**PROCESRAPPORT**

Sidernes Verden – Webshop til salg af eventyrlige bøger



Michael Aggerholm | Svendeprøve | 9. april - 14. maj 2024

Datatekniker m. speciale i programmering

Indhold

[Indledning 2](#_Toc163647056)

[Læsevejledning 2](#_Toc163647057)

[Forord 3](#_Toc163647058)

[Case 4](#_Toc163647059)

[Problemformulering 4](#_Toc163647060)

[Projektplanlægning 5](#_Toc163647061)

[Metodevalg og teknologi 6](#_Toc163647062)

[PHP 6](#_Toc163647063)

[GitHub 6](#_Toc163647064)

[Laravel 6](#_Toc163647065)

[Laravel's Blade Framework 7](#_Toc163647066)

[Bootstrap 7](#_Toc163647067)

[NPM 7](#_Toc163647068)

[Composer 8](#_Toc163647069)

[Laravel Sail 8](#_Toc163647070)

[Konklusion 9](#_Toc163647071)

[Bilag 10](#_Toc163647072)

# Indledning

SidernesVerden er udarbejdet som svendeprøve projekt under uddannelsen som datateknikker med speciale i programmering på Tech College i Aalborg.

I rapporten beskrives udviklingsprocessen fra projektstart til endelig aflevering samt hvilke teknologier og metoder der er valgt.  
Der vil være dokumenteret hvilke udfordringer som er opstået undervejs, hvordan de er løst, samt hvilke alternativer der eventuelt ville kunne være valgt.

Formålet med rapporten er at give læseren klart indblik i den udviklingsproces og de valg som er foretaget under udviklingen af SidernesVerden.

Under læsningen kan det være en fordel at have adgang til projektet, det findes frit tilgængeligt via GitHub url:  
<https://github.com/MichaelAggerholm/Apprentice_Test>

# Læsevejledning

Denne del er projektets procesrapport, den bør læses før produktrapporten da den indeholder beskrivelsen af processen der er gennemgået for at ramme det endelige produkt.

Procesrapporten indeholder information om projektets planlægning, design, udvikling, test og implementering. Procesrapporten giver også en detaljeret beskrivelse af de udfordringer der er stødt på undervejs, samt hvordan jeg har valgt at løse dem.

# Forord

Sidernes Verden – En webshop til salg af eventyrlige bøger, er et projektet der forener passionen for eventyrlige fortællinger med den moderne digitale handelsverden. Firmaet 'SidernesVerden' søger en løsning som lever op til kundernes forventninger, når de træder ind i den virtuelle webshop af bøger.

Målet for projektet er at udarbejde en webshop som er enkel og funktionel for kunder, samtidig med at der er overskuelighed og gennemtænkt funktionalitet for firmaets ansatte som skal vedligeholde webshoppen. Til dette formål er valget faldet på Laravel, et kraftfuldt og moderne PHP-framework, med blade frameworket som frontend og Laravel backend. Dette sikrer en robust og skalerbar platform, der kan imødekomme både nutidige og fremtidige behov.

Med anvendelsen af en SQLite-database opnås der hurtig og effektiv datalagring Dette er særligt relevant for et projekt i denne størrelsesorden, hvor ressourceoptimering er en central faktor.

der blive implementeret en detaljeret hændelseslog, som dokumenterer alle administrative interaktioner på webshoppen som gør det muligt at identificere, hvilke medarbejdere der har foretaget ændringer i forbindelse med bøger eller andre elementer, hvilket kan være afgørende for at identificere fejl og sikre en hurtig rettelse af fejlen, hvilket vil bidrage til højere kvalitet af webshoppen samt bedre brugeroplevelse for kunder samt ansatte.

Endvidere vil der blive fokuseret på implementeringen af en funktion til import af bøger via Excel-ark. Dette vil ikke blot lette processen med at opdatere sortimentet før lanceringen af webshoppen, men også sikre, at virksomheden kan udvide sit udvalg effektivt og i takt med kundernes efterspørgsel.

Gennem denne rapport bliver der dykket ned i de tekniske aspekter af projektet, samt undersøgt hvordan valget af Laravel, blade frameworket og valget af SQLite databasen kan bidrage til at opfylde fyldestgørende målsætning for webshoppen til 'SidernesVerden'.

# Case

Firmaet 'SidernesVerden' ønsker en webshop til salg af bøger om eventyrlige fortællinger.

Webshoppen skal give en hurtig og intuitiv oplevelse for kunderne. Det er vigtigt, at kunderne nemt kan finde de ønskede bøger og at grænsefladen er brugervenlig.

Ejeren og medarbejderne hos 'SidernesVerden' skal have mulighed for at få et fuldt overblik over alle aspekter af webshoppen, herunder bøger, forfattere, udgivere, salg og brugere. Desuden skal medarbejderne have adgang til en log, hvor de kan se, hvilke ændringer der er foretaget på shoppen, og af hvem.

Det er også vigtigt for virksomheden at have mulighed for at importere nye bøger fra Excel-filer, både før lanceringen af webshoppen og på sigt, hvis der skulle komme aftaler med nye leverandører.

# Problemformulering

Er det muligt at udvikle en webshop, hvor forbrugere kan registrere sig og foretage køb af bøger, samtidig med at medarbejdere kan logge ind, administrere og have overblik over webshoppen?

# Projektplanlægning

Under projektudviklingen bliver der benyttet Kanban til projektstyring, det er valgt da Kanban er en agil metode som er nem at forstå og anvende til mindre projekter som dette, det giver overblik over opgaver samt sikrer at man ikke overser lavere prioriterede opgaver. Der er valgt at benytte GitHubs Kanban board da projektet i forvejen ligger i GitHub, det giver også let adgang til at kunne uddelegere opgaver på tværs af udviklere på sigt. Igennem projektet bliver opgaver fulgt via Kanban boardet og alle nye opgaver får hver sin ”task” som bliver fulgt til ende. Projektstyringen er opstillet således at der afholdes stand-up møde ved Kanban boardet med status og opdatering til opgaver 3 gange om ugen, grunden til at afholde stand-up ’s så ofte er da projektet styres af et enkeltmandsteam og over en kort periode med forholdsvis mange opgaver.  
Der er taget billede undervejs af Kanban processen, de kan findes via bilag her:  
Bilag Kanban fase 1 s. 11  
Bilag Kanban fase 2 s. 12  
Bilag Kanban endelig fase s. 13

Inden udviklingsopstart er der truffet nogle valg for at komme frem til de frameworks, tredjeparts libraries mm. Som er valgt til projektet, dette er blandt andet valgt ud fra mange muligheder som vist via mindmap (se bilag side 213123), i mindmap er der skrevet op til flere frameworks, databaser og diverse muligheder for at skave lige præcis det opnået udgangspunkt, efterfølgende har teamet nøje udvalgt elementer som arbejder godt sammen ift. Projektets størrelse og det ønskede resultat samt en del baseret på teamets erfaring og den mængde tid der er sat til forløbet.

Herefter bliver hver opgave defineret og beskrevet i kravspecifikationen (se dokumenteret kravspecifikation i produktrapport s. 11111), i kravspecifikationen er basal funktionalitet slået sammen for at skabe overblik, det gør sig gældende blandt andet for håndtering af bøger, herunder CRUD-funktionalitet mm. herefter bliver hver opgave opstillet i testspecifikation (se dokumenteret test specifikation i produktrapport s. 2222).

Ved projektstart er der udarbejdet en estimeret tidsplan (se bilag estimeret tidsplan s. 123) som teamet baseret på tidligere erfaring forventer at processen vil forløbe efter, sideløbende udfyldes en realiseret tidsplan for at løbende at følge og dokumentere den faktiske fremdrift (Se bilag: realiseret tidsplan, s. 12313).

I begyndelsen af udviklingsprocessen bliver der udarbejdet et database-diagram for at sikre, at relationerne og strukturen giver et solidt fundament (se bilag: database diagram s. 21323). Dernæst bliver et klassediagram opsat for at give et overblik over de controllers og klasser der skal oprettes i backend (se bilag: klassediagram, s. 213213123).

Efter fastlæggelse af database og klassestruktur går udviklingsfasen i gang, som beskrevet i Kanban-boardet og tidsplanen, opdelt i tre faser: Frontend, Database og Backend. Gennem hele projektforløbet bliver der løbende taget noter om tilbageblik på opgaver, udfordringer og genovervejelser. Disse noter bliver brugt til refleksion og læring fra de forskellige situationer og udfordringer, der opstår undervejs.

Under hele projektforløbet bliver der ført en logbog, hvor daglige opgaver og eventuelle udfordringer bliver dokumenteret (se bilag: logbog, s. 213123).

# Metodevalg og teknologi

I forbindelse med projektet er der truffet beslutninger vedrørende valg af metoder og teknologier. Disse valg og overvejelserne bag dem vil blive uddybet i det følgende afsnit.

## PHP

PHP er blevet valgt som hovedsproget til udviklingen af dette projekt af flere årsager. For det første har PHP en lang historie inden for webudvikling og har vist sig at være pålideligt og skalerbart til opbygning af komplekse webapplikationer. Derudover har PHP et stort udviklermiljø og et væld af dokumentation og ressourcer, hvilket gør det nemt at finde løsninger og få hjælp, når det er nødvendigt.

## GitHub

GitHub er en onlineplatform, der tilbyder en bred vifte af værktøjer til projektstyring, herunder lagring af projekter og versionsstyring. Dets anerkendte ry og høje kvalitet gør det til det ideelle valg for projektet. Et af de centrale værktøjer som vi benytter os af, er GitHub Actions, også kaldet ”Pipelines”, disse kører automatisk ved hver eneste kodeændring som bliver pushet op via Git, som sikrer en konstant kvalitetskontrol og fejlfinding i projektet.

Derudover leverer GitHub også et intuitivt Kanban board, som bliver anvendt til projektstyring. GitHub leverer alt i alt en omfattende pakke af funktioner og værktøjer som bidrager til en effektiv og struktureret gennemførelse af projektet.

## Laravel

Laravel blev valgt som det primære framework til udviklingen af flere vigtige årsager. For det første tilbyder Laravel en omfattende og velstruktureret løsning til webudvikling.  
Dets arkitektur og indbyggede funktioner gør det muligt at opbygge applikationer på en struktureret og effektiv måde.

Laravel tilbyder en bred vifte af indbyggede funktioner og værktøjer, herunder en kraftfuldt ORM (Object–Relational Mapping) kaldet Eloquent, migrations, og et omfattende sæt af kommandoer til at lette udviklingsprocessen.

Laravel er et meget anvendt framework med en gennemarbejdet dokumentation som gør det nemt at finde hjælp når det er nødvendigt.

## Laravel's Blade Framework

Til frontend-delen af projektet er der valgt Laravel's Blade framework. Blade har en række stærke funktioner, herunder Blade's evne til at lette integrationen af PHP-kode, der sendes fra controlleren. Dette gør det muligt for os at opretholde en klar adskillelse mellem frontend og backend, samtidig med at vi udnytter PHP's styrke til at dynamisk generere indhold.

Blade har en stærk evne til at genanvende komponenter, hvilket betyder at vi kan opbygge og vedligeholde vores frontend-kode på en mere struktureret og effektiv måde. Dette resulterer i en mere skalerbar og vedligeholdelsesvenlig kodebase. Samlet set gør Blade frameworket det muligt for os at opnå en stilfuld og funktionel frontend, der integreres sømløst med Laravel's backend, hvilket sikrer en sammenhængende og robust webapplikation.

## Bootstrap

Bootstrap er et moderne css framework som giver mulighed for hurtigt at opbygge responsiv brugergrænseflade. Med Bootstrap kommer der et bredt udvalg af færdiglavede komponenter, layouts og css klasser som kan spare tid og reducere behovet for alt for meget nyudvikling af kode. Desuden tilbyder Bootstrap mange tilpasningsmuligheder kun ved hjælp af tilføjelser af klasser sin gør det simpelt at opstille design.

## NPM

Projektet benytter NPM (Node Package Manager) i Laravel for at håndtere afhængigheder og pakker, eksempelvis installationen af Bootstrap. Ved hjælp af NPM kan der nemt installeres / afinstalleres og administreres forskellige frontend biblioteker og værktøjer.

En af de primære fordele ved at bruge NPM er muligheden for at se live kodeændringer direkte browseren under udviklingsfasen ved hjælp af ”npm run dev” kommandoen.

Når projektet er klar til at blive lanceret, benyttes kommandoen ”npm run build” i stedet, som genererer en optimeret version af min frontend kode, eksempelvis ved at minify CSS og JavaScript, som reducerer filstørrelsen og forbedrer ydeevnen i produktionsmiljøet.

## Composer

Composer er et kraftfuldt værktøj til PHP, der gør det nemt at installere, opdatere og administrere tredjeparts biblioteker og frameworks.

Desuden gør Composer det muligt at udnytte et stort økosystem af open-source PHP-pakker, der spænder over alt fra frameworks til utilities og libraries som eksempelvis det logging library der er gjort nytte af i webshoppens administration (Link til logging library).

Ved hjælp af kommandoer som ”composer install” og ”composer update” holdes projektet opdateret og korrekt konfigureret, det giver en mere stabil og pålidelig webapplikation.

## Laravel Sail

Sail er Laravel’s indbyggede Docker-miljø som gør det nemt at opsætte og køre projektet i et isoleret og konsistent udviklingsmiljø uanset hvilket system jeg er på, samt når projektet skal gå live på en web server.

Ved hjælp af Sail er det muligt at starte hele Laravel projektet op, med installation of composer pakker og lignende, alt sammen ved hjælp af én enkelt kommando og på få sekunder / minutter hvis det er første kørsel på ny enhed.

## Stripe

Stripe er valgt som betalingsløsning for webshoppen. Stripe tilbyder en brugervenlig platform og gør det nemt at integrere betalingsfunktionalitet i webshoppen.

Med Stripe kan kunder betale via kreditkort, dankort og debetkort. En af de mest tiltalende funktioner ved Stripe et meget brugervenlige onlinepanel, her kan man via en token logge ind og dermed have overblik over alle salg, betalinger og lignende som er foretaget fra webshoppen.

# Konklusion

## Referencer

# Bilag

Herunder findes bilag.

**Mindmap**

Også tilgængeligt via url: <https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/mindmap.drawio.pdf>

****

**Databasediagram**

Også tilgængeligt via url: <https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/database_diagram.drawio.pdf>



**Klassediagram**

Også tilgængeligt via url: <https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/klasse_diagram.drawio.pdf>



**Kanban Fase 1 – opstart af projekt**

Dette billede fra Kanban boardet er taget ved projektets opstart, lige inden opsætningen af Laravel projektet.

For at se Kanban boardet ’s nuværende fase, kan det tilgås via GitHub URL:  
<https://github.com/users/MichaelAggerholm/projects/2/views/1>

****

**Kanban Fase 2 – midtvejs i projekt**

**Kanban Fase 3 – Afslutning af projekt**

# Logbog

Logbog føres primært for at dokumentere daglige opgaver, udfordringer og løsninger. Det giver oversigt over projektets udvikling og muliggør refleksion over processen.

**09/04 – Tirsdag**

Intro til svendeprøve på Tech College.  
Efterfølgende er der valgt udarbejdelse af projekt fra delvis hjemmekontor/arbejdsplads fremadrettet.  
Opstart/opstilling af produktrapport samt procesrapport.  
Opsætning af Git repository til lagring af alle projektfiler.   
Mindmap, valg af metoder og teknologier.  
Udarbejdelse af Case beskrivelse og problemformulering.  
Klargøring af Kanban board til projektstyring via GitHub.

**10/04 – Onsdag**

Arbejde på produktrapport samt procesrapport.  
Udarbejdelse af Estimeret tidsplan.  
Opsætning af Kravspecifikation og Testspecifikation.  
Opsætning af databasediagram  
Opsætning af klassediagram

**11/04 – Torsdag**

Udarbejdet begrundelse for valg af metoder og teknologier  
Beskrivelse af produktets arkitektur i produktrapporten  
Beskrivelse af projektplanlægningen og det fremtidige forløb i procesrapporten.

**12/04 – Fredag**

Opstilling af udviklingstasks på Kanban board.  
Opsætning af Laravel applikation  
Opsætning af GitHub Actions til automatisk test workflow.

**15/04 – Mandag**

Yderligere tests af GitHub workflow.  
Påbegyndt design af frontend.  
Påbegyndt design af admin panel.  
Opsætning af migrations.  
Oprettet modeller, controllers og migreret til database.  
Påbegyndt opsætning af authentication.

**16/04 – Tirsdag**

Opsætning af administrerings funktionalitet for formater.  
Opsætning af authentication og fejlhåndtering ved un-authenticated adgang til panel.  
Opsætning af Adminpanel og routes samt middleware for adgangsstyring.  
Test og gennemgang af formater, opret, slet, genskab.

**17/04 – Onsdag**

**18/04 – Torsdag**

**19/04 – Fredag**

**22/04 – Mandag**

**23/04 – Tirsdag**

**24/04 – Onsdag**

**25/04 – Torsdag**

**26/04 – Fredag**

**29/04 – Mandag**

**30/04 – Tirsdag**

**01/05 – Onsdag**

**02/05 – Torsdag**

**03/05 - Fredag**