

B3 – Eisenpakket sprint 2

2.1 Als gebruiker wil ik dat taken ook een deadline hebben, in de overzichten komt deze er ook bij te staan en is de lijst bovendien op deadline gesorteerd. Met deadline bedoelen we alleen een datum, zonder tijd dus.

⚠ *Technische tip:* gebruik een input van het type “date”, deze geeft aan PHP een waarde terug die meteen geschikt is om in de database op te slaan in een DATE-kolom.

- ❑ Pas het formulier in `tasks/create.php` aan, zodat je hier ook een deadline kunt invoeren.
- ❑ Pas de create-actie in de controller aan zodat de deadline vanuit het formulier ook echt opgeslagen wordt (dus opnemen in de validatie en daarna de query). Test goed of dit werkt.
- ❑ Pas het formulier in `tasks/edit.php` aan, zodat je hier ook de deadline kunt aanpassen.
- ❑ Pas de update-actie in de controller aan zodat de deadline vanuit het formulier ook echt opgeslagen wordt (dus opnemen in de validatie en daarna de query). Test goed of dit werkt.
- ❑ In `tasks/index.php`: pas de query aan zodat deze sorteert op deadline. Toon de deadline ook bij iedere taak in het overzicht.
- ❑ In `tasks/done.php`: pas de query aan zodat deze sorteert op deadline. Toon de deadline ook bij iedere taak in het overzicht.

2.2 Als gebruiker wil ik alleen de taken van één afdeling kunnen bekijken. Dit gebeurt op een nieuwe pagina, die verder wel lijkt op het bestaande takenoverzicht.

⚠ *Technische tip:* maak een pagina die een waarde uit de URL leest met behulp van de variabele `$_GET`. Dus als ik bijvoorbeeld ga naar `tasks/afdeling.php?afdeling=horeca`, dan zie ik alleen de taken van de afdeling horeca. Je moet dus zelf de juiste URL invullen.

⚠ *Technische tip:* je haalt dus de waarde ‘afdeling’ uit de GET-variabele. Die waarde gebruik je dan voor de WHERE in je query. Behalve dat stuk is je pagina helemaal gelijk aan `tasks/index.php`.

2.3 Als gebruiker wil ik kunnen inloggen met een bestaand account. Als het inloggen slaagt kom ik op de homepage terecht.

⚠ *Technische tip:* de aangeleverde database bevat drie testaccounts met als gebruikersnaam “user1” t/m 3 en als wachtwoord “password1” t/m 3.

- ❑ Maak een inlogpagina met daarop een formulier.
- ❑ Maak een `loginController.php` die het inloggen afhandelt.
- ❑ Als het inloggen is geslaagd, sla dan gegevens op in de sessie.
- ❑ Als het inloggen is geslaagd, redirect dan naar de homepage.

2.4 Als opdrachtgever wil ik dat alle pagina’s beveiligd zijn, dus alleen te bezoeken door een ingelogde gebruiker.

2.5 Als gebruiker wil dat bij een nieuwe taak mijn user-id wordt opgeslagen.

⚠ *Technische tip:* in de taken-tabel kun je het user-id opslaan in de kolom “user”. Je haalt het user-id natuurlijk uit de SESSION, op het moment dat je echt een INSERT-query gaat uitvoeren.

2.6 Als gebruiker wil ik een pagina met daarop alleen de taken die ik zelf heb gemaakt.

⚠ *Technische tip:* deze opdracht lijkt een beetje op 2.2, maar dan haal je nu het user-id uit de SESSION om die te gebruiken in de WHERE. Je maakt dus een nieuwe pagina zoals `tasks/my.php`.

⚠ Het is handig om vanaf het grote takenoverzicht heen-en-weer te linken naar deze pagina.