtomatik prefikslarni qo‘shish orqali ishlab chiquvchilar uchun muhim vazifalarni soddalashtiradi. Prefikslarni qo‘lda yozish zarurati yo‘qoladi, bu esa vaqtni tejash va xatoliklarni kamaytiradi.

ISSN 2181-3187

```
@mixin responsive($breakpoint) {
  @media (max-width: $breakpoint) {
    @content;
  }
}

.container {
  width: 100%;
  @include responsive(768px) {
    width: 80%;
  }
}
```

Preprocessorlar nesting va mixin funksiyalari orqali murakkab interfeyslar uchun kodni qisqartiradi va yanada tozalaydi. Murakkab komponentlar ichma-ich yozish orqali birgalikda boshqariladi, bu esa, xatoliklarni kamaytiradi va o‘qiluvchanlikni oshiradi.

LESS va Sass avtomatik ravishda prefikslar qo‘shib, kodni barcha brauzerlarda moslashtirishga yordam beradi. Bu vositalar yordamida ishlab chiquvchilar har bir brauzer uchun alohida moslashtirilgan kod yozishga hojat qoldirmaydi. Kengaytirilgan dinamik CSS. o‘zgaruvchilar yordamida birgina parametrni o‘zgartirish orqali butun dizaynga ta’sir ko‘rsatish mumkin. Bu xususiyat loyihaning kelajakda kengayishiga moslashishni osonlashtiradi. Masalan, rang sxemasini o‘zgartirish zarurati tug‘ilganda, faqat asosiy rang o‘zgaruvchisiga o‘zgartirish kiritiladi va barcha bog‘liq elementlar avtomatik yangilanadi.

Ishlab chiqish jarayonini soddalashtirish: preprocessorlar kodning murakkab qismlarini avtomatlashtirish orqali ishlab chiqish jarayonini sezilarli darajada osonlashtiradi. Shuningdek, ular orqali ishlab chiquvchilar dizaynning texnik tafsilotlariga emas, balki interfeysning vizual jihatlariga ko‘proq e’tibor qaratishi mumkin bo‘ladi. LESS sintaksisi CSS ga juda yaqin bo‘lib, bu boshlovchi ishlab chiquvchilar uchun qulay. Sass esa ikki xil sintaksisni qo‘llab-quvvatlaydi. klassik Sass (odatiy o‘chirilgan qavsli sintaksis) va SCSS (CSS-ga o‘xshash sintaksis). SCSS yirik loyihalar uchun ko‘proq tanlanadi.

LESS o‘zgaruvchilarni belgilash uchun @ belgisi ishlatadi, masalan, @primary-color: #3498db;. Sass esa $ belgisi yordamida o‘zgaruvchilarni yaratadi, masalan, $primary-color: #3498db;.

LESS JavaScript asosida ishlaydi va brauzer ichida ham ishlashi mumkin. Sass esa Ruby asosida ishlab chiqilgan va oldin kompilyatsiyani talab qiladi, lekin tezlik va imkoniyatlar jihatidan boyroq.

CSS preprocessorlardan foydalanishning dolzarb holatlari

ISSN 2181-3187

Korporativ darajadagi yirik veb-loyihalarda Sass yoki LESS ishlatish dizaynni boshqarishni soddalashtiradi. Modularga asoslangan yondashuv kodni tartibli saqlashga yordam beradi. Sass va LESS vositalari yordamida bir xil kodni bir nechta qurilmalar uchun moslashuvchan qilish oson. Masalan, mixin orqali bir nechta ekran o‘lchamlari uchun stilni bir joyda belgilash mumkin.

Bugungi kunda CSS preprocessorlari o‘zining funksionalligi bilan ishlab chiquvchilarning eng ishonchli vositalaridan biri bo‘lib qolmoqda. Ko‘plab kompaniyalar o‘z mahsulotlarini ishlab chiqishda Sass yoki LESS-dan foydalanishni afzal ko‘radi. Katta jamoalar uchun kodni boshqarish qulayligi. Kelajakda dizaynni kengaytirish imkoniyati. Murakkab dizaynlarni sodda boshqarish va xatolarni kamaytirish. Mixins bir nechta qayta ishlatiladigan kodlarni yaratish imkonini beradi. Bu nafaqat kodni qisqartiradi, balki bir xil stilni turli elementlarga bir marta yozib qo‘llashga imkon beradi.

Masalan:

CSS preprocessorlari foydalanuvchi interfeysi dizaynida ishlab chiquvchilarning ishi samaradorligini oshirishda katta rol o‘ynaydi. Sass va LESS yordamida interfeyslarni nafaqat chiroyli, balki samarali tarzda ishlab chiqish mumkin. Preprocessorlar kodni strukturaviy, qayta foydalanishga yaroqli va optimallashtirilgan holga keltiradi. Zamonaviy frontend rivojlanish tendensiyalari va avtomatlashtirilgan ishlab chiqish jarayonida CSS preprocessorlari muhim vosita sifatida o‘z sahamiyatini saqlab kelmoqda. Shunday qilib, murakkab interfeyslar bilan ishlashda bu vositalarni qo‘llash nafaqat qulaylik, balki zaruratga aylangan.

Foydalanilgan adabiyotlar:

ISSN 2181-3187

1. React Native Documentation – React Native rasmiy hujjatlari. Onlayn manba
2. Expo Documentation – Expo rasmiy hujjatlari, React Native uchun rivojlantirish muhitini taqdim etuvchi platforma. Onlayn manba
3. Brown, A. (2021). React Native Cookbook: Bringing the Web to Native Platforms. Packt Publishing.
4. Chinnathambi, A. (2020). React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Applications. Apress.
5. Kumar, N. (2019). Mastering React Native. Packt Publishing.