**Functioneel Ontwerp (FO)**

**Inleiding**

**Doel van het project**

Het doel van dit project is om een eenvoudige, maar functionele website te ontwikkelen voor het bedrijf Web-Essentials. De website biedt gebruikers dagelijkse tools, namelijk een rekenmachine, een agenda en een to-do lijst. Deze tools moeten eenvoudig te gebruiken zijn en consistent zijn qua design, met de mogelijkheid om in de toekomst meer tools toe te voegen.

**Scope van de applicatie**

De scope van de applicatie omvat:

* Een rekenmachine met standaard functies.
* Een agenda die afspraken bijhoudt, sorteert en bewerken mogelijk maakt.
* Een to-do lijst die taken kan toevoegen, bewerken en afvinken.
* Een uniform design voor alle tools met de bedrijfsnaam in de header en footer.
* Mogelijkheid om in de toekomst meer tools toe te voegen.

**Betrokken stakeholders**

* Web-Essentials (opdrachtgever)
* Eindgebruikers (klanten van Web-Essentials)
* Ontwikkelteam

**Functionele eisen**

**Gedetailleerde beschrijving van de eisen van de gebruiker**

* **Rekenmachine**: Gebruikers moeten eenvoudige en complexe berekeningen kunnen uitvoeren, inclusief optellen, aftrekken, vermenigvuldigen, delen, en clearen van de invoer.
* **Agenda**: Gebruikers moeten afspraken kunnen toevoegen, bewerken en verwijderen. Elke afspraak bevat een datum, tijd en beschrijving. Afspraken moeten chronologisch weergegeven worden.
* **To-Do lijst**: Gebruikers moeten taken kunnen toevoegen, bewerken en verwijderen. Taken kunnen als voltooid gemarkeerd worden, wat zichtbaar moet zijn door doorgestreepte tekst.
* **Uniform design**: Alle tools moeten een consistent design hebben met de bedrijfsnaam in de header en footer.

**Use cases**

* **Rekenmachine**:
  + Als gebruiker wil ik een eenvoudige rekensom kunnen uitvoeren zodat ik snel een antwoord krijg.
  + Als gebruiker wil ik de invoer kunnen wissen zodat ik opnieuw kan beginnen.
* **Agenda**:
  + Als gebruiker wil ik een nieuwe afspraak kunnen toevoegen zodat ik mijn planning kan bijhouden.
  + Als gebruiker wil ik een bestaande afspraak kunnen bewerken zodat ik wijzigingen kan aanbrengen.
  + Als gebruiker wil ik een afspraak kunnen verwijderen zodat mijn agenda up-to-date blijft.
* **To-Do lijst**:
  + Als gebruiker wil ik een nieuwe taak kunnen toevoegen zodat ik mijn taken kan bijhouden.
  + Als gebruiker wil ik een taak kunnen bewerken zodat ik wijzigingen kan aanbrengen.
  + Als gebruiker wil ik een taak als voltooid kunnen markeren zodat ik mijn voortgang kan zien.

**Procesbeschrijvingen**

**Overzicht van de belangrijkste processen en workflows**

* **Rekenmachine**: Gebruikers voeren nummers en operatoren in, drukken op de "=" knop voor het resultaat, en kunnen de invoer wissen met de "C" knop.
* **Agenda**: Gebruikers voegen een afspraak toe door datum, tijd en beschrijving in te voeren, kunnen afspraken bewerken door bestaande gegevens aan te passen, en kunnen afspraken verwijderen.
* **To-Do lijst**: Gebruikers voegen een taak toe door een beschrijving in te voeren, bewerken een taak door bestaande gegevens aan te passen, en markeren taken als voltooid.

**Stappen die gebruikers nemen om taken te voltooien**

* **Rekenmachine**:
  1. Open de rekenmachine.
  2. Voer nummers en operatoren in.
  3. Druk op de "=" knop voor het resultaat.
  4. Gebruik de "C" knop om de invoer te wissen.
* **Agenda**:
  1. Open de agenda.
  2. Klik op "Voeg afspraak toe".
  3. Vul de datum, tijd en beschrijving in.
  4. Klik op "Opslaan".
  5. Om een afspraak te bewerken, klik op de afspraak en pas de gegevens aan.
  6. Om een afspraak te verwijderen, klik op de verwijderknop bij de afspraak.
* **To-Do lijst**:
  1. Open de to-do lijst.
  2. Klik op "Voeg taak toe".
  3. Vul de beschrijving in.
  4. Klik op "Opslaan".
  5. Om een taak te bewerken, klik op de taak en pas de gegevens aan.
  6. Om een taak als voltooid te markeren, vink het selectievakje naast de taak aan.

**Navigatiestructuur**

**Sitemap**

* **Homepage**: Overzicht van de beschikbare tools.
* **Rekenmachine**: Pagina met de rekenmachine.
* **Agenda**: Pagina met de agenda.
* **To-Do lijst**: Pagina met de to-do lijst.

**Wireframes of mockups**

* **Homepage**: Header met bedrijfsnaam, links naar de rekenmachine, agenda en to-do lijst, footer met bedrijfsnaam.
* **Rekenmachine**: Numpad met standaard functies, display voor invoer en resultaten.
* **Agenda**: Tabel met afspraken, formulier voor het toevoegen/bewerken van afspraken.
* **To-Do lijst**: Lijst met taken, formulier voor het toevoegen/bewerken van taken, mogelijkheid om taken af te vinken.

**Functionaliteit per scherm**

**Homepage**

* Links naar de verschillende tools.
* Bedrijfsnaam in header en footer.

**Rekenmachine**

* Numpad met nummers en operatoren (+, -, \*, /).
* Display voor invoer en resultaten.
* Knop om invoer te wissen.

**Agenda**

* Tabel met afspraken, gesorteerd op datum en tijd.
* Formulier voor het toevoegen van nieuwe afspraken.
* Mogelijkheid om afspraken te bewerken en te verwijderen.

**To-Do lijst**

* Lijst met taken, met mogelijkheid om taken af te vinken.
* Formulier voor het toevoegen van nieuwe taken.
* Mogelijkheid om taken te bewerken en te verwijderen.

**Rapportages en exports**

**Agenda**

* Mogelijkheid om een lijst van afspraken te exporteren naar een CSV-bestand.

**To-Do lijst**

* Mogelijkheid om een lijst van taken te exporteren naar een CSV-bestand.

**Technisch Ontwerp (TO)**

**Inleiding**

**Overzicht van de technologieën en tools**

* **Frontend**: HTML, CSS, JavaScript
* **Backend**: PHP, MySQL
* **Frameworks**: Laravel (PHP framework)
* **Tools**: Git, Trello, PHPUnit (voor unit testing)

**Doel van het technisch ontwerp**

Het technisch ontwerp beschrijft hoe de functionele eisen geïmplementeerd zullen worden met behulp van de gekozen technologieën en tools. Het richt zich op de technische aspecten en de architectuur van de applicatie.

**Architectuur**

**Beschrijving van de algemene architectuur**

* **MVC-architectuur**: De applicatie zal gebruikmaken van het Model-View-Controller (MVC) ontwerpprincipe, waarbij de business logica, de gebruikersinterface en de controlelogica gescheiden worden.

**Componentendiagrammen**

* **Frontend**: HTML, CSS, JavaScript (Vue.js)
* **Backend**: PHP (Laravel), MySQL database

**Datamodel**

**ER-diagrammen**

* **Gebruikers**: id, naam, email, wachtwoord
* **Afspraken**: id, user\_id, datum, tijd, beschrijving
* **Taken**: id, user\_id, beschrijving, voltooid

**Tabellenstructuur**

* **Gebruikers**:
  + id: int, primary key, auto-increment
  + naam: varchar(255)
  + email: varchar(255), unique
  + wachtwoord: varchar(255)
* **Afspraken**:
  + id: int, primary key, auto-increment
  + user\_id: int, foreign key (link naar gebruikers)
  + datum: date
  + tijd: time
  + beschrijving: text
* **Taken**:
  + id: int, primary key, auto-increment
  + user\_id: int, foreign key (link naar gebruikers)
  + beschrijving: text
  + voltooid: boolean

**Technische specificaties**

**Gedetailleerde beschrijving van de implementatie**

* **Rekenmachine**: JavaScript voor het verwerken van invoer en het uitvoeren van berekeningen.
* **Agenda**: Laravel controllers voor het beheren van afspraken, MySQL database voor het opslaan van afspraken.
* **To-Do lijst**: Laravel controllers voor het beheren van taken, MySQL database voor het opslaan van taken.

**API's en endpoints**

* **Gebruikers**:
  + POST /register: registreer een nieuwe gebruiker
  + POST /login: log een gebruiker in
  + POST /logout: log een gebruiker uit
* **Afspraken**:
  + GET /afspraken: haal alle afspraken op voor de ingelogde gebruiker
  + POST /afspraken: voeg een nieuwe afspraak toe
  + PUT /afspraken/{id}: bewerk een bestaande afspraak
  + DELETE /afspraken/{id}: verwijder een afspraak
* **Taken**:
  + GET /taken: haal alle taken op voor de ingelogde gebruiker
  + POST /taken: voeg een nieuwe taak toe
  + PUT /taken/{id}: bewerk een bestaande taak
  + DELETE /taken/{id}: verwijder een taak

**Gegevensstromen en interacties**

* **Frontend**: Interactie met gebruikers via formulieren en knoppen.
* **Backend**: Verwerking van gebruikersverzoeken en communicatie met de database.

**Beveiliging**

**Beschrijving van de beveiligingsmaatregelen**

* **Authenticatie**: Gebruikers moeten inloggen om toegang te krijgen tot de agenda en to-do lijst.
* **Autorisatie**: Alleen geautoriseerde gebruikers kunnen hun eigen afspraken en taken beheren.
* **Data encryptie**: Wachtwoorden worden gehashed en veilig opgeslagen.

**Veiligheidsprotocollen en best practices**

* Gebruik van HTTPS voor veilige communicatie.
* Regelmatige updates en patches voor alle gebruikte software.

**Prestaties en schaalbaarheid**

**Verwachte prestaties en schaalbaarheid**

* **Prestaties**: Optimalisatie van databasequery's en gebruik van caching waar mogelijk.
* **Schaalbaarheid**: Mogelijkheid om extra servers toe te voegen bij toenemende gebruikersaantallen.

**Specificatie van optimalisaties**

* **Load balancing**: Verdeling van de belasting over meerdere servers.
* **Caching**: Gebruik van caching voor veelgebruikte gegevens om de responstijd te verbeteren.

**Deployment en infrastructuur**

**Deploymentstrategie**

* **Staging**: Testomgeving voor nieuwe features.
* **Productie**: Live-omgeving voor eindgebruikers.
* **CI/CD**: Gebruik van continuous integration en continuous deployment voor het automatisch testen en uitrollen van nieuwe code.

**Overzicht van de benodigde infrastructuur**

* **Servers**: Hosting van de applicatie en database.
* **Netwerkconfiguratie**: Beveiligde verbindingen en firewalls.

**Onderhoud en support**

**Plannen voor onderhoud**

* Regelmatige back-ups van de database.
* Periodieke controles en updates van de software.

**Procedures voor bugtracking en updates**

* Gebruik van een bugtracking systeem (bijv. Jira).
* Versiebeheer met Git voor het bijhouden van wijzigingen en updates.