

# Projektuppgift

*Dt208g – Programmering i TypeScript*

**Höga kusten-högskolan**

**Markus Vickman**



**Mittuniversitetet**

MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. Campus Sundsvall Holmgatan 10, SE-851 70 Sundsvall.  
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.  
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

**MITTUNIVERSITETET**  
**Avdelningen för informationssystem och -teknologi**

**Författare:** Markus Vickman, [mavi2302@student.miun.se](mailto:mavi2302@student.miun.se)  
**Utbildningsprogram:** Webbutveckling, 120 hp  
**Huvudområde:** Datateknik  
**Termin, år:** VT, 2024

## Sammanfattning

Rapporten beskriver arbetets gång med att skapa en webbplats för ett fiktivt lärosäte vid namn Höga kusten-högskolan. Målet var att webbplatsen skulle likna ett riktigt lärosäte och även kunna hantera och presentera ett stort antal kurser. Webbplatsen behövde också vara responsiv.

Utvecklingen gjordes med ramverket Angular, modulbaserade tillägget Material UI och programmerades i TypeScript. Inspiration hämtades från lärosätena Mittuniversitet, Linnéuniversitetet och Luleå tekniska högskola. För att skilja sig från mängden blev högskolans tema dess naturnära läge och blå valdes till en genomgående färg för webbplatsen. Mobilanpassningar av tabellerna krävde mycket arbete där flera kolumner doldes för små skärmar. Detta gjorde att en "läs mer"-knapp behövde läggas till i tabellen för att all information skulle kunna visas i tabellen.

Resultatet för projektet blev en responsiv webbplats med modern design som listar alla kurser i en tabell. Det går även att lägga till kurser till ett eget ramschema. Utifrån tidsbegränsningarna lämnades förbättringsmöjligheter som att bygga ut fler undersidor med informations som länkas från startsidan. Även personliga inloggningar för elever och lärare hade skapat fler möjligheter för funktionalitet.

**Nyckelord:** TypeScript, Angular, Material UI.

# Innehållsförteckning

<b>Sammanfattning.....</b>	<b>3</b>
<b>1    Introduktion.....</b>	<b>5</b>
1.1    Bakgrund och problemmotivering.....	5
1.2    Avgränsningar.....	5
1.3    Detaljerad problemformulering.....	5
<b>2    Teori.....</b>	<b>6</b>
<b>3    Metod.....</b>	<b>7</b>
3.1.1    Sammanställd metodlista.....	8
<b>4    Konstruktion.....</b>	<b>9</b>
4.1    Designinspiration.....	9
4.2    Wireframe i Figma.....	9
4.3    Bilder från Unsplash.....	9
4.4    Logotyp i Adobe Express.....	9
4.5    Favicon i favicon.cc.....	9
4.6    Angular-projekt med Material UI.....	10
4.7    Komponenter.....	10
4.8    Tabellkomponenter.....	11
4.9    Routing.....	13
4.10    Angular-services.....	13
4.11    Test av webbplatsen.....	14
4.12    Responsivanpassning.....	14
4.13    Publicering till Netlify.....	14
<b>5    Resultat.....</b>	<b>16</b>
<b>6    Diskussion.....</b>	<b>17</b>
<b>Källförteckning.....</b>	<b>18</b>
<b>Bilaga 1. Webbplatsen.....</b>	<b>20</b>
<b>Bilaga 2. Wireframe.....</b>	<b>21</b>
<b>Bilaga 3. Mobily.....</b>	<b>22</b>

# 1 Introduktion

## 1.1 Bakgrund och problemmotivering

Rapporten beskriver arbetsprocessen för att skapa en webbplats åt en fiktiv högskola. Högskolan behöver en ny webbplats med en modernare design som även är mer anpassad för mobila enheter.

## 1.2 Avgränsningar

Fokus för uppgiften är att skapa en prototyp med tilltalande design för en högskolas webbplats. Avgränsningen är att inte göra en fullstor webbplats åt en högskola då den avsatta tiden för uppgiften inte räcker till för det.

## 1.3 Detaljerad problemformulering

För att kunna presentera en trovärdig prototyp som den fiktiva högskolan ska kunna utvärdera behövs flera funktioner integreras. Problem som ska lösas:

### **Modernt utseende**

Högskolan behöver fånga intresse och vara ett attraktivt alternativ för blivande studenter. Det är viktigt att högskolan inte upplevs som omodern.

### **Tydligt tema**

Ett tema och grafisk profil ska tydligt visa högskolans fördelar. Den ska också ge ett sammanhängande och seriöst intryck.

### **Responsiv design**

Tidigare webbplats var inte anpassad för små enheter. Enligt internetstiftelsens rapport "Svenskarna och internet 2018" [1] är mobiltelefoner de vanligaste enheterna för att använda internet. Därför är det viktigt att webbplatsen är responsiv både för stora och små enheter.

### **Presentera kurser**

För att studenter ska veta vilket utbud högskolan har och kunna hitta sina kursplaner behöver hela kursutbudet presenteras på ett lättöverskådligt sätt.

### **Personliga kurser**

För att kunna välja egna kurser och planera sina studier ska elever kunna lägga till kurser till ett eget ramschema.

## 2 Teori

### TypeScript

TypeScript är ett starkt typat programmeringsspråk som bygger vidare på JavaScript och är även kompatibelt med det[2].

### Angular

Angular är ett ramverk för utveckling av komponentbaserade "single page"-applikationer och är skapat av utvecklare hos Google[3].

### Angular - Service

En Angular-service är en enkel funktion eller ett värde som en komponent kan använda[4].

### Angular - Routing

För att binda ihop en webbplats eller delarna i en Angularapplikation används modulerna RouterModule och Routes i filen app-routing.module.ts där sökvägar kopplas ihop.[5]

### Angular - komponenter

Komponenter är byggstenar i Angular och innehåller en fil för html-, en för css och en för TypeScript. Det går att implementera dessa på huvud html-filen eller i annan valfri komponent.[6]

### Material UI

Material UI finns som ett tillägg till Angular och består av komponenter innehållandes HTML, TypeScript och CSS. Dessa går att implementera på ett egen projekt och anpassa till det.[7]

### Paginering

Paginering är sidouppdelning för innehåll som är av liknande karaktär eller likadant.[8]

## 3 Metod

Utvecklingen kommer att ske med Ramverket Angular och programmeringsspråket TypeScript. Angular är ett ramverk för utveckling av komponentbaserade "single page"-applikationer och är skapat av utvecklare hos Google[3]. TypeScript är ett starkt typat programmeringsspråk som bygger vidare på JavaScript och är även kompatibelt med det[2].

Metod för att lösa problem enligt problemdefinitionen:

### **Modernt utseende**

Webbplatsens inspiration hämtas från redan befintliga lärosäten som Mittuniversitetet, Luleå tekniska höghskola och Linnéuniversitetet. Material UI ska användas för att ge en modern design. Material UI finns som ett tillägg till Angular och är komponenter innehållandes HTML, TypeScript och CSS[7].

### **Tydligt tema**

Höghskolans namn ska vara Höga kustén-höghskolan och skolans tema ska bli natur. Temat ska vara genomgående för webbplatsen och ska framhävas i både text och med bilder från Unsplash. Unsplash är en plattform där användare delar bilder som är fria att använda till egna projekt[9].

### **Responsiv design**

Webbplatsens layout ska designas för mobil och utförligt testas på både datorer, Androidtelefoner och Iphone för att säkerställa ett responsivt resultat. Den ska även testas i de tre stora webbläsarmotorerna Chromium, Firefox och Safari.

### **Presentera kurser**

Kurserna ska presenteras i en tabell från Material UI. Funktioner ska implementeras för sortering, filtrering och paginering. Data till tabellen ska hämtas i en Angular-service med Angular-modulen HttpClient från JSON-data med alla kurser. En Angular-service är en enkel funktion eller ett värde som en komponent behöver. HttpClient används för att hämta data genom http-protokollet[10]. Det ska även gå att lägga till kurser till en personlig kurslista som sparas i localStorage.

### **Personliga kurser**

Liksom hela utbudet av kurser ska personliga kurser presenteras i en tabell från Material UI. Funktioner ska implementeras för sortering, filtrering och paginering. Data till tabellen av personliga kurser ska hämtas från localStorage genom en Angular-service. Kurser ska gå att ta bort från listan.

### 3.1.1 Sammanställd metodlista

1. Designinspiration
2. Wireframe i Figma
3. Bilder från Unsplash.
4. Logotyp i Adobe Express
5. Favicon i favicon.cc
6. Angular-projekt med Material UI
7. Komponenter
8. Tabellkomponenter
9. Angular-services.
10. Routing
11. Testa webbplatsen
12. Responsiv anpassning
13. Publicera på Netlify



## 4 Konstruktion

### 4.1 Designinspiration

Webbplatsens inspiration hämtades från de redan befintliga lärosätena Mittuniversitetet(miun), Luleå tekniska högskola (ltu) och Linnéuniversitetet(lnu). Designen som passade uppgiften bäst var hos ltu och det går tydligt att se inspiration därför på den färdiga webbplatsen. Se bilaga 1.

Gemensamt för alla tre lärosäten var att de lyfte forskning och därför tog jag med det i designen på förstasidan också. Miun hade fokus på människor och student-samarbeten därför tas campus och studentkåren med i designen på förstasidan.

För att sticka ut från övriga lärosäten valdes temat och högskolans nisch till "Natur".

### 4.2 Wireframe i Figma

Wireframen designades i Figma och skapades för att ge en bild av vilket innehåll som skulle vara med och hur layouten skulle vara. Den skapades både med en läge för dator och ett för småskärmar. Layouten lånar enligt designinspirationen från ltu men med egna fokusområden utifrån alla tre lärosäten. Se bilaga 2.

### 4.3 Bilder från Unsplash

Bilder till webbplatsen hittades på <https://unsplash.com/> med hjälp av sökorden university, forest, research och mountain biking. Bilderna valdes utifrån naturtema och för att passa till header och till rubrikerna forskning, natur, campus och studentkår. Med hjälp av bildredigeringsprogrammet Gimp skalades bilderna till storlekarna: header till 3556 x 250 pixlar, hero-bilden till 1200 x 600 pixlar och artikelbilderna till 500 x 334 pixlar. Samtliga bilder komprimerades till 75% kvalitet i jpg-formatet. Bilderna infogades senare i webbplatsen på startsidan och i header/footer som html-taggar och sparades i katalogen src/assets/image/.

### 4.4 Logotyp i Adobe Express

I webbverktygen Logo Maker skapades en logotyp genom att skriva in ett företagsnamn samt vilken bransch företaget tillhör. Här valdes namnet "Höga kust-högskolan", som bransch valdes education. Stilen för logotypen valdes till minimal och som sökord till ikonerna användes mountain. Ikonerna sparades och användes som huvudikon. Ikonerna sparades sedan i katalogen src/assets/image/icon.png

### 4.5 Favicon i favicon.cc

Faviconen ritades i webbverktyget [favicon.cc](https://favicon.cc). I det verktyget kan man rita pixel för pixel eller importera en bild. Jag valde att importera och konvertera huvudikonerna för webbplatsen till 16x16 pixlar i formatet .ico. Den andra faviconen skalades ner till 32x32 pixlar i bildredigeringsprogrammet Gimp och sparades i forma-

tet .png. Bilderna sparades i webbplatsens rotkatalog med filnamnen och formaten favicon.ico och favicon.png.

## 4.6 Angular-projekt med Material UI

Först ska node.js vara installerat. Sen installerades Angular CLI till node.js genom att skriva följande i terminalen:

- `npm install -g @angular/cli`

Ett Angular-projekt initierades genom att skriva kommandot nedan och då skapades projektet i en undermapp vid det namnet:

- `ng new ProjektNamnet`

För följande kommandon måste terminalen vara navigerad i projektkatalogen som skapades.

Sen skapades alla komponenter som skulle ingå i projektet med kommandot nedan för varje komponent:

- `ng generate component KomponentNamnet`

Komponenter skapades för header, footer, startsida, kurssida, ramschemasida, kurstabell, ramschematabell och pagenotfound.

För att lägga till material ui användes kommandot:

- `ng add @angular/material`

## 4.7 Komponenter

### Header

En komponent för headern skapades. För utseende användes flikar från Material UI med Angular-modulerna MatButtonModule och MatTabsModule. Dessa implementerades sedan i HTML-koden som går att se i figur 1. För länkar användes modulerna RouterLink och RouterLinkActive som också illustreras i figur 1.

Headern gavs också en bakgrundsbild av en molnig skog. Hur bilden valdes och formaterades går att läsa under tidigare rubrik "Bilder från Unsplash".

```
<header>
  <div id="background"><h1 id=logotext>Höga Kusten-Högskolan</h1></div>
  <a routerLink="/home"></a>
  <nav mat-tab-nav-bar [backgroundColor]="background" [tabPanel]="tabPanel">
    <a mat-tab-link routerLink="/home" routerLinkActive #rla="routerLinkActive" [active]=rla.isActive>Startsidan</a>
    <a mat-tab-link routerLink="/courses" routerLinkActive #rlb="routerLinkActive" [active]=rlb.isActive>Kurser</a>
    <a mat-tab-link routerLink="/frameschedule" routerLinkActive #rlc="routerLinkActive" [active]=rlc.isActive>Ramschema</a>
  </nav>
  <mat-tab-nav-panel #tabPanel></mat-tab-nav-panel>
</header>
```

Figur 1. Navigering

## Footer

Footern gavs samma bakgrundsbild som i headerna. Länkar med bilreferenser infogades, det gjorde även länkar för e-post och telefon. Ikoner för Facebook, X, Instagram och Youtube hämtades från [Flaticon.com](https://flaticon.com). Dessa ikoner infogades som inline-block längst ner i footern.

För att footern aldrig skulle visas i mitten av sidan även om sidan var tom använde jag följande kod för respektive footer, page-container som innehåller hela sidan (allt innanför body) och main:

```
#footer { position: absolute; bottom: 0; width: 100%;height: 150px; }  
#page-container { position: relative; min-height: 100vh; }  
#main { padding-bottom: 150px; }
```

## Startsida

På startsidan infogades längst upp en hero-bild med en knapp och texten "Pluggsugen?" ovanpå som länkar till kurssidan. Efter den skapades fyra artikel-element med vardera bild, rubrik och text. Dessa artiklar fick en blå bakgrundsfärg för att följa ett blått färgtema. Artiklarna visas två i bredd för stora skärmar och endast en i bredd för småskärmar.

## Kurssida och ramschemasida

Komponenterna för kurssidan och ramschemasidan innehåller endast sidornas rubriker och en kort beskrivande text. Dessa sidor innehåller komponenten för kurs-tabellen respektive ramschematabellen.

## Pagenotfound

En komponent för sidan som visas vid felinskriven adress skapades. Eftersom att den endast behöver visa att feladress är inskriven innehåller den endast följande HTML-kod:

```
<h2>404 - Site not found</h2><p>Vald undersida finns inte eller är ej tillgänglig</p>
```

## 4.8 Tabellkomponenter

### Kurstabell

Kurstabellen hämtar sin data från servicen för kursdata som det står mer om under rubriken Angular-services. Tabellen som används är en Material UI-modul med tillägg för paginering och som även har stöd för sortering. För att använda tabellen infogades kod för den i både html och i TypeScript export klassen. I tabellens html ändrades antal kolumner, kolumnnamnen och vilken information som skulle visas utifrån kursdata.

Tabellen innehöll redan ett inputfält för filtrering men saknade en valknapp för att kunna välja vilket ämne som skulle visas i tabellen. Denna knapp skapades som en select med options i. Dessa options byggdes upp från en array med alla ämnen som fanns med i kurslistan. Denna array av ämnen skapades i TypeScript-koden genom att gå igenom hela kurslistan och lägga till varje ämne i en array-variabel som inte redan var mer i variabeln. För att sedan bygga upp alla ämnen loopades variabeln igenom med hjälp av Angular-modulen NG-For.

All TypeScript-kod är skriven i klassen CourseTabelComponent. Från denna klass kan HTML-dokumentet hämta och skriva in data. I klassen skapades variabler för filtrerade kurser, kurser och ramschema. Nödvändiga variabler för material ui infogades också för kolumnnamn, kursämnen, datasource(kursdata, information om sortering och filtrering), paginering, sortering och till selectknappen.

För att ladda in data och initiera metoder användes modulen AfterViewInit på en metod vid samma namn. Den metoden används för att hämta data och prenumera på data från en angular-service. Den tilldelar också denna data till datasource som är den variabel som tabellen använder och till en variabel för kursdata. Eventlistener initieras också för de knappar i tabellen som senare läggs till i DOM med hjälp av TypeScript. Här initierades även metoden som går lägger till alla ämnen i ämnesvariabeln genom att alla kurser i kursvariabeln loopas igenom med en for-loop.

Metoderna för select och filtrering från tabellen initieras från html. Se figur 2.

```
//Lägger in filtret och initierar paginering från första sidan
public applyFilter(event: Event): void {
  const filterValue = (event.target as HTMLInputElement).value;
  this.dataSource.filter = filterValue.trim().toLowerCase();

  if (this.dataSource.paginator) {
    this.dataSource.paginator.firstPage();
  }
}

//Metod som använder filtrering för att endast visa valt ämne i tabellen och sen startar paginering.
public selectFilter(): void {
  this.filteredCourses = this.Courses.filter((Courses) => Courses.subject.includes(this.selected));
  this.dataSource = new MatTableDataSource(this.filteredCourses);
  this.dataSource.paginator = this.paginator;
  this.dataSource.sort = this.sort;
  if (this.dataSource.paginator) {
    this.dataSource.paginator.firstPage();
  }
}
```

Figur 2. Filtrering och ämnesval

## Ramschematabell

Denna tabell skapades på samma sätt. Skillnader ligger främst i den service som används och även att metoden AfterViewInit initierar funktioner för att säkra ut antal kurser i listan och antal poäng totalt i listan. Se figur 3.

```
//Metoder för att räkna antal poäng
private calculateTotal(): void {
  this.totalPoints = 0;
  for (let i = 0; i < this.Courses.length; i++) {
    this.totalPoints = this.totalPoints + this.Courses[i].points;
  };
}

//Metoder för att räkna antal kurser
private countCourses(): void {
  this.numberOfCourses = 0;
  this.numberOfCourses = this.Courses.length;
}
```

Figur 3. Metoder för totalt antal poäng och antal kurser.

## 4.9 Routing

För att binda ihop webbplatsen användes modulerna RouterModule och Routes i filen app-routing.module.ts och sökvägar kopplades ihop. Se figur 4.

```
export const routes: Routes = [
  {path: 'home', component: StartPageComponent},
  {path: 'courses', component: CoursePageComponent},
  {path: 'frameschedule', component: FrameSchedulePageComponent},
  {path: '', component: StartPageComponent},
  {path: '**', component: NotFoundComponent}
];
```

Figur 4. Routing webbsidor

Header och footer lades till på alla tre sidor på webbplatsen genom att skriva in komponent namnen i app.component.html-filen ex. <app-header-menu />. På samma sätt lades tabellkomponenterna till i kurs och ramschama sidan.

## 4.10 Angular-services

Två enklare services skapades för att hantera kursdata till tabellerna som de kan prenumerera på.

### Kursservice

För alla kurser skapades en service som hämtar kursdata från en webbadress innehållandes JSON-data över alla kurser. Detta gjordes med hjälp av modulen HttpClient och en metod för att returnera kursdata till tabellkomponenten. Se figur 5.

### Ramschemaservice

För valda kurser till ramschemat skapades en service som hämtar kursdata från localStorage som konverteras från sträng till JSON-data. En metod som pushar alla kurser till en array av object skapades och den används också för att returnera kursdata till ramschemakomponenten skapades.

```
//Klass för att hämta kurser med HttpClient
export class GetCoursesService {
  url: string = 'https://matdah.github.io/DT208G---Programming-i-TypeScript/Moment%20-%20Projekt/miun_courses.json';

  constructor(private http: HttpClient) {}

  //Metod som används för att hämta och returnera kursdata
  getCourses(): Observable<Courses[]> {
    return this.http.get<Courses[]>({this.url});
  }
}
```

Figur 5. Service med HttpClient.

## 4.11 Test av webbplatsen

Webbplatsen testades löpande i utvecklingsläget på webbläsarna Firefox och Chrome utan några större problem. Den klarade också Html och CSS-validering utan problem. När webbplatsen testades på androidtelefoner var det enda problemet tabellerna. Tabellerna var bara responsiva ner till 700pixlar och jag ville att den färdiga webbplatsen ska vara responsiv ner till åtminstone 350pixlar. Efter testade med att dölja kolumner kom jag fram till att för mycket av informationen skulle gå förlorad utan ett specifikt mobilläge för tabellerna.

Enligt dos-lagen[11] omfattas offentliga verksamheter av lagen om tillgänglighet till digital offentlig service. Därför testades även sidan om denna följdes med hjälp av onlineverktyget "ace it"[12]. Testet gav resultatet att sidan inte följde lagen på grund av brister i alt-texter för ikoner och avsaknad av role="search" på sökrutorna. Dessa brister rättades till. Dock kvarstår vissa brister för tabellerna på grund av hur material UI har byggt sina tabeller och för att innehåller byggs upp dynamiskt.

## 4.12 Responsiv anpassning

För responsivitet valdes att bara behålla kolumnerna för kursnamn, hp-poäng synliga. Men eftersom att den informationen blev för sparsam skapades en Metod för tabellen som skapade en egen ruta med all tillgänglig information om kursen. Rutan visas genom att en knapp i tabellen klickas på och eventet on-click="showmore(specifik kurskod)" initierar metoden. I rutan finns även knapp för att lägga till kursen i kurslistan. För att stänga informationsrutan går det att klicka på bakgrunden eller klicka på stänga knappen. När en kurs läggs till i listan visas en text i rutan som meddelar att kursen är tillagd. Mer konkret hur resultatet av mobilläget blev går att se i bilaga 3.

Efter feedback från Iphone-användare upptäcktes att flera knappar inte fungerade. Därför skrevs koden om för eventlisteners och så mycket som möjligt aktiverades från html med "on-click"-event istället för eventlistener i TypeScript-koden.

## 4.13 Publicering till Netlify

Under hela arbetet har projektet varit versionshanterat med Git till Github. För att publicera till [Netlify](#) skapade jag en fil i src-katalogen för mitt projekt vid namn \_redirects. I den filen skrev jag in sökvägar för min index om endast webbadressen skrivs in och en sökväg ämnad för pagenotfound om en adress som inte finns för webbplatsen skrivs in. Detta gjordes med följande kod:

```
/* /index.html 200
```

```
/** /index.html 200
```

Vid publicering kände Netlify av att det var ett Angular projekt och ställde in rätt mapp åt mig. Jag behövde bara lägga till `ng build`, se figur 6.



**Figur 6. Publicering till Netlify från Github.**



## 5 Resultat

Resultatet blev en webbplats med design inspirerad från befintliga lärosäten. Webbplatsen har tabeller för att lista kurser. Tabellerna innehåller mycket information och har funktioner som sortering, sidouppdelning, filtrering och sökning. Möter webbplatsen problemdefinitionen?

### **Modernt utseende**

Med ett tydligt blått tema och raka linjer i både uppdelningar och menyer ger sidan ett modernt intryck. Moduler från Material UI för både navigering och tabeller känns sidan både proffsig och modern.

### **Tydligt tema**

Webbplatsen fick temat natur och tydliga inslag av natur syns på bilder både i header och footer men även i artikelbilderna om skolan. Den blåa färgen från artiklarna följer också med till övriga sidor i navigeringen och även i överkant av footern.

### **Responsiv design**

Det syns att extra anpassningar har gjorts för att webbplatsen ska fungera på små-enheter. Hela vägen från startsidan som designades med mobilen i första rum till tabellen som har ett eget läge för mobil. Detta läge säkerställer att mobilanvändarna får en lika bra upplevelse som om de suttit vid en dator. Ingen information tas bort utan all information finns tillgänglig även här.

### **Presentera kurser**

Hela högskolans utbud av kurser visas tydligt i en tabell. Kurserna går att söka, sortera eller lista efter ämne. Paginering gör också paginering/sidouppdelning gör också kurserna mer överskådliga där det tydligt visas antal kurser i listan eller sökningen.

### **Personliga kurser**

I kurslistan går det att lägga till kurser som studenten läser eller är intresserad av. Dessa kurser lagras då i localStorage och presenteras på ramschemasidan. På ramschema sidan går det också att ta bort kurser från listan. Dessa kurser presenteras i en likadan tabell som för kurssidans. En skillnad är att under tabellen visas antal poäng totalt och antal kurser i listan.



## 6 Diskussion

Jag är nöjd med resultatet och tycker att det ger ett proffsigt intryck. Webbplatsen i sin nuvarande form har inte så många funktioner men den ger ändå en bra förhandsvisning som en prototyp. Tabellerna fungerar bra för både mobiltelefoner och datorer. Men med tanke på hur svårt det var att mobilanpassa tabellen hade jag med facit i hand hellre skapat en tabell som visar mer information direkt i tabellen vid klick. Se figur 7.

Tiden för uppgiften har begränsat hur omfattande webbplatsen blev. Informationen på webbplatsen totalt sett blir väldigt smal och många fler sidor med information hade behövts för att det skulle likna en sida för ett riktigt universitet. Det saknas inloggning och sidor där specifik information för elever kan visas. Det hade också varit bra att bifoga länkar som söker på [www.antagning.se](http://www.antagning.se) för dirigera dit kurserna ska sökas.

De första förbättringarna jag hade gjort hade varit att länka artiklarna från startsidan till riktiga undersidor. Fler länkar till undersidor i footern hade också varit bra. En inloggning för elever och lärare hade också skapat fler möjligheter för webbplatsen. För att kunna presentera dessa förändringar på ett bra sätt hade det också behövts en omarbetad menyläge för mobiltelefoner.

Det roligaste med att skapa denna webbplats var att det går att skapa en proffsig webbplats med ganska enkel design. Det som krävdes var lite efterforskning och arbete med att implementera de moduler som passade in i projektet från Angular och material ui.

Table with expandable rows

name	weight	symbol	position	
Hydrogen	1.0079	H	1	▼
Helium	4.0026	He	2	▼
Lithium	6.941	Li	3	▲

3

Li

Lithium

6.941

Lithium is a chemical element with symbol Li and atomic number 3. It is a soft, silvery-white alkali metal. Under standard conditions, it is the lightest metal and the lightest solid element. -- Wikipedia

Figur 7. Exempel med förstorande rader i en tabell från materia ui[13].

## Källförteckning

- [1] Internetstiftelsen, "Svenskarna och internet"  
<https://svenskarnaochinternet.se/rapporter/svenskarna-och-internet-2018/allmant-om-internetutvecklingen/dator-vanligare-an-mobil-bland-aldre/>  
Publicerad 2018-11-03. Hämtad 2024-05-14.
- [2] TypeScript, "What is TypeScript?",  
<https://www.typescriptlang.org/> Ändrad 2023-08-15. Hämtad 2024-05-14.
- [3] Angular, "What is Angular?",  
<https://angular.io/guide/what-is-angular> Ändrad 2023-08-15. Hämtad 2024-05-14.
- [4] Angular, "Creating an injectable service",  
<https://angular.io/guide/creating-injectable-service> Uppdaterad 2022-08-02. Hämtad 2024-05-14.
- [5] Angular, "Angular Routing", <https://angular.io/guide/routing-overview>  
Ändrad 2022-02-28. Hämtad 2024-05-16.
- [6] Angular, "Angular components overview",  
<https://angular.io/guide/component-overview> Ändrad 2023-07-29. Hämtad 2024-05-16.
- [7] Angular Material, "Angular Material",  
<https://material.angular.io/> Hämtad 2024-05-14.
- [8] SEOptimer, "What is Pagination?",  
<https://www.seoptimizer.com/blog/what-is-pagination/> Hämtad 2024-05-16.
- [9] Unsplash, "Unsplash",  
<https://unsplash.com/> Hämtad 2024-05-14.
- [10] Angular, "HttpClient",  
<https://angular.io/api/common/http/HttpClient> Hämtad 2024-05-14.
- [11] DIGG, "Om lagen om tillgänglighet till digital offentlig service",  
<https://www.digg.se/analys-och-uppfoljning/lagen-om-tillganglighet-till-digital-offentlig-service-dos-lagen/om-lagen> Uppdaterad 2024-01-04.  
Hämtad 2024-05-16.
- [12] Ace it, "Automatiskt test", <https://ace.useit.se/ax/aceit.php> Uppdaterad 2024-01-04. Hämtad 2024-05-16.

- [13] Angular Material, "Examples",  
<https://material.angular.io/components/table/examples> Hämtad 2024-05-14.

## Bilaga 1. Webbplatsen



Höga Kusten-Högskolan

Startsidan

Kurser

Ramschema

Pluggsugen?

Våra kurser



### Forskning

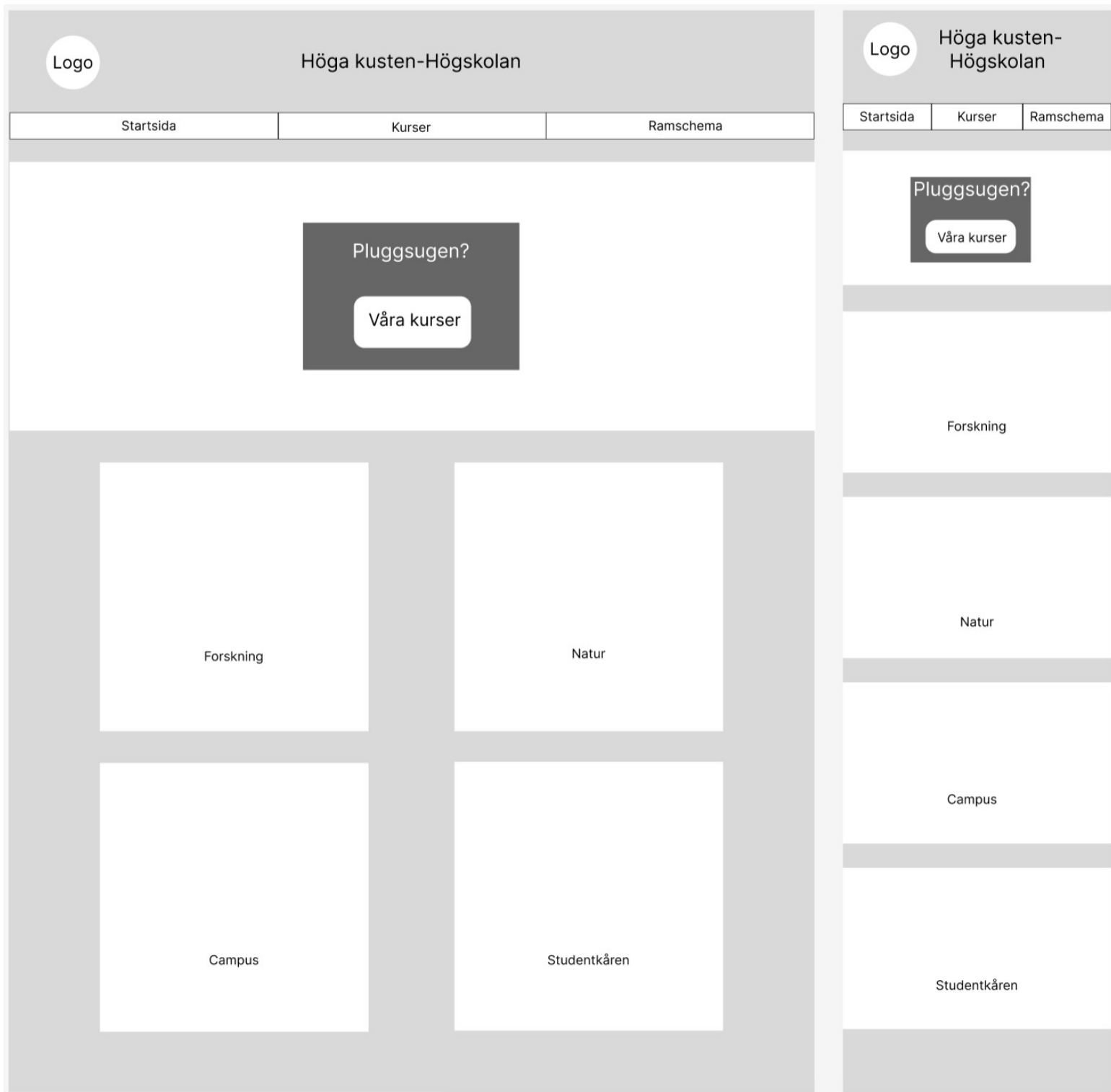
Efter studierna finns det stora möjligheter till att doktorera och forska vidare på högskolan.



### Natur

Höga Kusten-Högskolan erbjuder moderna lokaler. Våran högskola är belägen i skogen nära havet i hjärtat av höga kusten. Vi erbjuder bra utbildningar i en fridfull miljö.

## Bilaga 2. Wireframe





## Bilaga 3. Mobilvy

