Examen Eindgesprek

Het Eindgesprek: Tips & Vragen

Het eindgesprek is de afronding van je examen. De assessoren gebruiken dit gesprek om vast te stellen of je over de juiste competenties beschikt.

Authenticiteit

Tijdens het eindgesprek wordt kritisch gekeken naar de **authenticiteit**. De assessoren bepalen of het opgeleverde werk daadwerkelijk door jouzelf is gemaakt en of je de gemaakte keuzes kunt onderbouwen.

Kansen bij onvolkomenheden

Mochten er tijdens het beoordelen van je portfolio kleine zaken onjuist zijn of ontbreken, dan wordt hier tijdens het gesprek naar gevraagd. Dit is jouw kans om mondeling aan te tonen dat je de stof beheerst, wat je score positief kan beïnvloeden. Let op: door de beperkte tijd kunnen niet alle acht werkprocessen uitgebreid behandeld worden; focus je dus op de kern.

Focus van het gesprek

In principe kan elke onderdeel van je examen bevraagd worden. Vanwege de beperkte tijd (meestal 20-30 minuten) maken de assessoren vaak een selectie van een aantal werkprocessen waar zij meer over willen weten.

Zorg dat je jouw portfolio direct kunt presenteren. Bereid je extra goed voor op onderdelen waarvan je zelf al vermoedt dat ze minder sterk zijn. Zorg dat je applicatie **werkend** klaarstaat en dat alle documentatie direct oproepbaar is.

Checklist voor het gesprek

- Laptop volledig opgeladen en lader binnen handbereik?
- Systeemvrije dag? (Windows/Software-updates vooraf uitgevoerd?)
- Lokale server en tools operationeel? (XAMPP, VS Code, Git, etc.)

- Portfolio lokaal beschikbaar? (Voorkom vertraging door downloaden of unzippen)
- Werking van de applicatie gecontroleerd in de examensituatie?
- **Vergeet je geldig identiteitsbewijs niet!**

Vragen die je kunt verwachten

Hoewel de vragen variëren per project, zijn dit de meest voorkomende onderwerpen die de assessoren behandelen:

Algemeen

- Wie waren de belanghebbenden (stakeholders) bij dit project?
- Hoe verliep het contact met de opdrachtgever en/of collega's?
- Ben je tevreden met de kwaliteit van het eindproduct? Wat zou een 'versie 2.0' bevatten?
- Wat was, technisch gezien, het meest uitdagende onderdeel?
- Wat is de belangrijkste les die je tijdens dit examen hebt geleerd?
- Welk advies zou je een student geven die volgend jaar dit examen doet?

Planning & Organisatie

- In hoeverre kwam de realiteit overeen met je initiële planning?
- Welke onvoorziene omstandigheden dwongen je om je planning aan te passen?
- Je gaf in de planning aan dat je "*abc*" op een bepaalde dag zou afronden; kun je laten zien hoe dat is verlopen?

Ontwerp & Voorbereiding

- Op welke punten wijkt het eindproduct af van het technisch ontwerp en waarom?
- Waarom heb je specifiek voor deze database-structuur (ERD) gekozen?
- Hoe heeft het ontwerp je geholpen tijdens de realisatiefase?

Realisatie (Code & Techniek)

- Kun je de werking van User Story X demonstreren in de applicatie?

- Kun je de code van je belangrijkste algoritme of functie toelichten?
- Stel dat ik de functionele eis "..." toevoeg, waar in de code zou je dit dan implementeren?
- Hoe heb je de veiligheid van de gegevens gewaarborgd? (SQL-injection / XSS)
- Volgens welke code-conventies of standaarden heb je geschreven?
- Kun je de tabelrelaties in je database laten zien via PHPMyAdmin?

Testen & Kwaliteit

- Kun je Test X uit je testrapport nu 'live' uitvoeren?
- Wat was de meest kritieke bug die je tijdens het testen vond en hoe heb je deze opgelost?

Communicatie & Reflectie

- Hoe heb je de gemaakte afspraken met de opdrachtgever vastgelegd?
- Hoe heb je bepaald welk technisch niveau je presentatie moest hebben voor de doelgroep?
- Welke feedback kreeg je tijdens je voortgangsgesprekken en hoe is dit verwerkt in de code?
- Als je naar je eigen proces kijkt, wat zijn dan je drie belangrijkste verbeterpunten voor een volgend project?

Gids: De Code Walkthrough

Als de assessor vraagt: "Laat de code van je inlogsysteem eens zien", is het verleidelijk om simpelweg door de file te scrollen. Doe dit niet. Volg de 'Happy Path' methode.

1. Volg de 'Happy Path'

Leg je code uit aan de hand van een gebruikersactie. Begin niet bij regel 1, maar vertel een verhaal:

- **Input:** "De gebruiker vult het formulier in op `login.php`."
- **Verwerking:** "Zodra er op verzenden wordt geklikt, vang ik de data op en controleer ik eerst met `htmlspecialchars()` of er geen schadelijke scripts in staan."

- **Database:** "Daarna gebruik ik een **Prepared Statement** om de gebruiker op te zoeken in de database. Dit voorkomt SQL-injection."
- **Resultaat:** "Als het wachtwoord klopt via `password_verify()`, maak ik een `$_SESSION` aan en stuur ik de gebruiker naar het dashboard."

2. Benoem je 'Best Practices'

Assessoren 'vinken' begrippen af in hun hoofd. Zorg dat je deze termen letterlijk noemt terwijl je de code aanwijst:

- **Don't Repeat Yourself (DRY):** "Zoals je hier ziet, gebruik ik `include 'header.php'`, zodat ik de navigatiebalk maar op één plek hoef te onderhouden."
- **Separation of Concerns:** "Ik heb mijn database-connectie in een apart bestand (`db.php`) gezet, zodat mijn logica gescheiden blijft van mijn configuratie."
- **Datatypen & Validatie:** "Bij het invoeren van een kilometerstand (of ticket-ID) controleer ik eerst of het wel echt een `int` (integer) is."

3. Wees eerlijk over fouten of 'TODO's'

Als een assessor een stukje code aanwijst dat niet perfect is, raak dan niet in paniek. Gebruik het als een kans om je reflectievermogen te tonen:

> "Klopt, op deze plek is de validatie nog beperkt. Als ik meer tijd had gehad, had ik hier een Regex-controle toegevoegd om te checken of het kenteken wel aan het Nederlandse formaat voldoet. Dat staat ook in mijn verbeterpunten beschreven."

Presentatie-hacks:

- **Zoom in:** Gebruik CTRL + in VS Code zodat de assessoren je code goed kunnen lezen op afstand of op een kleiner scherm.
- **Dark Mode:** Zorg dat je editor een prettig contrast heeft.
- **Tabbladen:** Zet de belangrijkste bestanden (bijv. `db.php`, `index.php` en je CSS) alvast open in je editor voordat je de kamer binnenstapt.

Met deze voorbereiding en de checklists hierboven straal je zelfverzekerdheid uit. Je laat zien dat je niet alleen een 'coder' bent, maar een beginnend professional die begrijpt **wat** hij bouwt en **waarom**.

Gids: Het ERD Presenteren

De assessor wil niet horen welke tabelnamen je hebt, maar waarom de relaties tussen die tabellen essentieel zijn voor de werking van de applicatie.

1. De "Single Point of Truth"

Begin met de kern: de gebruikers. Leg uit hoe de verschillende entiteiten met elkaar verbonden zijn:

- **De Entiteiten:** "Ik heb gekozen voor drie kerntabellen: `Users`, `Tickets` (of Meldingen) en `Categories`."
- **De Relatie (1-op-N):** "De belangrijkste relatie is die tussen de gebruiker en het ticket. Dit is een **één-op-veel relatie**: één gebruiker kan meerdere meldingen maken, maar een melding hoort altijd bij één specifieke gebruiker."
- **Foreign Keys:** "Om dit technisch te realiseren, sla ik het `user_id` op als **Foreign Key** in de ticket-tabel. Hiermee waarborg ik de data-integriteit."

2. Normalisatie en Efficiëntie

Laat zien dat je hebt nagedacht over een schone database:

> "Ik heb de categorieën (Hardware/Software) in een aparte tabel gezet in plaats van als tekst in de ticket-tabel. Hierdoor is mijn database **genormaliseerd**. Als we later een categorienaam willen aanpassen, hoeft dat maar op één plek in de database en verandert het automatisch voor alle gekoppelde tickets."

3. Voorbereiding op "Stel dat..." vragen

Assessoren testen je ontwerp vaak door een wijziging voor te stellen:

- **Vraag:** "Stel dat een ticket door meerdere monteurs tegelijk opgepakt moet worden?"
- **Jouw antwoord:** "In mijn huidige ontwerp is dat een 1-op-N relatie (één monteur per ticket). Om dat mogelijk te maken, zou ik een **koppeltabel** moeten toevoegen om er een **N-op-M relatie** (veel-op-veel) van te maken."

Presentatie-hacks voor je ERD:

- **Pointer:** Gebruik je muis (of een laserpointer) om de lijnen tussen de tabellen aan te wijzen terwijl je de relatie benoemt.
- **PK/FK:** Wijs expliciet de **Primary Key** (het gouden sleuteltje) en de **Foreign Key** aan.
- **Legenda:** Zorg dat je ERD duidelijk de datatypen laat zien (bijv. `INT`, `VARCHAR(255)`, `DATETIME`).

Laatste tip: Oefen deze uitleg één keer hardop. Als je de termen *Primary Key*, *Foreign Key* en *Eén-op-veel relatie* vloeiend gebruikt, heb je de helft van de punten voor je database-ontwerp al binnen.

De 5 Meest Gemaakte Fouten

(en hoe ze te voorkomen)

Veel studenten gaan de mist in op zaken die niets met hun code te maken hebben, maar alles met hun houding en voorbereiding.

1. "Ik weet het niet" (Zonder vervolg)

Het is oké om een technisch detail even niet te weten, maar "Ik weet het niet" is een dooddoener.

- **De Oplossing:** Zeg: "Ik weet de exacte PHP-functie hiervoor niet uit mijn hoofd, maar ik kan je wel de logica uitleggen van hoe ik het heb opgelost..." of "Dat onderdeel heb ik opgezocht in de documentatie van PHP.net."

2. "Het werkte gisteren nog..."

De grootste nachtmerrie van elke student: een demo die crasht. Vaak komt dit door onverwachte updates of een vergeten database-import.

- **De Oplossing:** Test je applicatie **15 minuten voor het gesprek** op de locatie zelf. Maak een back-up van je database (SQL-export) en zet je code eventueel ook op een USB-stick als extra veiligheid.

3. Te passief afwachten

Studenten die alleen "Ja" of "Nee" antwoorden, dwingen de assessor om te blijven 'trekken'. Dit wekt de indruk dat je niet trots bent op je werk.

- **De Oplossing:** Neem het initiatief. Als een assessor vraagt naar je inlogscherm, vertel dan ook direct *waarom* je voor die specifieke beveiliging hebt gekozen. Laat zien dat je de expert bent van jouw eigen project.

4. Code laten zien die je niet begrijpt

Als je een stuk code van StackOverflow of AI hebt gebruikt zonder het te begrijpen, val je door de mand zodra de assessor vraagt: "Wat gebeurt er op regel 42?".

- **De Oplossing:** Gebruik alleen code die je kunt uitleggen. Heb je hulp gehad? Zorg dan dat je elk onderdeel van die code hebt uitgeplozen en kunt verantwoorden.

5. Geen bewijs van testen

Beweren dat je applicatie "bugvrij" is zonder dat je kunt laten zien *hoe* je dat hebt getest, is ongeloofwaardig.

- **De Oplossing:** Houd je testrapport bij de hand. Wijs een fout aan die je tijdens het bouwen vond en laat zien hoe je die hebt opgelost. Assessoren houden van studenten die kritisch zijn op hun eigen werk.

De "Gouden Regel" voor je gesprek:

"Don't just show, tell why." (Laat niet alleen zien wat je hebt gemaakt, maar vertel vooral waarom je het zo hebt gedaan.)

Je hebt nu de projectbriefing, de technische handleidingen, de code-walkthrough gids en de ERD-presentatie. Je bent er helemaal klaar voor. Veel succes met je examen!

Revision #8
Created 2023-12-23 14:05:27 UTC by Max
Updated 2025-12-18 09:20:07 UTC by Max