

Documentatie basic_project

Requirements	2
Hoe is dit project gebouwd?	3
Stap 1: Gebruikers Registreren	3
Stap 2: Installatie van tabellen, en Dashboard	4
Stap 3: Inloggen	4
Stap 4: Content pagina's	5
Stap 5: Afwerking	6
Hoe installeer je het project?	6
Waarom requirement first werken?	7
Overige informatie	7
Backup en verwijderen van install.php	7
Fair Use	7
Over de auteur/developer	7

Requirements

1. **Gebruikersregistratiesysteem:**

- Maak een gebruikersregistratieformulier met velden voor gebruikersnaam, e-mail en wachtwoord.
- Implementeer server-side validatie met PHP om de integriteit van de gegevens te waarborgen.
- Sla geregistreeerde gebruikersinformatie veilig op in een MySQL-database.

2. **Gebruikersauthenticatie:**

- Ontwikkel een inlogsysteem waar gebruikers kunnen inloggen met hun e-mail en wachtwoord.
- Implementeer sessiebeheer om de staat van de gebruiker over verschillende pagina's te behouden.

3. **Dynamisch Inhoud Weergeven:**

- Maak een pagina die inhoud uit de database weergeeft (bijv. lijst van door gebruikers ingediende berichten of commentaren).
- Gebruik PHP om gegevens uit de database op te halen en gestructureerd weer te geven.

4. **Integratie van JavaScript:**

- Voeg interactieve functies toe met JavaScript of jQuery (bijv. formuliervalidatie, dynamische inhoudsupdates).
- Implementeer een AJAX-functie, zoals live zoeken of asynchrone formulierinzending, die interageert met een PHP-backend.

5. **Beveiligingsoverwegingen:**

- Zorg ervoor dat uw applicatie beveiligd is tegen veelvoorkomende kwetsbaarheden zoals SQL-injectie, XSS en CSRF.
- Sanitize en valideer alle gebruikersinvoer en -uitvoer.

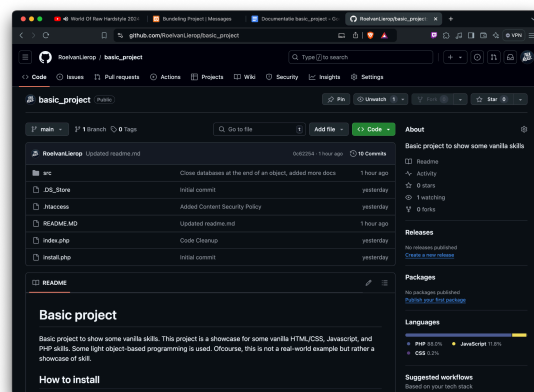
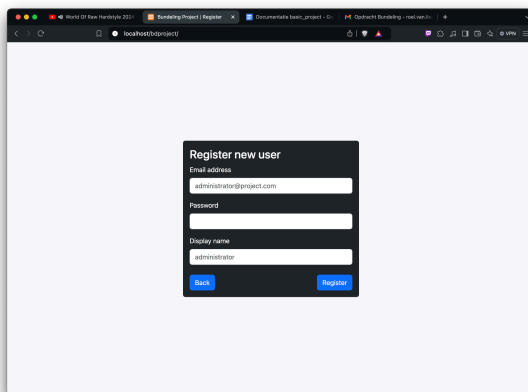
Hoe is dit project gebouwd?

Allereerst, welkom in de functionele documentatie voor basic_project. Een project op basis van (veelal) vanilla code, en een project om te zien waar mijn skills liggen op basis van deze code.

Het project is opgebouwd op basis van requirements first. Dat wil zeggen dat er een beginpunt is gekozen, en dat vanuit dat punt alle requirements zijn ingebouwd en aangevuld. Om een concreet voorbeeld te geven vanuit ons startpunt. Allereerst is er vanuit de requirements lijst gekeken welk punt het beste aanknopingspunt was om te beginnen. In dit geval het registreren van gebruikers. Wat hebben we daarvoor nodig, in combinatie met overige requirements. Vanuit daar ging het snel, en kon ik een natuurlijk flow volgen om het project op te bouwen.

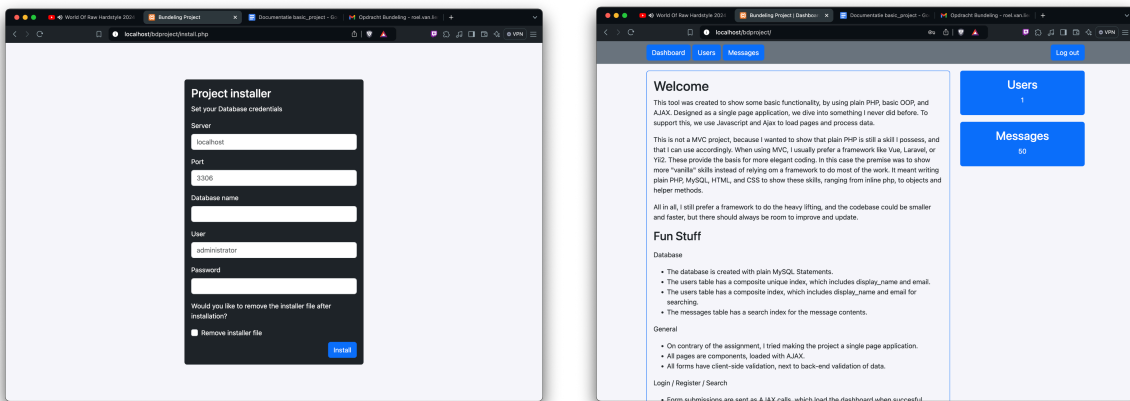
Stap 1: Gebruikers Registreren

- Bouw een single-page application, beginnende met het registratieformulier.
- Doe front-end validatie, voordat het formulier met AJAX naar enige back-end code wordt verzonden.
- Maak een object om een CSRF mee te geven en te vergelijken, en het formulier af te maken.
- Maak een index pagina die verwijst naar het registratieformulier.
- Sla de informatie op in een GitHub repository.
- **Yields:** *index.php, registratieformulier, CSRF object, app.js. GIT repository*



Stap 2: Installatie van tabellen, en Dashboard

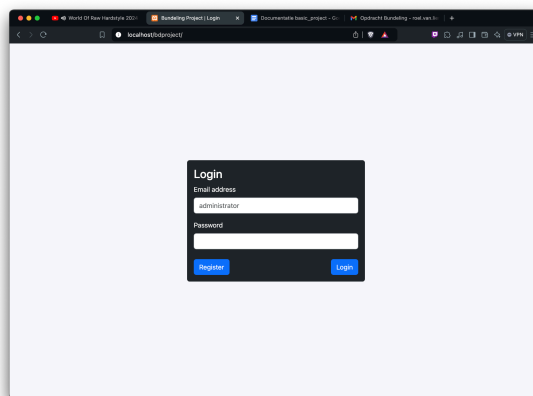
- Bouw een installer zodat het project een plek heeft om een migratie uit te voeren die de database vult met tabellen.
- Registreer gebruikers met het formulier gemaakt in stap 1, sanitize de informatie uit het formulier voordat het opgeslagen wordt. Sla vervolgens de gebruiker op in een sessie, zodat we deze kunnen gebruiken voor verificatie.
- Bouw een Dashboard pagina waarheen verwezen kan worden na registratie, met een log out button, die weer terug verwijst naar de index, en de sessie vernietigd.
- **Yields:** *Installer.php, Installer object, Users object, users tabel, dashboard pagina. logout flow.*



Met deze stappen kunnen we een gebruiker registreren, en verwijzen naar het dashboard. Echter kunnen we deze gebruiker nog niet inloggen, dat gebeurt in de volgende stappen.

Stap 3: Inloggen

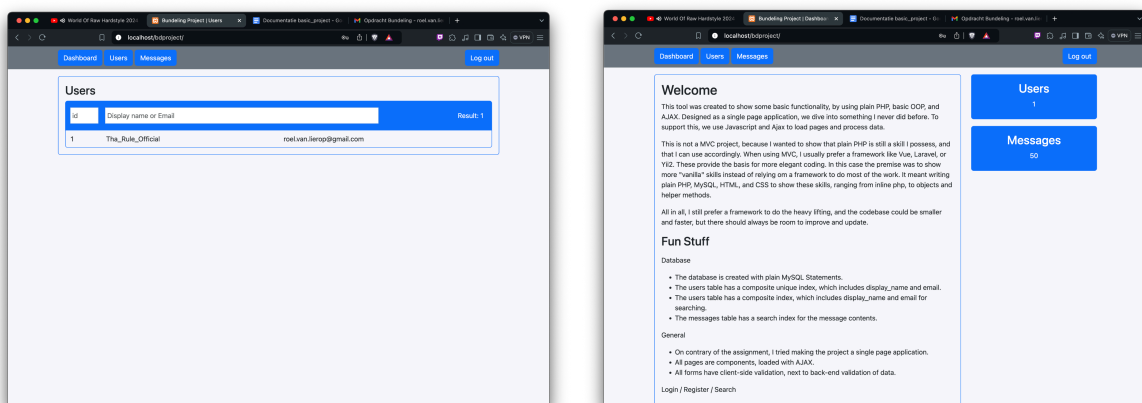
- Bouw een login formulier, en vervang de verwijzing in index.php naar het login formulier. Op dit login formulier staat een registratieknop, die verwijst naar de registratiepagina.
- Gebruik dezelfde front-end en back-end verificatie voor het inloggen als de registratie, ook de sessie van de gebruiker moet hier weer ingesteld worden indien de gegevens kloppen.
- **Yields:** *Login formulier*



Daarmee is het inloggen en registreren van een gebruiker op functioneel gebied klaar. We maken in de vierde stap nog even een aantal berichten aan, en de pagina's binnen het dashboard om alle gegevens weer te geven.

Stap 4: Content pagina's

- Breid het installer object uit. Schrijf de code om berichten in de database te zetten. We maken 50 random berichten aan op basis van user ID 1 (de eerst geregistreerde gebruiker).
- Maak een message object waarmee we informatie uit de database kunnen halen, en doorzoekbaar kunnen laten zijn.
- Breid het User object ook uit om dezelfde zoekcriteria te kunnen gebruiken.
- Maak op het dashboard de pagina's voor het weergeven van de gebruikersgegevens, en de berichten.
- Maak op de gebruikers en berichten pagina's ook de zoekmogelijkheden beschikbaar. Deze hoeven niet per definitie een front-end check te hebben. Een back-end check is wel handig.
- Maak knoppen met het aantal gebruikers en berichten weergeven op het dashboard, zo kunnen we de aantallen van de data ook zien.
- **Yields:** *Uitbreiding installer, messages tabel, Message object, uitbreiding User object, uitbreiding dashboard, users page, messages page.*



Stap 4 geeft ons de read-only pagina's en de doorzoekbaarheid van de data in de database. Nu rust alleen de afwerking nog, die we in stap 5 doen;

Stap 5: Afwerking

- Update de installer en maak het installatiescript onbereikbaar na installatie (Geef de beheerder/installer ook de keuze om deze stap te volgen).
- Maak de welkomsttekst op het dashboard.
- Test alle formulieren en zoekmogelijkheden nog een keer.
- Schrijf de technische documentatie (PHPDoc blocks, commentaren).
- Schrijf de readme.md.
- Schrijf de functionele documentatie (dit document).

```
/**
 * createTables Method
 *
 * Small method to streamline the creation of database tables
 *
 * @uses Installer::createUserTables()
 * @uses Installer::createMessageTables()
 */
public function createTables(): void
{
    // Call table creation methods
    $this->createUserTables();
    $this->createMessageTables();
}
```

Door PHPDoc blokken te gebruik icm commentaar kunnen we ieder klein stukje code makkelijk omschrijven, wat het voor andere ontwikkelaars makkelijker maakt om de code in te zetten.

Hoe installeer je het project?

De installatie is omdat het project geen gebruik maakt van Node, Composer, of andere back-end technologie, vrij simpel.

- Maak een database aan. Deze hebben we nodig om onze informatie in op te slaan.
- Maak een projectmap aan op een testserver, en haal het project binnen met GIT.
git clone https://github.com/RoelvanLierop/basic_project.git
- Ga naar de homepage van jouw project in een browser naar keuze, en navigeer naar /installer.php, volg daar het installatieproces.
- Wanneer dat gedaan is krijg je het login scherm te zien. Registreer eerst nog even een gebruiker door op "Register" te klikken, na registratie krijg je het Dashboard te zien en kun je bij de gegevens.

Waarom requirement first werken?

Ik ben nogal chaotisch van aard, ondanks dat ik secuur en snel kan werken. Door een beginpunt en eindpunt te definiëren, kan ik de tussenstappen invullen. Soms gaat dit niet omdat een ontwikkelproces zelden in een rechte lijn plaatsvindt, of een opdracht te groot is waardoor het onderverdeeld moet worden in kleinere subprojecten. Maar in 90% van de gevallen helpt een rode draad mij om een update, of heel project in dit geval tot een goed einde te brengen. Bij iedere stap ga ik terug naar de requirements om te zien of een stap het gewenste effect heeft. Door je requirements (of user stories in een SCRUM/Agile omgeving) te definiëren, weet het team waar hij/zij aan toe is. In het geval van dit project werk ik alleen, en is deze werkwijze te bloated. Daarom heb ik een subset van de requirements gepakt en ben begonnen met het toevoegen van gebruikers.

Overige informatie

Overige informatie over het project

Backup en verwijderen van install.php

In install.php is op regel 28 een waarschuwing ingebouwd over het verwijderen van install.php. Wanneer je tijdens het installeren kiest om install.php te verwijderen, zal er een backup worden gemaakt, en install.php worden verwijderd. Dit is om te zorgen dat de installatie niet meermaals wordt gedraaid. Wanneer je van regel 29 een comment maakt, wordt install.php niet verwijderd, ook al kies je er voor om dat wel te doen (de backup wordt nog steeds gemaakt).

Fair Use

Alle code in dit project blijft eigendom van de ontwikkelaar die het gemaakt heeft (ik), en mag ik daardoor dan ook gebruiken zoals ik nodig acht. Echter, vanuit development oogpunt zal dit project weinig updates krijgen, omdat een framework qua werktempo toch mijn voorkeur heeft. Daarnaast staat het opdrachtgevers vrij om deze code in te zetten in hun projecten, naast developers die de repository clonen.

Over de auteur/developer

Meer informatie over deze repository kunt u krijgen door de readme.md te lezen, of contact op te nemen met de ontwikkelaar van deze repository. Wiens naam en e-mailadres in de code omschreven zijn in het Author veld van de phpdoc blokken.

Roel van Lierop <roel.van.lierop@gmail.com>