

PROCESRAPPORT

Sidernes Verden – Webshop til salg af eventyrlige bøger

<http://128.199.39.171/>



Michael Aggerholm | Svendeprøve | 9. april - 14. maj 2024
Datatekniker m. speciale i programmering

Indhold

Indledning.....	2
Læsevejledning	2
Forord.....	3
Case	4
Problemformulering.....	4
Projektplanlægning	5
Metodevalg og teknologi	6
PHP	6
GitHub	6
Laravel.....	6
Laravel's Blade Framework	7
Bootstrap	7
NPM	7
Laravel Sail.....	7
Composer.....	8
Stripe.....	8
Laravel Debugbar	8
ActivityLog.....	9
laravel-Excel	9
Xdebug.....	9
Design og implementering.....	10
MVC designmønster	10
Service Provider.....	10
Middleware	10
Observer Pattern	10
Soft Deletes.....	11
Factories og Seeders	11
Konklusion	12
Diskussion.....	13
Bilag.....	14
Logbog.....	24

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 projektfiler, de findes frit tilgængeligt via GitHub url:

https://github.com/MichaelAggerholm/Apprentice_Test

Det er også muligt at tilgå webshoppen online via url: <http://128.199.39.171/>

Guide til brug af webshoppen findes i produktrapport side 21.

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 opstået undervejs, samt valg af løsninger hertil.

Forord

Sidernes Verden – En webshop til salg af eventyrlige bøger, er et projekt 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, dette 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 og bidrage til højere kvalitet af webshoppen samt bedre brugeroplevelse for kunder og ansatte.

Endvidere vil der blive fokuseret på at implementerer en 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 ugentligt. Der er taget billeder af Kanban processen undervejs, de kan findes via bilag her:

[Bilag: Kanban fase 1 s. 19](#)

[Bilag: Kanban fase 2 s. 20](#)

[Bilag: Kanban fase s. 21](#)

Inden udviklingsopstart er der truffet nogle valg for at komme frem til blandt andet det valgte programmeringssprog, frameworks og tredjeparts libraries som er valgt til projektet, dette er blandt andet valgt ud fra mange muligheder som vist via mindmap ([se bilag: Mindmap s. 16](#)), i mindmap er der skrevet op til flere frameworks, databaser og diverse muligheder for at skabe 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 projektet.

Herefter bliver hver opgave defineret og beskrevet i kravspecifikationen (se dokumenteret kravspecifikation i produktrapport s. 9), 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. 13).

Ved projektstart er der udarbejdet en estimeret tidsplan ([se bilag: Estimeret tidsplan s. 14](#)) som teamet baseret på tidligere erfaring forventer at processen vil forløbe, sideløbende udfyldes en realiseret tidsplan for løbende at kunne følge og dokumentere den faktiske fremdrift ([Se bilag: Realiseret tidsplan s. 15](#)).

I begyndelsen af udviklingsprocessen er der udarbejdet et database-diagram for at sikre, at relationerne og strukturen giver et solidt fundament ([se bilag: database diagram s. 17](#)). Dernæst et klassediagram for at give et overblik over de controllers og klasser der skal oprettes i backend ([se bilag: klassediagram, s. 18](#)).

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 projektførløbet bliver der ugentligt på stand-up lavet status og tilbageblik på opgaver, udfordringer og genovervejelser, denne viden bliver brugt til refleksion og læring fra de forskellige situationer og udfordringer som er opstået undervejs.

Under hele projektførløbet bliver der ført logbog hvor daglige opgaver og eventuelle udfordringer bliver dokumenteret ([se bilag: logbog, s. 24](#)).

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 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 samt et væld af dokumentation og ressourcer hvilket gør det nemt at finde løsninger og få hjælp, når det er nødvendigt.

PHP tilbyder egen dokumentation som kan findes på <https://www.php.net/>

GitHub

GitHub er en onlineplatform som 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 ændring som bliver pushet op via Git og sikrer en konstant kvalitetskontrol samt 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.

GitHub kan tilgås via <https://github.com/> derudover kan GitHubs egen gennemarbejdet dokumentation findes på <https://docs.github.com/en>

Laravel

Laravel er 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 effektiv og stabil 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 og øge produktiviteten.

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

Dokumentationen for Laravel versionen som er benyttet i dette projekt kan findes på <https://laravel.com/docs/11.x/readme>

Udover det er der oversigt over alle indbyggede kommandoer på <https://artisan.page/>

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 at opretholde en klar adskillelse mellem frontend og backend, samtidig med at udnytte PHP's styrke til dynamisk at generere indhold.

Blade har en stærk evne til at genanvende komponenter, hvilket betyder at der kan opbygges og vedligeholdes 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 at opnå en stilfuld og funktionel frontend som integreres sømløst med Laravel's backend, hvilket sikrer en sammenhængende og robust webapplikation.

Bootstrap

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

En rigtig god ressource er Bootstraps egen dokumentation på <https://getbootstrap.com/>

NPM

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

En af de primære fordele ved at bruge NPM er muligheden for at se live kodeændringer direkte i 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 frontend koden, eksempelvis ved at minify CSS og JavaScript, som reducerer filstørrelsen og forbedrer ydeevnen i produktionsmiljøet.

Den officielle dokumentation kan tilgås via <https://docs.npmjs.com/>

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 der arbejdes 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 af composer pakker og lignende afhængigheder, alt sammen ved hjælp af én enkelt kommando og på ganske kort tid.

Composer

Composer er et kraftfuldt værktøj til PHP, som gør det muligt 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, eller den specielt integrerede debug bar som der er gjort nytte af under udviklingsfasen for effektivt at fejlsøge under udfordringer. Logging framework, Debugbar og flere tredjeparts biblioteker bliver dokumenteret under deres eget emne senere i dokumentationen.

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

Composer har egen dokumentation som findes på <https://getcomposer.org/doc/>

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.

Stripe kan findes via <https://stripe.com/> men yderligere links til platform samt oversigt er dokumenteret i brugervejledningen hvor det beskrives hvordan man som administrator har overblik over alle betalinger både i det indbyggede Adminpanel samt direkte på Stripe.

Laravel Debugbar

For at effektivisere fejlfinding under udviklingsfasen benyttes der Laravel Debugbar. Dette værktøj giver en dybere indsigt i applikationens SQL-kald, diverse views tilknytning til controllers, routes og meget mere, direkte i browseren ([Se bilag: Debugbar s. 23](#))

Laravel debugbar har ikke en officiel website, men er open-source og kommer med en omfattende dokumentation som kan findes via GitHub på følgende url <https://github.com/barryvdh/laravel-debugbar>

ActivityLog

For at lette integrationen med at logge brugeraktivitet og vigtige hændelser på shoppen, anvendes der ActivityLog biblioteket. Dette giver mulighed for at logge diverse detaljer og hændelser i hele applikationen og sikrer at logging er skalerbart og vedvarende tilgængelig, også i tilfælde af at webshoppen udvides eller ændrer størrelse.

ActivityLog er udviklet af Spatie og den officielle dokumentation findes på

<https://spatie.be/docs/laravel-activitylog/v4/introduction>

laravel-Excel

Laravel Excel er et pakkebibliotek til Laravel som giver mulighed for at arbejde med Excel data på en enkel og effektiv måde, nogle af de fordele som biblioteket tilbyder er blandt andet et tæt samarbejde med Eloquent, Laravel's indbyggede ORM, som gør det let at Importere og Eksportere direkte fra Excel ark til database via projektets modeller.

Yderligere er der indbygget kø-funktionalitet i biblioteket som blandt andet optimerer indlæsning af store datasæt hvilket betyder at store filer kan håndteres i baggrunden uden at påvirke webshoppen.

Sidst men ikke mindst bidrager Excel biblioteket med understøttelse for data validering under importprocessen som hjælper med at sikre at den indkommende data overholder de forventede formater og krav.

Laravel-Excel har egen website og en gennemarbejdet dokumentation som kan tilgås på

<https://laravel-excel.com/>

Xdebug

Xdebug er et værktøj som giver php udviklere mulighed for at fejlsøge kode effektivt, nogle af de fordele som Xdebug bringer er understøttelse for break points og step-debugging i koden, dette giver mulighed for at observere funktioners adfærd i realtid.

Xdebug kræver opsætning direkte på udviklingsserveren, dokumentationen som er benyttet under opsætningen i dette projekt kan findes på <https://xdebug.org/docs/install>

DigitalOcean

DigitalOcean er en cloud-tjenesteudbyder, der leverer et væld af værktøjer, hvilket muliggør hurtig etablering af servermiljøer online med en betalingsstruktur baseret på forbrug og tidsanvendelse frem for faste månedlige gebyrer. En central del af denne opsætning er brugen af en "Droplet", som er navnet på en virtuel Docker-container med en tilknyttet IP-adresse.

I dette setup har der været konfigureret en Ubuntu 22.04 server udstyret med en Apache web-server og Docker. Docker kører webshop-projektet "lokalt" på serveren, og derefter kobles den lokale IP-adresse til en offentlig IP. Dette gør det muligt at præsentere webshoppen i et live-miljø, i modsætning til et begrænset udviklingsmiljø.

Digital Ocean kan findes via følgende url: <https://www.digitalocean.com/>

Design og implementering

I dette afsnit beskrives hvilke designmønstre projektet anvender samt de særlige funktionaliteter der er implementeret i webshoppen for at give en gennemført og stabil applikation.

MVC designmønster

Laravel anvender Model-View-Controller (MVC) arkitekturen, et designmønster som øger separation og giver kodebasen høj modularitet.

Model

Modellerne håndterer al data logik og database interaktion, dette omfatter hentning, indsætning, opdatering og sletning af data i databasen. I projektet er modeller som 'Produkt', 'Order' og 'User' lavet for at håndtere entiteter og relationer.

View

Views er ansvarlige for præsentation og indeholder den HTML som bliver sendt til klientens browser, baseret på den data som modtages fra controlleren. Ved hjælp af Laravel's Blade templating engine bliver der leveret dynamisk brugergrænseflade og produktvisning til klienten.

Controller

Her bearbejdes indgående forespørgsler og udføres logikken som håndterer data inden den videresendes til views.

Service Provider

Denne del er essentiel i Laravel for at binde forskellige komponenter af applikationen sammen, det er blandt andet for at registrere serviceklasser i Laravel's service container som resulterer i en ren måde at implementere Dependency Injection på. Dependency Injection sørger automatisk for at give applikationen de ressourcer og tjenester det har brug for, for at fungere effektivt.

Middleware

I Laravel giver Middlewares mulighed for at håndtere handlinger i Request Lifecycle. Projektet anvender middleware til at håndtere autentifikation og sikre at kun autentificerede brugere kan tilgå bestemte sektioner af webshoppen, som eksempelvis administrationspanelet.

Observer Pattern

Dette benyttes via event og listener klasser for at håndtere handlinger som reagerer på visse hændelser, såsom at logge en hændelse når en administrator logger ind eller ændrer en bog fra administrationen.

Soft Deletes

Der er implementeret en funktion kendt som 'Soft Deletes' gennem Laravel's Eloquent ORM. Dette gør at vi kan bevare data i databasen, selv efter at den er slettet af en administrator. Faktisk bliver denne data ikke fysisk fjernet fra databasen, i stedet markeres dataen med tidspunkt for sletning, hvilket gør det muligt at gendanne data på et senere tidspunkt, dette er også muliggjort fra administrationspanelet, hvis nødvendigt.

Denne del er særlig nyttig i håndtering af data hvor historiske data kan være vigtige. Ved at benytte Soft Deletes, sikrer vi at vigtige data ikke tabes, dette giver os en mere fejltolerant arkitektur.

Factories og Seeders

For at effektivisere udviklingsprocessen og sikre grundig testning af webshoppen er der implementeret brug af Factories og Seeders, som giver mulighed for nemt og hurtigt at kunne oprette realistisk testdata på shoppen.

Factories

Laravel's model factories tillader at definere skabeloner for hvordan hver type af data, såsom hvordan brugere og bøger skal genereres.

På den måde kan der opsættes modeller for at skabe konsistent og realistisk data som kan bruges både til udvikling, tests og automatiserede tests på shoppen.

Seeders

Laravel Seeders avendes til at indlæse data ud fra de factories der er defineret, ind i databasen, hvilket gør det muligt at berige webshoppen med relevant data ved hjælp at kun en enkelt kommando. Denne del er især vigtig i de tidlige stadier af udviklingen da det giver mulighed for præsentation og giver adgang til hurtig gendannelse af en korrekt og beriget webshop, hvilket er ideelt for tests og kvalitetssikring.

Brugen af factories og seeders sikrer at udviklingsteamet kan arbejde effektivt, samt at webshoppen kan testes grundigt på alle tidspunkter af udviklingscyklussen, hvilket fører til et mere robust og pålideligt slutresultat.

Dokumentation for hvordan webshoppen beriges via Seeding findes under installationsvejledningen i produktrapporten side 20.

Konklusion

Udviklingen af 'Sidernes Verden' har i høj grad forløbet som forventet, på trods af nogle få uventede udfordringer. Blandt disse var implementeringen af ActivityLog, som krævede en nyere version end den langtidssøttede (LTS) version for at understøtte Laravel version 11 som for nyligt blev udgivet. Projektet har også været krævende med hensyn til tid og mængden af nødvendig kode for at implementere funktioner som oversigt, oprettelse, redigering og sletning af enhver entitet. Trods disse udfordringer har det været muligt at skabe en fuldt funktionel administrativ del med omfattende funktionaliteter og god oversigt over webshoppen.

Desværre har det ikke været muligt at opfylde alle krav til responsiv visning på tværs af platforme. Selvom webshoppen fungerer responsivt på tablet og desktops, har mobiltilpasningen været nedprioriteret for at kunne fokusere på at udvikle flere funktioner. Derudover er det ikke lykkedes at implementere "Cascade delete" med soft-delete, kun via permanent sletning har det været mulig, hvilket ikke er helt optimalt. Tidsmæssige begrænsninger har simpelthen forhindret mig i at løse denne udfordring.

Det har været yderst lærerigt at udvikle denne webshop. Jeg sætter stor pris på den tidlige investering i opsætningen af Xdebug og Debugbar, hvilket har gjort fejlfinding betydeligt lettere.

Min erfaring fra tidligere projekter samt læreplads har kun budt på andre PHP-frameworks som Symfony og CodeIgniter og overgangen fra Symfony til Laravel har beriget min tekniske erfaring, især fordi Laravel tilbyder en behagelig udviklingsoplevelse med nem konfiguration og god feedback i form af fejlmeddelelser. Omkring halvdelen af min udviklingstid har jeg dedikeret til at følge Laravel's egen dokumentation, hvilket har været en stor hjælp.

Diskussion

Igennem udviklingen af 'Sidernes Verden' har jeg stødt på flere udfordringer, der potentielt kunne få alvorlige konsekvenser i en rigtig webshop. En af de væsentligste udfordringer var, at webshoppens design fungerer godt for en begrænset mængde data, men skalerer dårligt med større datamængder. Det blev tydeligt, at uden implementering af funktioner som søgning, paginering og filtrering i administrationen, ville det blive uoverkommeligt for en administrator at navigere i en stor datamængde, hvilket kunne resultere i tab af effektivitet og øgede fejl.

Desuden viste tidsplanlægningen sig at være en udfordring da jeg efterfølgende sammenlignede estimeret tidsplan med realiseret tidsplan, da nogle opgaver, såsom tests og implementering af ruter, logisk set passede bedre til at blive udført sideløbende med opsætningen af controllers fremfor som en selvstændig opgave.

Gennem projektet har jeg opnået dybere indsigt i brugen af teknologier som Laravel og dets ecosystem, og jeg har lært betydningen af at have robuste værktøjer til fejlfinding som Xdebug og Debugbar. Disse værktøjer har været uvurderlige ved opståede problemer.

Et element som ikke er nævnt så ofte i dokumentationen, men som har været en stor hjælp under udviklingen er de workflows som er opsat til automatisk tests ved deployment til GitHub, her har jeg en håndfuld gange fået god feedback om forældet kode i forhold til den anvendte PHP-version, det opstår blandt andet da jeg på min læreplads arbejder flere version bag den php version som anvendes her i projektet.

Bilag

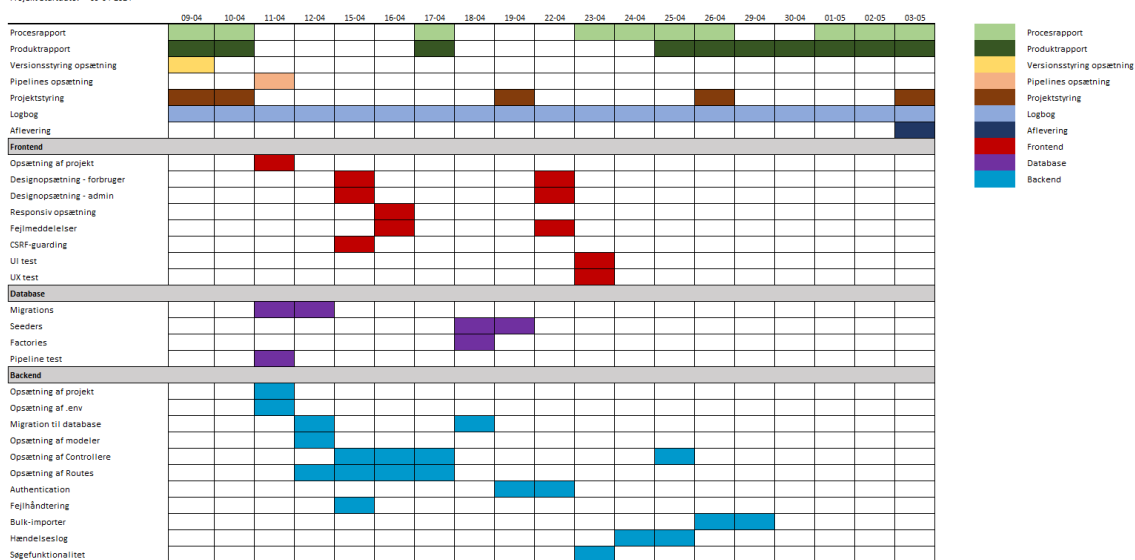
Estimeret tidsplan

Også tilgængelig via url:

[https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Images/Estimeret tidsplan.png](https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Images/Estimeret%20tidsplan.png)

Sidernes Verden - Estimeret tidsplan

Projekt startdato: 09-04-2024



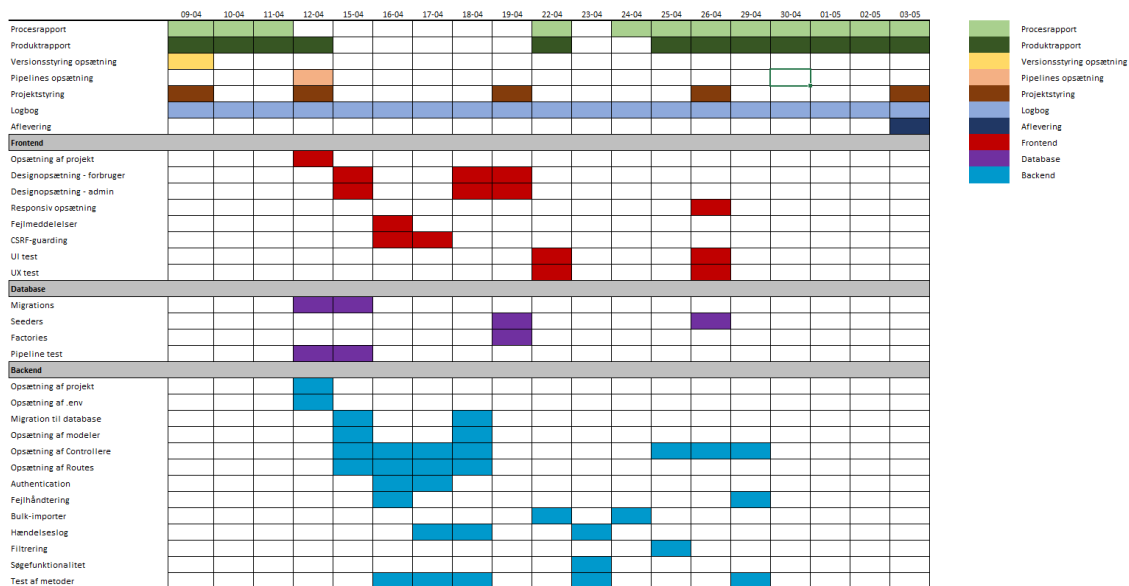
Realiseret tidsplan

Også tilgængelig via url:

https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Images/Realiseret_tidsplan.png

Sidernes Verden - Realiseret tidsplan

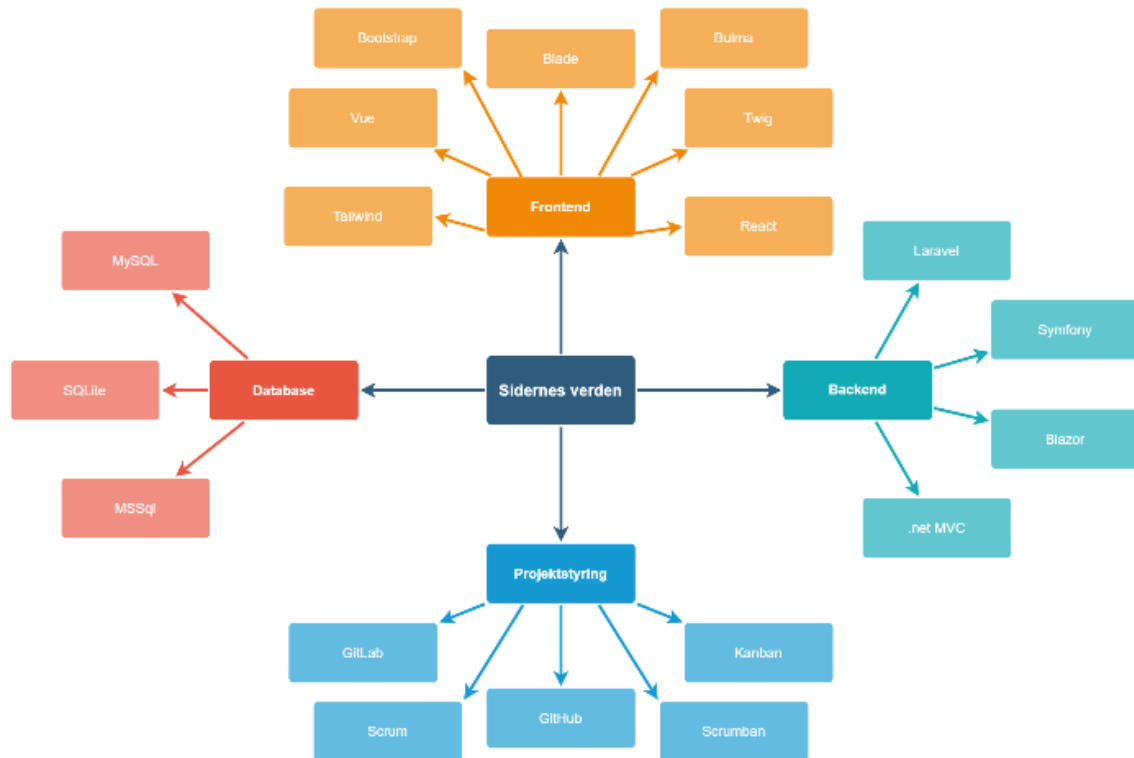
Projekt startdato: 09-04-2024



Mindmap

Også tilgængeligt via url:

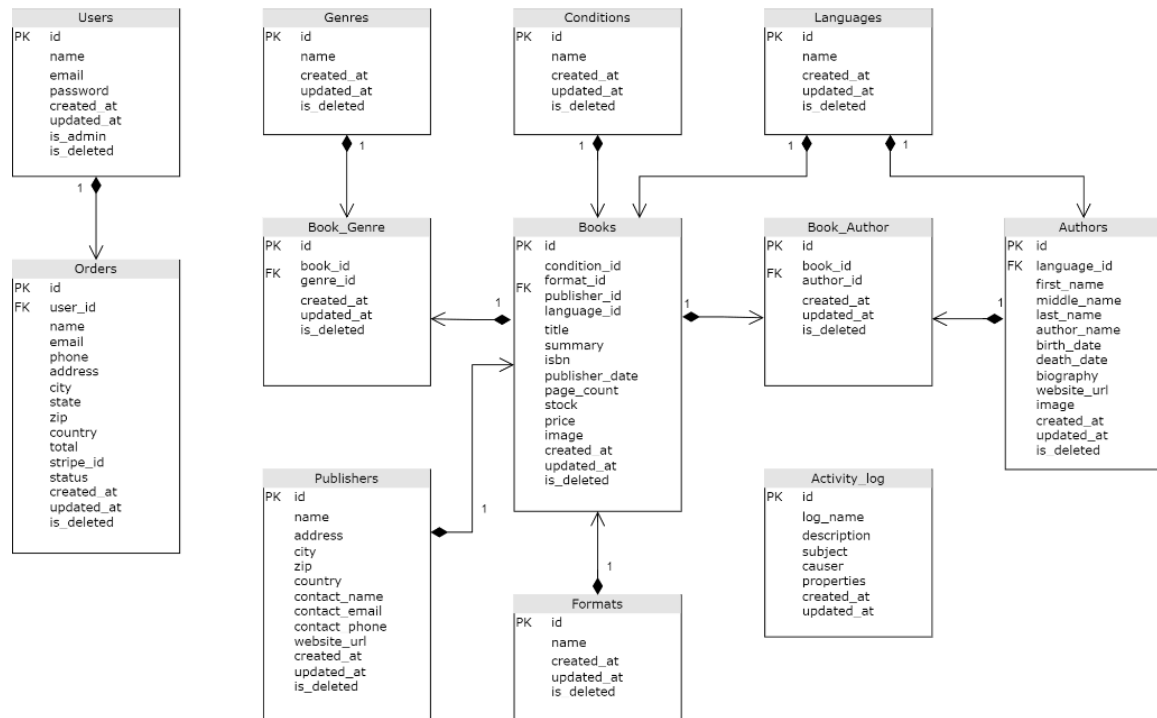
https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/mindmap.drawio.pdf



Databasediagram

Også tilgængeligt via url:

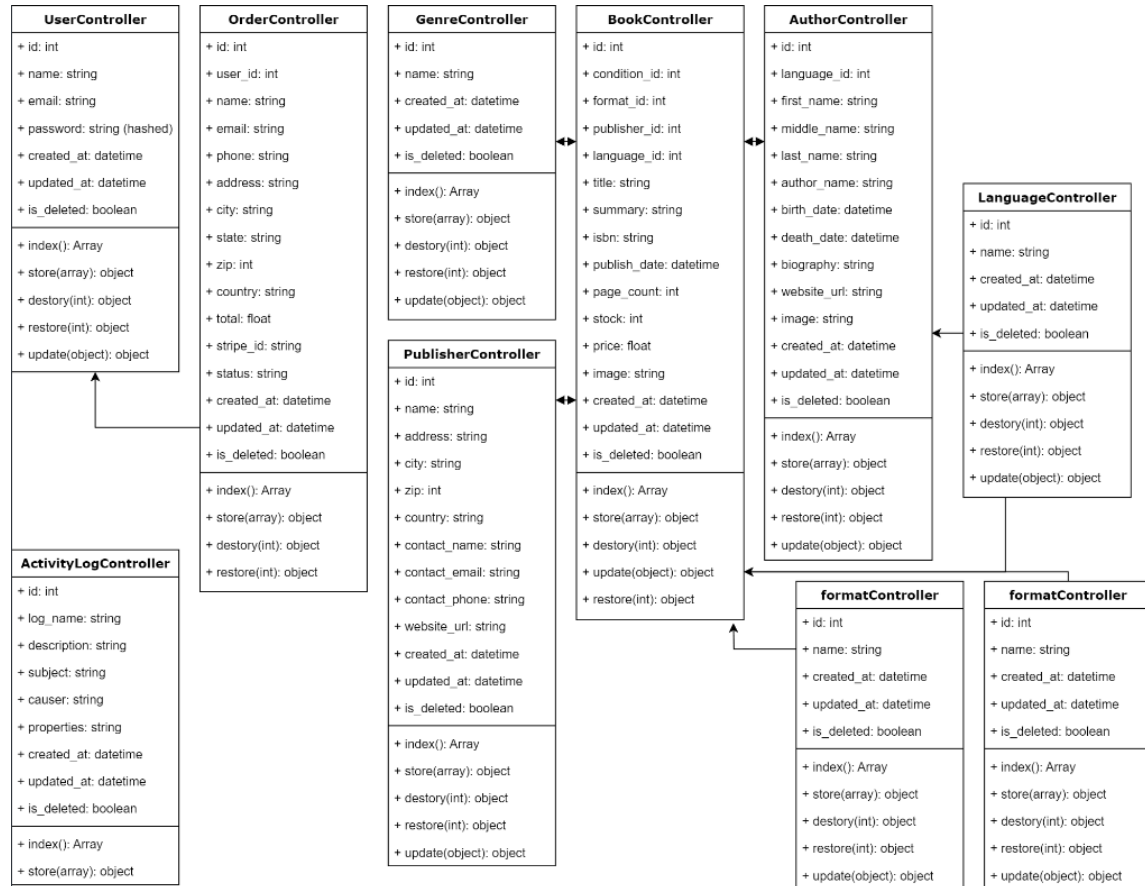
https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/database_diagram.drawio.pdf



Klassediagram

Også tilgængeligt via url:

https://github.com/MichaelAggerholm/Apprentice_Test/blob/main/Assets/Documents/klasse_diagram.drawio.pdf

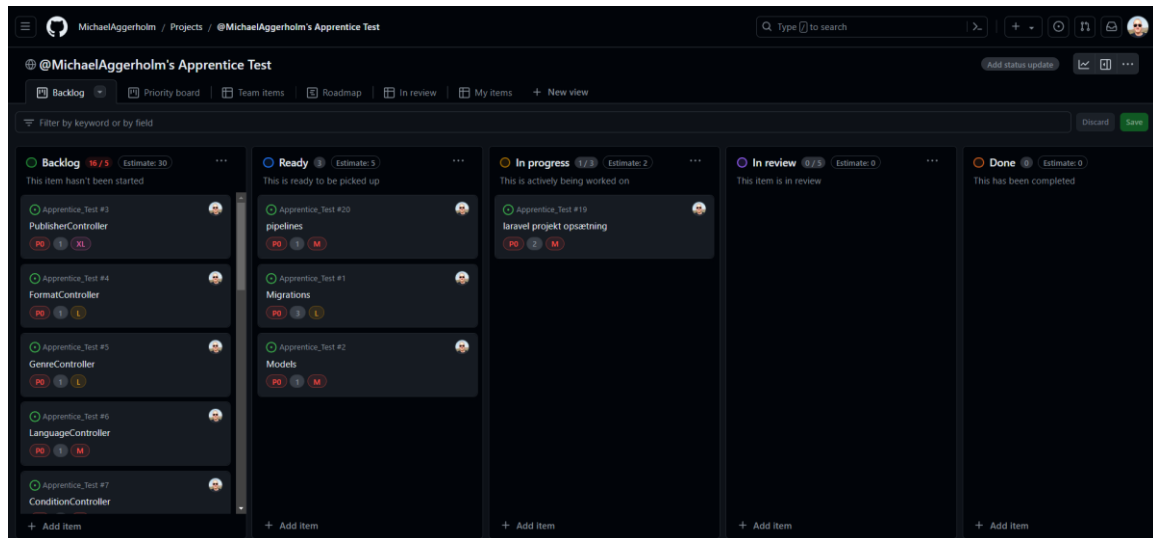


Kanban Fase 1 – opstart af projekt

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

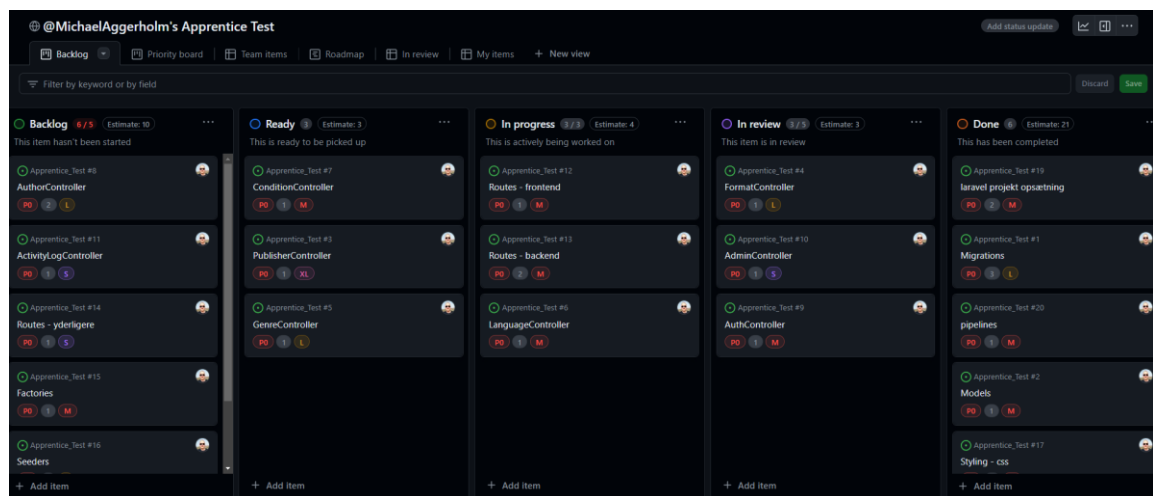
For at se Kanban boardet 's nuværende fase, kan det tilgås via GitHub URL:

<https://github.com/users/MichaelAggerholm/projects/2/views/1>



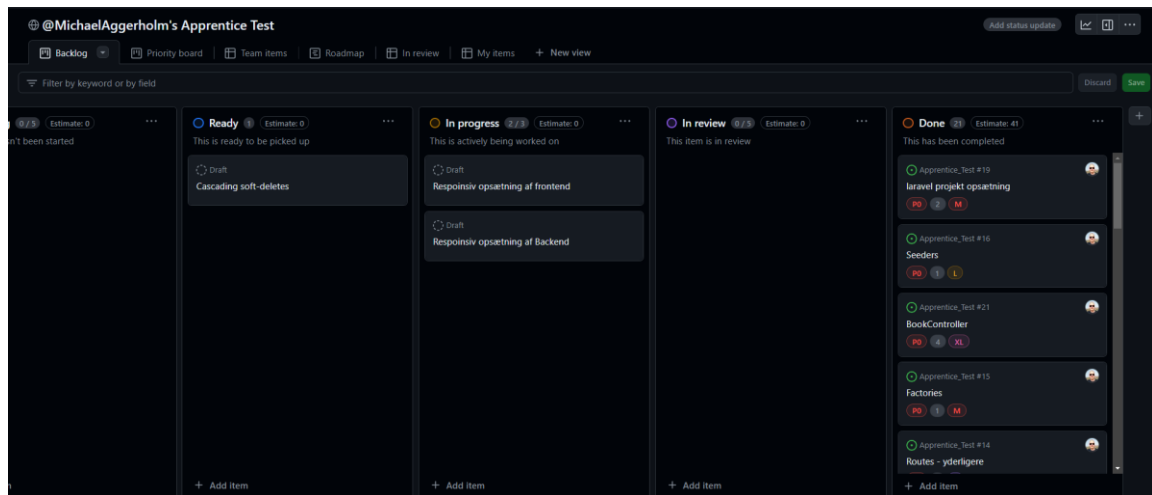
Kanban Fase 2 – midtvejs i projekt

Dette billede fra Kanban boardet er taget ved projektets mid-phase.



Kanban Fase 3 – Afslutning af projekt

Dette billede fra Kanban boardet er taget ved projektets afsluttende fase.







Pipelines kørsel

For at se alle pipelines, kan det tilgås via GitHub Actions URL:
https://github.com/MichaelAggerholm/Apprentice_Test/actions

Actions		All workflows			
New workflow		Showing runs from all workflows			
All workflows		Filter workflow runs			
Workflows		7 workflow runs			
Larastan Code Analysis		Event Status Branch Actor			
Laravel Application Tester		tilføjet workflow badges til readme fil			
PHP Packages Vulnerability Checker		Laravel Application Tester #3: Commit 5ab12c0 pushed by MichaelAggerholm			
Management		tilføjet workflow badges til readme fil			
Caches		Larastan Code Analysis #2: Commit 3ab13d9 pushed by MichaelAggerholm			
Runners		tilføjet workflow badges til readme fil			
		PHP Packages Vulnerability Checker #2: Commit 3ab13d9 pushed by MichaelAggerholm			
		tilføjet workflows og ændret mappestruktur for projektet			
		PHP Packages Vulnerability Checker #1: Commit e7a403c pushed by MichaelAggerholm			
		tilføjet workflows og ændret mappestruktur for projektet			
		Larastan Code Analysis #1: Commit e7a403c pushed by MichaelAggerholm			
		tilføjet workflows og ændret mappestruktur for projektet			
		Laravel Application Tester #2: Commit e7a403c pushed by MichaelAggerholm			
		rettet workflow fra master til main			
		Laravel Application Tester #1: Commit ea7ad44 pushed by MichaelAggerholm			

Laravel Debugger

Vores nyeste bøger



Messages | Timeline | Exceptions | Views | Routes | **Queries** | Models | Mails | Gate | Session | Request

5 statements were executed

7.71ms

GET | 0MB | 3.26s | 8.3.3

Connection Established

select " from "books" where "books"."deleted_at" is null order by "created_at" desc limit 4

5.13ms

/var/www/html/database.db | PagesController.php#29

select "genres"."", "book_genre"."book_id" as "pivot_book_id", "book_genre"."genre_id" as "pivot_genre_id" from "genres" inner join "book_genre" on "genres"."id" = "book_genre"."genre_id" where "book_genre"."book_id" in (1, 2, 5, 6, 7) and "genres"."deleted_at" is null

754µs

/var/www/html/database.db | PagesController.php#32

select " from "conditions" where "conditions"."id" in (1, 2) and "conditions"."deleted_at" is null

618µs

/var/www/html/database.db | PagesController.php#32

select " from "formats" where "formats"."id" in (2) and "formats"."deleted_at" is null

580µs

/var/www/html/database.db | PagesController.php#32

select " from "languages" where "languages"."id" in (1) and "languages"."deleted_at" is null

640µs

/var/www/html/database.db | PagesController.php#32

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 svendepåtegning 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.
Påbegyndt opsættelse af administrative funktioner for tilstande
Udfordringer med at få logging til at virke korrekt, Events bliver ikke triggeret korrekt og event listeners er ændret i Laravel 11 som nyligt er udkommet.

17/04 – Onsdag

Udfordringer med Logging løst, event listeners skulle registreres anderledes i Laravel 11.
Færdiggørelse af administrativ funktionalitet for tilstande.
Færdiggørelse af administrativ funktionalitet for genre.
Færdiggørelse af administrativ funktionalitet for sprog.
Færdiggørelse af administrativ funktionalitet for forfattere.
Færdiggørelse af administrativ funktionalitet for udgivere.
Påbegyndt opsætning af funktionalitet for administration af bøger.

18/04 – Torsdag

Færdiggjort administrative funktioner for bøger.
Oprettelse af kurv.
Designtilpasninger for visning af bøger.
Funktionalitet og logik for kurv, mulighed for tilføjelse af antal, pris, samlet pris, osv.
Test af kurv og bøger samt opsætning af multi-select for bøgeres mange-til-mange.
Påbegyndt opsætning af stripe integration og checkout funktionalitet.

19/04 – Fredag

Færdiggjort opsætning af stripe integration og checkout funktionalitet.
Opsætning af factories og seeders for nem adgang til test data ved ny migration eller skift til anden pc.
Opsætning af views og styling heraf på frontend / Adminpanel.
Gennemgang af Kanban board, opgaver og refleksion over ugens arbejde.

22/04 – Mandag

Påbegyndt opsætning af bulk-import, forsøgt komplet import med bøger, forfattere, udgivere, genre osv. Alt sammen i ét go, det bliver for omfattende, der er ændret retning til at lave bog-import som er afhængig af at forfatter og yderligere allerede eksisterer på siden, hertil bør der muligvis laves en log over hvilke bøger der ikke blev importeret, og hvorfor.
Kort test af UX, virker rigtig overskueligt og enkelt for både forbruger samt administrator.
Kort test af UI, overholder ikke responsiv krav på nuværende tidspunkt, tilsidesættes til efter opgaver med højere prioritet færdiggøres.

23/04 – Tirsdag

Færdiggjort søgefunktion på forsiden, med mulighed for at søge bøger frem på titel, format, tilstand osv.
Grundig test og tilpasning af søgefunktionalitet da der var mere omfattende SQL-Joins end forventet.

24/04 – Onsdag

Færdiggørelse af bulk-importer samt opsætning af hændelses-log eksporter til Excel ark download i browseren.

Opsat administration af brugere fra Adminpanel med mulighed for at give brugere admin. Opfølgning på libraries og implementering i procesrapporten.

25/04 – Torsdag

Opsætning af filtrering på side med alle bøger, samt optimeret søgning for at kunne søge på tilstande, formater og titler mm.

Opfølgning på dokumentation for nye tilføjelser i produkt/procesrapport.

Test af UX for søgning og filtrering.

26/04 – Fredag

Færdiggørelse af alle redigerings funktionaliteter i Adminpanel, således admin kan redigere alle elementer fra panelet.

Dokumentation for installationsvejledning.

Tilpasset visning af bøger på forsiden samt opsat skråbanner for brugte bøger.

Test af UI på forsiden.

Opfølgning på Kanban boardet.

29/04 – Mandag

Opsætning af live web server gennem <https://www.digitalocean.com/> samt opsætning af webshop og "go-live" til produktionsmiljø fremfor Development konfiguration.

30/04 – Tirsdag

01/05 – Onsdag

02/05 – Torsdag

03/05 – Fredag