1. **Verbetervoorstellen**   
     
   Naam: Justin Mank  
   Klas: 3A  
   Datum: 13-5-2024  
   Versie: 1.0  
     
     
    Algemeen  
   1. Beter plannen, Ik merk op dat ik mentaal niet erg ready was voor het examen, daarnaast zat ik ook met andere opdrachten van het bedrijf waardoor ik me niet een volledige focus had kunnen geven op het examen en daardoor wat langzamer was op bepaalde onderdelen  
     
   2. Tempo wat verhogen bij taken, Wanneer ik een taak zat uit te voeren waarbij ik niet echt uitkwam zat ik soms echt minuten lang na te denken hoe je het zou kunnen fixen. Inplaats daarvan in de toekomst is het misschien beter om momenten waarbij ik niet iets begrijp beter kan opschrijven en dit later kan oppakken.  
      
    (acceptatie) Test  
     
   1. **Scheid de verantwoordelijkheden**: Ik merk dat mijn Login-klasse momenteel belast is met zowel het afhandelen van de inloglogica als het beheren van databaseconnecties. Hierdoor wordt mijn klasse minder modulair en complexer om te testen. Misschien moet ik overwegen om de functionaliteit voor het beheren van databaseconnecties naar een aparte klasse te verplaatsen, zoals een DatabaseManager-klasse, zodat mijn Login-klasse zich uitsluitend kan concentreren op inloglogica
2. **Voeg foutafhandeling toe**: Het lijkt verstandig om de foutafhandeling in mijn code te verbeteren, vooral bij databasebewerkingen. Door te controleren of databasequery's succesvol worden uitgevoerd en het toevoegen van foutmeldingen wanneer er problemen optreden, kunnen zowel gebruikers als ikzelf beter begrijpen wat er fout is gegaan.
3. **Gebruik PDO-uitzonderingen**: Ik zou kunnen overwegen om try-catch blokken te gebruiken om PDO-uitzonderingen op te vangen bij databasebewerkingen. Op deze manier kan ik databasefouten afhandelen en erop reageren, in plaats van dat mijn applicatie crasht.
4. **Beveiliging inlogproces**: Betere beveiliging, ik maak gebruik van password\_verify om het wachtwoord te laten controleren en ik maak ook gebruik van hashed\_passwords voor in de database, maar voor de volgende keer overwegen om meer beveiligingsmaatregelen toe te voegen
5. **Gebruik maken van een framework**: Het kan de moeite waard zijn om een PHP-framework zoals Laravel (of andere frameworks) te overwegen. Een framework zoals dit biedt een breed variatie aan functionaliteiten voor het ontwikkelen van veilige en schaalbare webapplicaties, inclusief ingebouwde authenticatie- en autorisatiemechanismen..
6. **Testgedreven ontwikkeling**: Misschien moet ik eens kijken naar testgedreven ontwikkeling (TDD) als ontwikkelingsaanpak. Door eerst tests te schrijven en vervolgens de functionaliteit te implementeren om aan die tests te voldoen, kan ik code produceren die beter testbaar, modulair en onderhoudsvriendelijk is.