

Studentnamn: Abdulaziz Turk

Avancerad webbutveckling med HTML och CSS

Innehåll

Introduktion	1
Bootstrap	1
1. Fördelar:	2
2. Jämförelse:	2
Projektet:	3
1. index.html	3
2. form.html.....	5
3. login.html.....	5
4. register.html.....	5
5. team.html.....	5
6. style.css	5
Slutsats	5
Referenser	6

Introduktion

Att skapa en fullständig webbsida med hjälp av ett ramverk är målet för denna inlämningsuppgift. Jag kommer att undersöka ramverkets olika funktioner, komponenter, layoutalternativ, formulär och mycket mer. Samtidigt kommer jag att tänka på de val jag gjorde under utvecklingen.

Bootstrap

Bootstrap är en front-end ramverk som använder CSS och JS komponenter för att bygga webbsidor på ett snabbt och responsivt sätt.

Jag valde att bygga min webbsida med Bootstrap med anledning av att Bootstrap har många fördelar jämfört med andra ramverk som Foundation, Bulma och Tailwind.

1. Fördelar:

- Bootstrap är okomplicerat och lätt att använda och förstå med anledning av dess omfattande bibliotek som har förbyggda komponenter och välorganiserad dokumentation.
- Bootstrap har olika tillägg som Sass och Less, detta gör utveckling lättare med tanke om man vill använda sina egna teman och komponenter.
- Bootstrap är ständigt uppdateras och underhålls med tanke på att nya funktioner och uppdateringar läggs ofta till det.
- Bootstrap erbjuder ett ramverk för användargränssnitt som följer uppsatta principer, vilket gör att skapa en webbsida som ser modern ut båda enkelt och effektivt.

2. Jämförelse:

CSS-ramverken Bootstrap, Tailwind, Bulma och Foundation hjälper och underlättar utvecklarens jobb för att skapa professionella och responsiva webbplatser, de ramverken dock skiljer sig något i egenskaper och tillvägagångssätt.

Bootstrap har flera fördelar jämfört med andra ramverk, vilket gör att jag föredrar Bootstrap framför andra.

- Bootstrap är välkänt utvecklingsramverk eftersom det inkluderar en stor samling av komponenter och erbjuder ett stort utbud av färdiga element som knappar, formulär och många andra. Dessa element kan snabbt och enkelt inkluderas i projektet vilket minskar utvecklingstid. Utöver det som nämnts ovan delar Bootstrap-sidan upp sig i tolv kolumner, vilket gör implementeringen av innehållsplacering och layout enkel och effektiv.
- Tailwind är en CSS-ramverk som erbjuder ett brett utbud av lågnivåvertygklasser, dessa klasser kan läggas till HTML-komponenter vilket ger utvecklaren finjustera utseendet. Tailwinds tillvägagångssätt skiljer sig från Bootstrap genom att den inte inkluderar färdiga komponenter. Därför valde jag att inte använda Tailwind.
- Bulma har ett minimalistiskt CSS-ramverk med ett begränsat antal komponenter och uteslutning av JavaScript. Det var därför jag valde att inte använda Bulma, trots att det är enklare än Bootstrap men delar samma grid-struktur.
- Foundation är ett CSS-ramverk som erbjuder färdiga komponenter, en responsiv grid-struktur och typografi. Det är känt för sin flexibilitet och passar perfekt för storskaliga projekt. Därför valde jag att inte använda Foundation.

Med tanke på allt ovanstående valde jag att bygga min webbplats med hjälp av Bootstrap. Det innehåller ett brett utbud av färdiga komponenter, ett responsivt grid-system och omfattande dokumentation. Det gör det till ett utmärkt val för alla utvecklare som söker ett välstött CSS-ramverk.

Projektet:

I denna del av analysen kommer jag att beskriva hur min webbplats är byggd med hjälp av ramverket Bootstrap. Jag kommer att förklara mer om mina val av klasser för att skapa bättre förståelse för läsaren om vad varje klass innebär och varför den används. Samtidigt vill jag ge översikt av hur koden är strukturerad.

Projektet består av fem HTML-filer och en CSS-fil. Jag kommer att ge en mer detaljerad förklaring av innehållet på varje sida.

1. index.html

Index är startsidan för webbplatsen och består av navbar, landing, features, our work, our blog, subscribe och footer. Nedan kommer jag att förklara detaljerat om varje sektion och vilka klasser de innehåller.

- `<head>` först och främst surfade jag till <https://getbootstrap.com/> och <https://fontawesome.com> och laddade ner Bootstrap-filen och fontawesome-filen. Sedan länkade jag till de filerna som jag behövde använda i min kod. Nästa steg var att välja mitt typsnitt från Google fonts och länka till det i min kod.
- `<body>` det består av sju huvuddelar enligt följande:
 - Navbar `<nav>`: det är en tagg för navbar och det innehåller olika Bootstrap-klasser som:
 - ``navbar``: definierar elementen som utgör en navbar.
 - ``navbar-expand-lg``: ange att navbar kommer att expandera till en stor navbar när skärmen är ``lg`` eller större.
 - ``sticky-top``: klassen håller fast navbar högst upp på sidan.
 - ``container``: definierar en behållare runt elementen och ger padding runt innehållet.
 - ``navbar-brand``: ger logotyp en viss stil för att se bättre ut.
 - ``navbar-toggler``: klassen hjälper att dölja länkarna när skärmen är liten.
 - ``collapse``: definierar att elementen ska kollapsa och expandera vid behov.
 - ``navbar-collapse``: definierar ett kollapsbart navigeringsfält.
 - ``navbar-nav``: definierar navbar-länkar.
 - ``ms-auto``: används för att placera länkar till höger i behållaren.
 - ``mb-2 mb-lg-0``: ger lite marginal på små skärmar och tar bort marginal på stora skärmar.
 - ``nav-item``: definierar objekt för varje element i listan.
 - ``nav-link``: definierar utseende för länkar.
 - ``p-2 p-lg-3``: ger lite padding på små skärmar och mer marginal på stora skärmar.
 - ``active``: ange vilken länk som är aktiv.
 - ``search``: definierar stilen för sökrutan.
 - ``ps-3 pe-3``: ger padding från höger och vänster sidan.
 - ``rounded-pill``: ger en rundad, pillformad design.
 - Landing: det är en hero sektion, som innehåller några element som har Bootstrap-klasser som:
 - ``landing``: det är en egen klass som jag anpassade stilen med hjälp av CSS.
 - ``d-flex``: en Bootstrap-klass som tillämpar flexbox.

- ``justify-content-center align-items-center``: två klasser som centrerar innehållet i mitten.
 - ``text-center``: centrerar innehållet horisontellt.
 - ``text-light``: ändrar textfärgen till ljus.
 - ``fs-6``: ändrar fontstorleken till 6.
 - ``text-white-50``: ändrar textfärgen till vit med 50% opacitet.
 - ``mb-5``: ger en margin-bottom på 5.
 - ``btn``: en klass som definierar att detta element är en knapp.
- Features: en sektion som beskriver syftet av denna webbplats och vilka funktioner tillhandhålls, här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``pt-5 pb-5``: lägger till padding för topp och botten.
 - ``mt-5 mb-5``: lägger till marginal för topp och botten.
 - ``position-relative``: positionerar element relativt till dess normala position.
 - ``text-black-50``: ändrar textfärgen till svart med 50% opacitet.
 - ``text-uppercase``: ändrar texten till versaler.
 - ``row``: skapar en rad för att placera kolumner.
 - ``col-md-6 col-lg-4``: definierar en kolumn som tar en tredjedel av bredden på stora skärmar och hälften på medium skärmar.
 - ``position-absolute bottom-0``: definierar ikonerna som ska vara längst ner i sin behållare.
 - ``lh-1``: justerar radhöjden.
 - Our work: en sektion som visar tidigare arbete. här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``bg-white``: ändrar bakgrundsfärgen till vit.
 - ``img-fluid``: gör att bilden fyller sin div-element.
 - Blog: en sektion visar viktiga blogg. här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``card``: definierar ett kort som innehåller kort information.
 - ``card-img-top``: placerar bilden längst upp på kortet.
 - ``card-body``: innehåller huvudinnehållet inom kortet.
 - Subscribe: en sektion för premierationsdelen. här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``text-center text-md-start``: centrerar innehållet på små skärmar och börjar det till vänster på större skärmar.
 - ``fw-bold``: gör texten fet.
 - ``fs-5``: ändrar textstorleken.
 - ``w-100``: sätter bredden på input-fältet till 100%.
 - ``bg-transparent``: sätter bakgrundsfärgen till transparent.
 - Footer: en sektion för footer delen. här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``gap-5``: skapar mellanrum mellan varje element.
 - ``list-unstyled``: tar bort liststilar.
 - ``link-underline``: visar en linje under länken.
 - ``link-underline-opacity-0``: ändrar opacity till 0.

2. form.html

det är kontaktsida möjliggör kontakt mellan användare och företaget. Sidan består av navbar, contact-form och footer. Nedan kommer jag att förklara detaljerad om contact-form och nya klasserna som används i det.

- contact-form: här är några klasser som används i denna sektion som inte förklarades tidigare:
 - ``form``: indikerar att detta är ett formulär.
 - ``form-label``: ger formatering för en label.
 - ``form-control``: ger ett standardinputfält för formulär.
 - ``d-grid``: tillämpar grid för att kunna placera knappar.

3. login.html

loggingsidan hjälper användaren att loggar in på sitt konto och består av navbar, login-form och footer.

4. register.html

Registersidan hjälper användaren att skapa ett konto och består av navbar, contact-form och footer.

5. team.html

Teamsidan visar anställda på företaget och deras befattningar, och består av navbar, team och footer.

6. style.css

I CSS-filen, har jag deklarerat nya färger som jag anser passar bäst för min sida. Trots att CSS-filen kan se lång ut, används den främst för att definiera färger.

Slutsats

Slutligen har alla ramverk som Bootstrap, Tailwind, Bulma och Foundation visat sig vara avgörande verktyg för webbutveckling, och de har gjort utvecklingens arbete mycket lättare. Var och en av dessa ramverk har sina egna fördelar och funktioner för att effektivt organisera och strukturera koden. Genom att noggrant överväga deras respektive för- och nackdelar kan utvecklare fatta välgrundade beslut när de väljer den mest lämpliga ramverk för deras specifika projekt.

Bootstrap har visat sig att det är effektivt ramverk. Genom att skriva Bootstrap klasser i min kod har jag inte bara sett förbättringar i prestanda och effektivitet på mina webbsidor, utan jag har också förstått betydelsen av att skriva kod som är välstrukturerad.

Referenser

- Ohlsson, Christian / HV. (1, 2024). *AWTH 2024 5.3 Bootstrap demo*. [Video]. Youtube. https://www.youtube.com/watch?v=2E_cgPvSX54&ab_channel=HVChristianOhlsson .
- Ohlsson, Christian / HV. (1, 2024). *AWTH 2024 5.4 Foundation demo*. [Video]. Youtube. https://www.youtube.com/watch?v=tYZN21wVLF8&ab_channel=HVChristianOhlsson .
- Ohlsson, Christian / HV. (1, 2024). *AWTH 2024 5.5 Bulma demo*. [Video]. Youtube. https://www.youtube.com/watch?v=LFUwy7GkxI8&ab_channel=HVChristianOhlsson .
- Ekwuno, O. (2022). Comparing Tailwind CSS to Bootstrap: Is it time to ditch UI kits?. <https://blog.logrocket.com/comparing-tailwind-css-bootstrap-time-ditch-ui-kits/> .
- Amaechi, A. (2022). Bulma vs. Tailwind CSS: Which is the better Bootstrap alternative?. <https://blog.logrocket.com/bulma-vs-tailwind-css-better-bootstrap-alternative/> .
- Njiru, P. (2023). Tailwind CSS vs. Bootstrap vs. Bulma vs. Semantic UI vs. Material-UI: A Comparison of Popular CSS Frameworks. <https://pnjiru.medium.com/tailwind-css-vs-eecfd72f389> .