

# Planering

Jag ska skapa en enkel och användarvänlig webbsida som visar dagens fotbollsmatcher för de fyra mest populära ligor: Premier League, La Liga, Bundesliga och Serie A. Användaren ska kunna snabbt få överblick över dagens matcher och relevant information.

## Funktioner

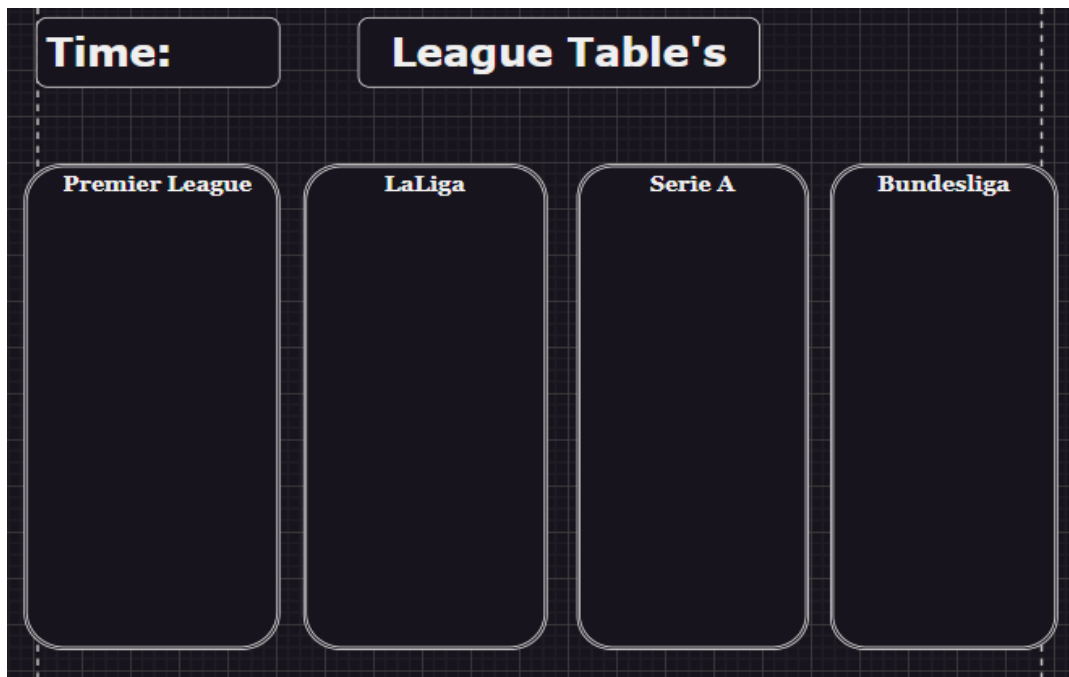
### Dagens Matcher:

- De ska lista dagens matcher för de valda ligorna.
- Visa lagens namn och matchstart tid..

### Metod/Verktyg:

- HTML/CSS: Grundläggande struktur och stil för webbsidan.
- Bootstrap: Använda Bootstrap för att snabbt implementera en responsiv och snygg design.
- API: Använda ett fotbolls-API för att hämta aktuell information om dagens matcher.
- Javascript: Göra sidan dynamisk och interaktiv genom att hämta aktuell matchinformation från ett API.

**BILD PÅ SKISS UNDER:**



## Documentation

### Sidans Utseende

Jag använde grundläggande HTML för att strukturera innehållet på ett sätt som är lätt att förstå och sökbart. Med hjälp av CSS skapade jag en ren design som fungerar bra på olika enheter.

### Dynamiskt Innehåll

För att hålla webbsidan uppdaterad och intressant för användarna, använde jag ett fotbolls-API. Det hjälpte mig att hämta och visa aktuell information om lag och deras placering i olika ligor.

### Användning av Andra Tjänster

Förutom fotbolls-API:et, använde jag även ett tidszons-API för att visa den nuvarande tiden i Stockholm. Det ger användarna en känsla av tid och plats när de besöker sidan.

### Utmaningar och Hur Jag Löste Dem

Jag stötte på vissa problem, särskilt när jag jobbade med externa API:er. För att övervinna detta, följde jag noga dokumentationen och testade min kod för att vara säker på att allt fungerade som det skulle. Att se till att sidan ser bra ut och fungerar korrekt på olika enheter var också en utmaning. Jag använde tester och enklare CSS-tekniker för att se till att allt ser bra ut och fungerar smidigt, oavsett vilken enhet användarna använder.

## Kodförklaring

1. **const leagueIds = ...**: Skapar ett objekt med ligornas namn och deras motsvarande ID.
2. **Object.entries(leagueIds).forEach(([league, leagueId]) => {**: Loopar igenom varje nyckel-värde-par i **leagueIds**.
3. **fetch(...)**: Gör en HTTP-förfrågan till API:et för varje liga med specifikt ID.
4. **.then(response => response.json())**: Konverterar svaret till JSON-format.
5. **data => {**: Hanterar JSON-data för varje liga.
6. **const teams = ...**: Skapar en array med rank och namn för varje lag i ligan.
7. **console.log(...)**: Skriver ut data och liginformation för varje liga till konsolen.
8. **const leagueTableElement = ...**: Hämtar HTML-elementet baserat på ligans namn.
9. **const teamsHTML = ...**: Skapar en HTML-sträng med liginformation.
10. **leagueTableElement.innerHTML = ...**: Uppdaterar HTML-elementet med liginformation.
11. **fetch(...)**: Gör en HTTP-förfrågan till Timezone API för att hämta aktuell tid i Stockholm.
12. **.then(response => response.json())**: Konverterar svaret till JSON-format.
13. **response => {**: Hanterar JSON-data för aktuell tid.
14. **const currentTime = ...**: Skapar ett Date-objekt från API-svaret.
15. **const formattedTime = ...**: Formaterar tiden som en sträng.
16. **const timeElement = ...**: Hämtar HTML-elementet för tiden.
17. **timeElement.textContent = ...**: Uppdaterar HTML-elementet med formaterad tid.
18. **updateTime();**: Anropar funktionen `updateTime` initialt.
19. **setInterval(updateTime, 30000);**: Sätter ett intervall för att anropa `updateTime` var 30:e sekund.