

Projektplanering

1. Inledning

Kort beskrivning av projekt

jag ska göra en startside för olika webbläsare. På den så ska det finnas en överblick över dagen, såsom nyheter, temperatur och kanske google calendar och andra viktiga saker, som kommer fungera som widgetar som namn kan ändra på via en inställningsmeny. jag kommer cacha sparad innehåll, och man kommer kunna spara artiklar som poppar upp på sidan.

Beskrivning av ev kund

eftersom sidan kommer vara som en "hem"-sida för webbläsaren så kommer målgruppen vara personer som primärt använder sig av datorer. åldersgruppen kan vara ganska bred, men användare borde vara intresserade av att läsa av temperatur och nyheter, (sidan kommer alltså inte vara för 3 åringar), och personen som använder sidan behöver också kunna va lite "tech-savy", för att kunna gå in i inställningar och ändra utseende på sidan. (Alltså inte din average 90-plussare.)

Vilket betyg satsar du på

Eftersom sidan jag kommer göra är komplicerad och använder sig av många olika funktioner, samt är responsive på olika sätt, så kommer mitt projekt att få ett A.

2. Ingående beskrivning av projektet

Beskrivningen

Målet med detta projekt är att skapa en anpassningsbar startside för webbläsare som ger användarna en översikt över sin dagliga information på ett användarvänligt och personligt sätt. Startsidan kommer att fungera som en central plats där användarna kan få åtkomst till olika widgetar som visar relevant information, inklusive nyheter, väder, Google Calendar och andra viktiga uppgifter. Nedan följer en mer detaljerad beskrivning av projektet:

1. Anpassningsbara widgetar: Användarna kommer att ha möjlighet att lägga till, ta bort eller ordna om olika widgetar på startsidan. Varje widget kommer att vara en separat sektion på sidan som visar specifik information. De kommer också att kunna ändra inställningar för varje widget via en inställningsmeny, som kan omfatta val som nyhetskällor, plats för väderinformation och kalenderhändelser.
2. Dagens översikt: Startsidan kommer att erbjuda en snabb överblick över användarens dag. Det kan inkludera aktuella nyheter, väderrapport för användarens plats och kommande evenemang från deras Google Calendar.
3. Anpassningsbar graf: Temperatur Informationen som kommer att finnas på sidan kommer visas i en graf. I grafen ska man kunna lägga till olika platser temperatur för att jämföra väder, och välja vilka tidsperioder som grafen visar. det här kommer göras dynamiskt, och grafen kommer att vara en separat sak som man kan integrera lätt på andra sidor.
4. Cachad information: För att förbättra prestanda och minska belastningen på externa servrar kommer projektet att använda LocalStorage för att lagra och cacha tidigare hämtad information. Detta gör att användarna kan snabbt ladda sidan och få tillgång till tidigare hämtad information även när de är offline.

5. Spara artiklar: Användarna kommer att ha möjlighet att spara artiklar från nyhets widgeten för senare läsning. Sparade artiklar kommer att vara tillgängliga på startsidan för snabb åtkomst.
6. API-integration: Projektet kommer att integrera olika API:er för att hämta information från externa källor. Till exempel kan News API användas för nyheter och OpenWeatherMap API för väderinformation. Google Calendar API kommer att användas för att hämta kalenderhändelser.
7. Responsiv design: Startsidan kommer att vara responsiv och fungera bra på olika enheter, inklusive datorer, surfplattor och smartphones.
8. Användarvänligt gränssnitt: Gränssnittet kommer att utformas med fokus på användarvänlighet och enkel navigering. Användarna kommer att kunna anpassa sin startsida enligt sina egna preferenser och behov.

Lista allt som skall göras för att projektet skall bli till

1. hämta temp från API
 - jag kommer använda mig av openweathermap
 - API-key kan bli intressant
2. hämta nyheter från API
 - nyheter kommer från newsapi.org
 - API-key kan bli intressant
3. välj land och plats för både temp och nyheter.
 - Inställning kommer finnas i inställningsmeny (separat sida)
 - plats kommer behöva sparas
 - när sidan först laddas tar vi pos av geolocation. informationen sparas ner.
 - ifall informationen inte finns eller om användare klickar på "refresh" knapp för position så tar vi pos igen.
 - både automatiskt plats-tillval och manuellt.
4. attraktivt sätt att visa nuvarande temp på startsidan
 - detalj-text ska man kunna ändra ordning och info på via dropdown.
 - val mellan olika enheter
5. dynamisk nyhets-display
6. anpassningsbar graf för temperatur
7. extra tillval för graf
8. inställningsmeny
9. teman för webbsidan
 - mörk / ljust
 - accentfärg
10. dynamisk bakgrund beroende på väder
11. statisk bakgrund, antingen färg eller bild
12. colorpicker för bakgrunden
13. widget-builder där all placering på graf, huvud-temp och nyheter ändras.

Beskriv eventuella undersider webbplatsen kommer att använda sig av samt andra speciella saker, script osv.

1. inställnings-sida

- man kommer in på sidan via ett kugghjul högst upp i högra hörnet på huvudsidan.
- sidan kommer vara en scrollmeny med olika inställningar
- i inställningarna kommer det att vara en "website builder", där man kan lägga till och ordna om element som ska finnas på startsidan. det kommer också att gå att customize elementen och lägga dem tillsammans med varandra.

2. get_data()

- jag kommer skriva en funktion som callar openweathermap och sparar ner json filen till datorn via localStorage.
- jag väljer att spara ner till localStorage för att slippa vänta på fetch-requesten varje gång man öppnar sidan.
- för att få väder info så måste jag också få användarens position. för det så skriver jag en annan funktion som också sparar ner platsen i localStorage.
- när vi har sparat ner vår nya data så vill vi berätta det för resten av scripten. eftersom vi har våra funktioner i olika dokument så kan man inte bara använda sig av en function call eller variabel. istället så använder vi oss av ett custom event, och event listeners i de andra script-sidorna.

3. display_info()

- när vi fått väder information så måste vi visa den för användaren. för att veta vart vi ska lägga info så har vi skapat olika divs i html filen, med olika klasser. vi tar divsen med querySelect(), och skapar nya element med vår information.

4. draw_graph()

- vi vill visa upp temperaturen med en graf.
- i funktionen loopar vi igenom all temp data och ritar linjer med position i listan med temperaturen, och temperaturen.

Tekniker jag behöver lära mig för att utveckla webbsidan

För att strukturera koden på ett smidigt sätt behöver jag lära mig hur man använder custom events för att veta när vissa delar av min kod har körts. jag har valt den här tekniken för att jag har flera olika filer för min javascript för att göra det mer lättöfläsligt. för att kommunicera via filer så är events det bästa sättet. andra alternativ skulle vara att ändra DOM content på sidan, och ha event listeners som lyssnar på ändringar. det här är däremot inte en lika snygg lösning som användandet av custom events.

Jag behöver också bli bättre på att använda localStorage. Jag har använt mig av det en gång tidigare till slutprojekt i webbdev1, men då använde jag det på det mest grundläggande sättet som finns.

För att modifiera och ladda data på sidan behöver jag också bli bättre på DOM-manipulation.

3. Tidsplanering

Här skall du specificera egna deadlines för ditt projekt, när måste olika saker vara klara för att du ska vara klar med allt i tid. Dvs specificera deadlines och hållpunkter, dvs. exakt hur långt ska ni ha kommit vid specifika datum.

vecka	vad som ska vara klart
v42	planering
v43	statisk sida med temporära värden för att se slutdesign
v44	hämtad data, nersparning med localstorage
v45	avläsning av data, display av data
v46	inställnings sida, nersparning inställningar, website builder
v47	läsa av json → displaya widgets i rätt ordning