May 27, 2024

Reno Rovers, Joey Rehmann, Milan leijser

tia4v2b

test plan

OneGram

Contents

[Inleiding 2](#_Toc167711110)

[Soorten tests 3](#_Toc167711111)

[Unit tests 3](#_Toc167711112)

[Integratie tests 3](#_Toc167711113)

[End-to-end tests 4](#_Toc167711114)

[Handmatige tests 4](#_Toc167711115)

[Criteria livegang 5](#_Toc167711116)

[Unit tests 5](#_Toc167711117)

[Integratie tests 5](#_Toc167711118)

[End-to-end tests 5](#_Toc167711119)

[Test scenario’s 5](#_Toc167711120)

[Coverage 5](#_Toc167711121)

[Test scenario’s 6](#_Toc167711122)

[Test scenario #1: 6](#_Toc167711123)

# Inleiding

Dit testplan beschrijft de testaanpak voor het social media platform OneGram, ontwikkeld met Next.js. De applicatie dient te voldoen aan de gestelde eisen zowel voor de functionaliteiten van de applicatie als in de kwaliteit van de tests. In de rest van het document worden de specifieke testgebieden, testaanpak, testscenario’s en opleveringscriteria verder behandeld.

# Soorten tests

De applicatie zal getest worden doormiddel van een aantal verschillende tests. De soorten tests die gebruikt gaan worden zijn als volgt:

* Unit tests
* Integratie tests
* E2e tests
* Handmatige tests aan de hand van testscenario’s

## Unit tests

Unit tests zijn een belangrijk onderdeel van het testproces. Zo kunnen unit tests gebruikt worden om componenten afzonderlijk te testen. Deze testen zijn bedoeld om te verifiëren dat de afzonderlijke componenten correct functioneren.

Binnen deze applicatie zullen unit tests geschreven worden voor alle (daar waar mogelijk) componenten. Het testen van deze componenten zal worden uitgevoerd met behulp van test frameworks zoals Jest en React Testing Library. Het is de bedoeling een zo hoog mogelijke testdekking te behalen.

## Integratie tests

Integratie tests spelen ook een grote rol in het testproces van onze applicatie. Ze zorgen ervoor dat niet alleen de afzonderlijke onderdelen goed werken, maar ok dat ze samen als geheel goed functioneren.

Voor het social media platform zullen er integratie tests geschreven worden om te checken of de verschillende onderdelen goed met elkaar samenwerken. Er moet kunnen worden aangetoond dat de verschillende juist met elkaar en met Firebase communiceren.

Deze tests helpen ons te controleren of alles goed werkt zoals we verwachten. We zullen hiervoor tools zoals Jest en React Testing Library’s integratietestfunctionaliteit gebruiken.

Het doel is om zo veel mogelijk scenario’s te testen, zodat er aangetoond kan worden dat de applicatie juist werkt.

## End-to-end tests

End-to-end (e2e) tests zullen gebruikt worden om de volledige gebruikservaring van begin tot eind te verifiëren. Ze controleren niet alleen de individuele componenten en integraties, maar ook de werking van de applicatie als geheel.

Deze tests dekken de volgende scenario’s: gebruikersregistratie, inloggen, het posten van berichten en het liken en delen van content. Hiermee wordt gecontroleerd of de applicatie juist reageert op gebruikersacties en of gegevens juist worden verwerkt en getoond.

Voor deze tests zal er gebruik gemaakt gaan worden van Playwright. Dit is een framework dat browser automatisering mogelijk maakt. Met Playwright kunnen we gebruikersinteracties simuleren, zoals het klikken op knoppen, het invullen van formulieren en het navigeren door de applicatie. Het doel van deze tests is om een zo volledig mogelijke dekking te bereiken en ervoor te zorgen dat de applicatie in zijn geheel betrouwbaar en gebruiksvriendelijk is.

## Handmatige tests

Handmatige tests zijn een ander belangrijk onderdeel van het testproces. Hiermee kan de applicatie gecontroleerd worden op verschillende scenario’s die lastiger te automatiseren zijn. Er zullen verschillende test scenario’s geschreven worden om alle functionaliteiten van het dashboard te testen. Zie *Test scenario’s* voor de geschreven test scenario’s.

# Criteria livegang

Alle geschreven tests en testscenario’s moeten succesvol worden uitgevoerd voordat de website live kan gaan.

## Unit tests

Alle unit tests moeten worden uitgevoerd en slagen zonder fouten. Dit betreft het controleren van de functionaliteit en de verwachte output van individuele componenten van de applicatie.

## Integratie tests

Integratie tests moeten worden uitgevoerd om ervoor te zorgen dat verschillende onderdelen van de applicatie correct met elkaar integreren. Alle integratie tests moeten succesvol zijn.

## End-to-end tests

Alle e2e tests moeten worden uitgevoerd en slagen zonder fouten om de volledige functionaliteit van de applicatie te kunnen verifiëren.

## Test scenario’s

Handmatige tests moeten worden uitgevoerd voor verschillende scenario's en gebruiksomgevingen. Alle handmatige tests moeten succesvol worden afgerond, waarbij wordt gecontroleerd of de applicatie goed functioneert en goede gebruiksvriendelijkheid levert.

## Coverage

Een hoog niveau van testdekking moet worden bereikt, wat betekent dat alle belangrijke functionaliteiten en gebruikersscenario's grondig zijn getest.

# Test scenario’s

## Test scenario #1:

|  |  |
| --- | --- |
| **Test case ID:** | **Prioriteit:** |
| **Pre-condition:** | **Test uitgevoerd door:** |
| **Test case title:** | **Test uitgevoerd op:** |
| **Test case description:** | **Test status:** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stap | Test stap | Verwacht resultaat | Data | Resultaat | Status |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Test case ID:** | **Prioriteit:** |
| **Pre-condition:** | **Test uitgevoerd door:** |
| **Test case title:** | **Test uitgevoerd op:** |
| **Test case description:** | **Test status:** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stap | Test stap | Verwacht resultaat | Data | Resultaat | Status |
|  |  |  |  |  |  |