



## Realisaties Mass project

ITFactory

Noah Apers r0868806

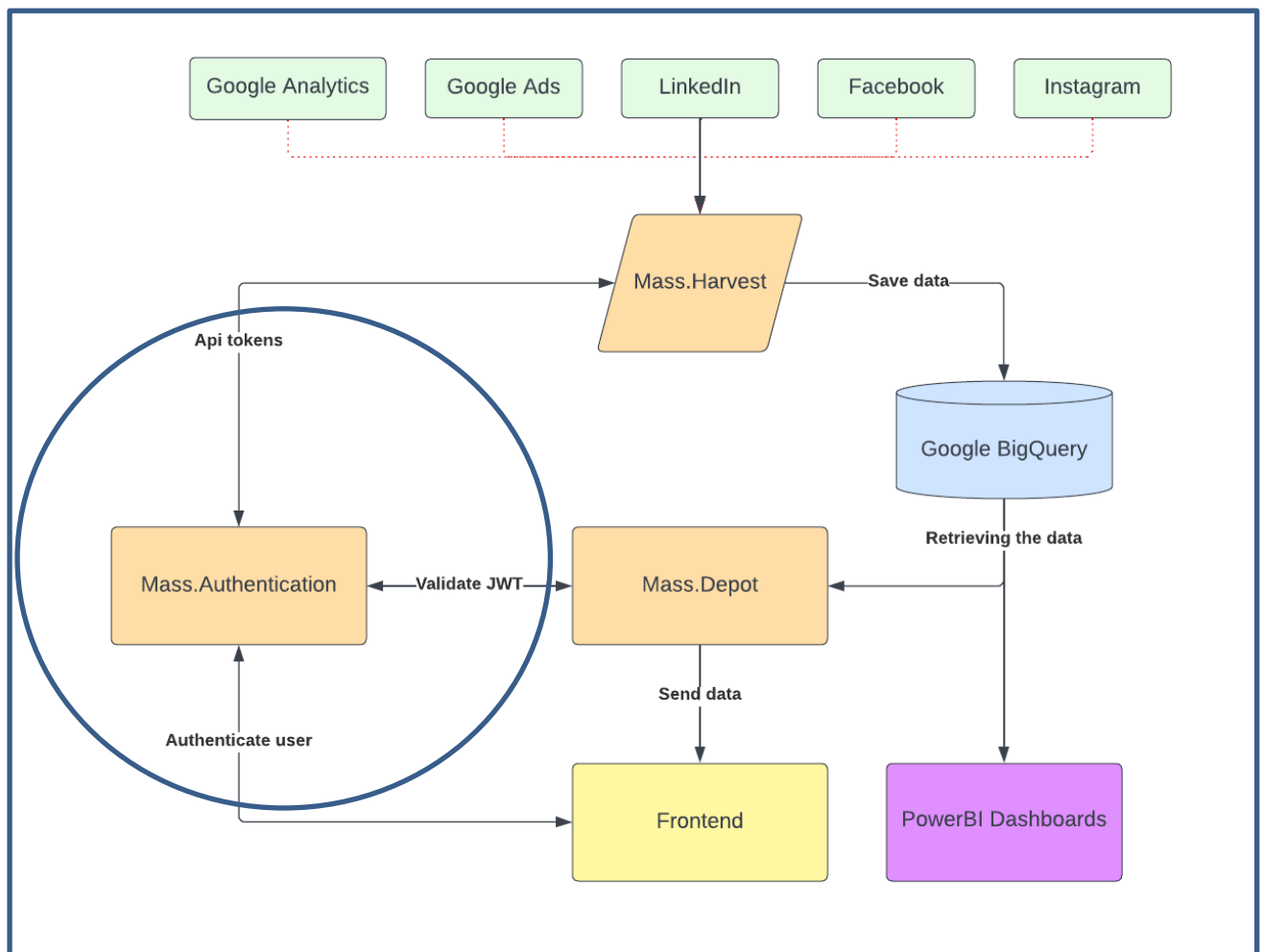
Academic Year 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel


<b>2</b>	<b>REALIZATIES .....</b>	<b>4</b>
<b>2.1</b>	<b>Samenvatting gemaakte endpoints ( WebAPI ) .....</b>	<b>5</b>
<b>2.2</b>	<b>Api credentials .....</b>	<b>6</b>
2.2.1	/api/v{v}/ApiCredentials.....	6
2.2.2	/api/v{v}/ApiCredentials/{id} .....	6
<b>2.3</b>	<b>Authenticate .....</b>	<b>6</b>
2.3.1	/api/v{v}/Authenticate/Login .....	6
2.3.2	/api/v{v}/Authenticate/Validate .....	6
<b>2.4</b>	<b>LinkedIn.....</b>	<b>7</b>
2.4.1	/api/v{v}/ApiCredentials/LinkedIn/oauth/url .....	7
2.4.2	/api/v{v}/ApiCredentials/LinkedIn/oauth/authenticate .....	7
2.4.3	/api/v{v}/ApiCredentials/LinkedIn/setup/organizations .....	7
2.4.4	/api/v{v}/ApiCredentials/LinkedIn/setup/organisations .....	7
2.4.5	/api/v{v}/ApiCredentials/LinkedIn/setup/refreshToken .....	7
<b>2.5</b>	<b>Organisations.....</b>	<b>8</b>
2.5.1	/api/v{v}/Organisations .....	8
2.5.2	/api/v{v}/Organisations/{id} .....	8
2.5.3	/api/v{v}/Organisations/{id} .....	8
2.5.4	/api/v{v}/Organisations/{id} .....	8
2.5.5	/api/v{v}/Organisations/{organisationId}/users .....	8
2.5.6	/api/v{v}/Organisations/{organisationId}/connectedServices .....	8
2.5.7	/api/v{v}/Organisations/apiCredentials .....	8
2.5.8	/api/v{v}/Organisations/OrganisationWithUser .....	8
2.5.9	/api/v{v}/Organisations/{id}/inviteCode .....	9
2.5.10	/api/v{v}/Organisations/users/invite .....	9
2.5.11	/api/v{v}/Organisations/{organisationsId}/users/{userId} .....	9
2.5.12	/api/v{v}/Organisations/{organisationsId}/users/{userId}/role .....	9
2.5.13	/api/v{v}/Organisations/OrganisationInviteCodeCheck .....	9
<b>2.6</b>	<b>Users.....</b>	<b>10</b>
2.6.1	/api/v{v}/Users .....	10
2.6.2	/api/v{v}/Users .....	10
2.6.3	/api/v{v}/Users/{id}.....	10
2.6.4	/api/v{v}/Users/{id}.....	10
2.6.5	/api/v{v}/Users/{id}.....	10
2.6.6	/api/v{v}/Users/me .....	11
2.6.7	/api/v{v}/Users/addOrganisation .....	11
<b>3</b>	<b>TESTING.....</b>	<b>12</b>
<b>4</b>	<b>OUTPUT CACHING .....</b>	<b>13</b>

# 1 REALIZATIES

Het doel van dit document is om een gedetailleerd overzicht te geven van mijn bijdragen aan het massaproject. Mijn belangrijkste verantwoordelijkheid binnen het project is het ontwerpen en ontwikkelen van de backend-infrastructuur die verantwoordelijk is voor talrijke cruciale processen. Deze processen omvatten gebruikers- en organisatie-authenticatie, gebruikersbeheer, operationele taken van de organisatie en het beheer van API-gegevensreferenties die nodig zijn voor gegevensopvraging.




## 1.1 Samenvatting gemaakte endpoints ( WebAPI )

**Mass Authentication** 

<https://localhost:7016/swagger/v1/swagger.json>

Api for Mass authentication

Elmos - Website  
Send email to Elmos

Authorize 

### ApiCredentials

- GET /api/v{v}/ApiCredentials
- GET /api/v{v}/ApiCredentials/{id}

### Authenticate

- POST /api/v{v}/Authenticate/Login
- POST /api/v{v}/Authenticate/Validate

### LinkedIn

- GET /api/v{v}/ApiCredentials/LinkedIn/oauth/url
- POST /api/v{v}/ApiCredentials/LinkedIn/oauth/authenticate
- GET /api/v{v}/ApiCredentials/LinkedIn/setup/organizations
- POST /api/v{v}/ApiCredentials/LinkedIn/setup/organizations
- GET /api/v{v}/ApiCredentials/LinkedIn/setup/refreshToken

### Organisations

- GET /api/v{v}/Organisations
- GET /api/v{v}/Organisations/{id}
- PUT /api/v{v}/Organisations/{id}
- DELETE /api/v{v}/Organisations/{id}
- GET /api/v{v}/Organisations/{organisationId}/users
- GET /api/v{v}/Organisations/{organisationId}/connectedServices
- GET /api/v{v}/Organisations/apiCredentials
- POST /api/v{v}/Organisations/OrganisationWithUser
- PUT /api/v{v}/Organisations/{id}/InviteCode
- POST /api/v{v}/Organisations/users/invite
- DELETE /api/v{v}/Organisations/{organisationId}/users/{userId}
- PUT /api/v{v}/Organisations/{organisationId}/users/{userId}/role
- POST /api/v{v}/Organisations/OrganisationInviteCodeCheck

### Users

- GET /api/v{v}/Users
- POST /api/v{v}/Users
- GET /api/v{v}/Users/{id}
- PUT /api/v{v}/Users/{id}
- DELETE /api/v{v}/Users/{id}
- GET /api/v{v}/Users/me
- POST /api/v{v}/Users/addOrganisation

## 1.2 Api credentials

API datareferenties spelen een essentiële rol in het verkrijgen en ophalen van data van specifieke social media API's. Deze referenties, meestal bestaande uit een API sleutel, token of een combinatie van beide, verlenen geautoriseerde gebruikers de nodige machtigingen om met de API van het social media platform te communiceren.

### 1.2.1 /api/v{v}/ApiCredentials

Deze route heeft de functie om alle api-credentials op te halen.

### 1.2.2 /api/v{v}/ApiCredentials/{id}

Deze route heeft de functie om een specifieke api-credentials op te halen.

## 1.3 Authenticate

### 1.3.1 /api/v{v}/Authenticate/Login

Om toegang te krijgen tot ons systeem, moet u een specifieke route volgen die leidt tot het inlogproces. We hebben een geavanceerd authenticatiesysteem geïmplementeerd op basis van JSON Web Tokens (JWT's) om een veilige en efficiënte inlogervaring te garanderen.

Nadat een gebruiker succesvol is ingelogd, ontvangt hij of zij een uniek JWT-token. Dit token fungeert als een digitale sleutel die voor een vooraf bepaalde periode toegang geeft tot het systeem. Het bevat een tijdstempel die de vervaldatum aangeeft, zodat gebruikers automatisch worden uitgelogd wanneer de toegewezen tijd is verstreken. Deze proactieve benadering beschermt gebruikersaccounts door ongeautoriseerde toegang buiten de vastgestelde tijdspanne te voorkomen.

### 1.3.2 /api/v{v}/Authenticate/Validate

Deze route biedt de functionaliteit om een verzoek tot het ophalen van gegevens te valideren. Het voert verschillende controles uit om ervoor te zorgen dat het verzoek geautoriseerd is en dat de gebruiker de nodige toestemmingen heeft.

De eerste stap is om het JWT-token dat in het verzoek is opgenomen te valideren. Het token bevat de gebruikers-ID, wat cruciaal is voor het daaropvolgende validatieproces. De route verifieert de authenticiteit en integriteit van het token om te garanderen dat er niet mee geknoeid is.

Eenmaal het token is gevalideerd, gaat de route verder met het controleren of de gebruiker is geassocieerd met de organisatie van waaruit zij gegevens willen ophalen. Deze stap zorgt ervoor dat gebruikers alleen toegang hebben tot gegevens van organisaties waarmee zij verbonden zijn. De route vergelijkt de gebruikers-ID die uit het JWT-token is gehaald met de lijst van gebruikers die

geassocieerd zijn met de organisatie. Als er een overeenkomst is, gaat het validatieproces verder.

After confirming the user's affiliation, the route verifies whether the organization has the proper API credentials to retrieve the requested data or to be authorized for accessing it. This could involve validating an API key or checking for specific authorization roles. The route compares the organization's API credentials with the required credentials. If the organization is authorized, the process return 200 and the other backend knows the validation succeeded.

## **1.4      LinkedIn**

Om een koppeling te maken tussen een organisatie en LinkedIn, is een reeks onderling verbonden routes nodig. Deze verbinding wordt vergemakkelijkt door het gebruik van OAuth, een open standaardprotocol voor veilige autorisatie.

**1.4.1      /api/v{v}/ApiCredentials/LinkedIn/oauth/url**

**1.4.2      /api/v{v}/ApiCredentials/LinkedIn/oauth/authenticate**

**1.4.3      /api/v{v}/ApiCredentials/LinkedIn/setup/organizations**

**1.4.4      /api/v{v}/ApiCredentials/LinkedIn/setup/organisations**

**1.4.5      /api/v{v}/ApiCredentials/LinkedIn/setup/refreshToken**

Deze route dient om de acceskey van een organisatie tot de LinkedIn API te vernieuwen. Het wordt gebruikt in andere backends wanneer de geldigheidsperiode van de key is verstreken.

## 1.5 Organisations

### 1.5.1 /api/v{v}/Organisations

Het doel van deze route is om een lijst van alle organisaties opgeslagen in ons systeem op te halen. Om de flexibiliteit te vergroten, heb ik een paginering functie geïntegreerd, waarmee je het gewenste aantal organisaties kunt specificeren dat per query wordt weergegeven.

### 1.5.2 /api/v{v}/Organisations/{id}

Deze route geeft de gespecificeerde organisatie terug.

### 1.5.3 /api/v{v}/Organisations/{id}

Deze PUT-route maakt het mogelijk om elk veld van een organisatie te wijzigen.

### 1.5.4 /api/v{v}/Organisations/{id}

De delete-route faciliteert het verwijderen van een specifieke organisatie uit ons systeem. Het maakt gebruik van een soft delete benadering, waardoor er geen gegevens permanent verloren gaan. Als gevolg hiervan wordt de status van de organisatie gemarkeerd als "verwijderd", terwijl alle relevante gegevens behouden blijven.

### 1.5.5 /api/v{v}/Organisations/{organisationId}/users

Deze route maakt het mogelijk om een uitgebreide lijst van alle gebruikers geassocieerd met de organisatie op te halen. Door gebruik te maken van deze route, kunnen we moeiteloos toegang krijgen tot en informatie verzamelen over elke gebruiker binnen de organisatie.

### 1.5.6 /api/v{v}/Organisations/{organisationId}/connectedServices

Deze route vergemakkelijkt het ophalen van alle verbonden diensten die geassocieerd zijn met een organisatie. Deze verbonden diensten kunnen vergeleken worden met platformen zoals LinkedIn.

### 1.5.7 /api/v{v}/Organisations/apiCredentials

Deze route dient om alle API-gegevensreferenties op te halen die geassocieerd zijn met organisaties en hun respectievelijke gebruikers. Het is ontworpen als een beveiligde route, toegankelijk alleen voor onze backend systemen. Om veilige toegang te waarborgen, moet een handtekening in de verzoekheader worden opgenomen. Zonder de juiste beveiligingssleutel zal de handtekening niet geldig zijn, waardoor ongeautoriseerd gebruik van deze route effectief wordt voorkomen.

### 1.5.8 /api/v{v}/Organisations/OrganisationWithUser

Door deze POST-route te gebruiken, kunt u een nieuwe organisatie creëren samen met een beheerder die specifiek aan die organisatie is toegewezen. Dit gestroomlijnde proces zorgt voor een efficiënte opzet van organisatiestructuren, wat een naadloos beheer en administratie mogelijk maakt.

### **1.5.9     /api/v{v}/Organisations/{id}/inviteCode**

Deze route maakt het vernieuwen van een uitnodigingscode voor een organisatie mogelijk. Door deze route te gebruiken, kunnen organisaties een nieuwe uitnodigingscode genereren om te delen met nieuwe leden of gebruikers. Dit proces zorgt ervoor dat organisaties controle kunnen houden over wie toegang heeft tot hun platform, terwijl ze een bijgewerkte en geldige code verstrekken voor uitnodigingsdoeleinden.

### **1.5.10    /api/v{v}/Organisations/users/invite**

Deze route vergemakkelijkt de uitnodiging van gespecificeerde e-mailadressen. Bij het aanroepen van deze route wordt een e-mail verzonden naar de verstrekte lijst met e-mails, met een link naar de frontend-toepassing. Wanneer de ontvangers op de link klikken, worden ze geleid naar een vooraf ingevuld formulier waar hun e-mailadres en uitnodigingscode al zijn ingevuld. Van daaruit kunnen ze doorgaan met het aanmaken van een nieuwe gebruikersaccount, wat het registratieproces stroomlijnt en zorgt voor een naadloze onboarding-ervaring.

### **1.5.11    /api/v{v}/Organisations/{organisationsId}/users/{userId}**

De delete-route maakt het mogelijk om een specifieke gebruiker te verwijderen uit een bepaalde organisatie. Het maakt gebruik van een soft delete benadering, waardoor er geen permanente gegevensverlies optreedt. Door deze route te gebruiken, wordt de associatie van de gebruiker met de organisatie gemarkeerd als verwijderd, terwijl de relevante informatie behouden blijft.

### **1.5.12    /api/v{v}/Organisations/{organisationsId}/users/{userId}/role**

Deze route is specifiek ontworpen voor het wijzigen van de rol van een gebruiker binnen een organisatie. Door deze route te gebruiken, kunnen beheerders of geautoriseerd personeel gemakkelijk de aan een gebruiker toegewezen rol bijwerken, hen verschillende rechten en toegangsniveaus verlenen op basis van hun verantwoordelijkheden en vereisten binnen de organisatie. Deze functionaliteit biedt flexibiliteit en naadloos rolbeheer binnen de organisatiestructuur.

### **1.5.13    /api/v{v}/Organisations/OrganisationInviteCodeCheck**

Deze route maakt het mogelijk om informatie op te halen over de organisatie waartoe een gespecificeerde uitnodigingscode behoort. Door gebruik te maken van deze route, kunnen gebruikers een uitnodigingscode als invoer geven en details verkrijgen over de geassocieerde organisatie. Deze functionaliteit maakt een efficiënte opzoeking en verificatie van uitnodigingscodes mogelijk, waardoor gebruikers kunnen bepalen met welke organisatie ze zijn verbonden.



## 1.6 Users

De gebruikerstabel bevat alle essentiële informatie over een gebruiker, waaronder hun geassocieerde bedrijf, wachtwoorden, gebruikersnaam, e-mail en alle andere relevante details die nodig zijn voor het effectief creëren en beheren van gebruikers binnen uw systeem.

### 1.6.1 `/api/v{v}/Users`

De GET-route haalt gebruikers op uit onze database, waardoor u kunt kiezen hoeveel gegevens u wilt ophalen met behulp van paginering.

### 1.6.2 `/api/v{v}/Users`

De POST-route stelt gebruikers in staat om nieuwe accounts in ons systeem aan te maken met behulp van een uitnodigingscode van een organisatie. Deze gebruiker wordt vervolgens toegevoegd aan de organisatie.

### 1.6.3 `/api/v{v}/Users/{id}`

Deze get-route haalt een specifieke gebruiker op.

### 1.6.4 `/api/v{v}/Users/{id}`

De PUT-route stelt gebruikers in staat om al hun profielvelden individueel te wijzigen op onze frontend profielpagina. Door een enkel verzoek met de bijgewerkte gegevens te versturen, zorgen we voor naadloze en nauwkeurige updates van de backend-record van de gebruiker. Deze aanpak vereenvoudigt het proces en verbetert de efficiëntie.

### 1.6.5 `/api/v{v}/Users/{id}`

De DELETE-route is verantwoordelijk voor het verwijderen van de gespecificeerde gebruiker uit ons systeem. Door een soft delete strategie te implementeren, treedt er geen gegevensverlies op tijdens dit proces. In plaats van de informatie van de gebruiker permanent te wissen, wordt de status van de gebruiker gewoon bijgewerkt naar "verwijderd". Deze aanpak zorgt ervoor dat relevante gegevens behouden blijven terwijl wordt aangegeven dat de gebruiker niet langer actief is in het systeem.

**1.6.6     /api/v{v}/Users/me**

Deze route valideert het JWT-token van de gebruiker om de identiteit van de ingelogde gebruiker te bepalen. Daarnaast verstrekt het informatie over de organisatie waartoe de gebruiker behoort.

**1.6.7     /api/v{v}/Users/addOrganisation**

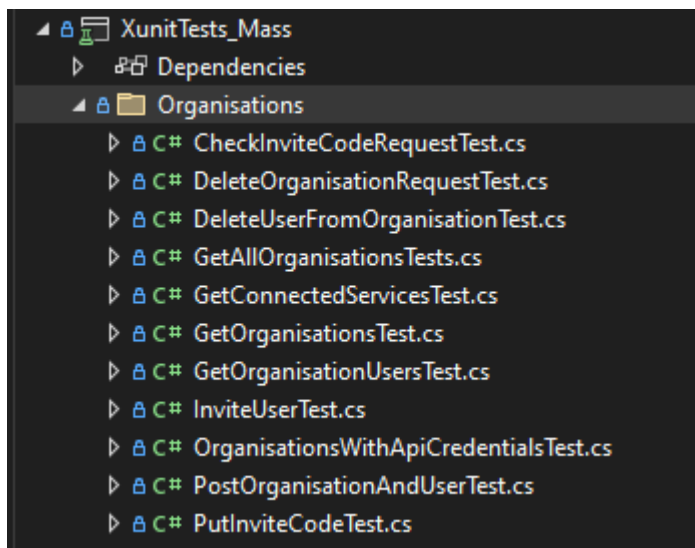
Deze route maakt het mogelijk om een bestaande gebruiker toe te voegen aan een gespecificeerde organisatie met behulp van een uitnodigingscode.

## 2 TESTING

Een goede testdekking is van cruciaal belang bij het ontwikkelen van een robuuste web API. Het testen van de belangrijkste request handlers is een essentieel onderdeel van het waarborgen van de betrouwbaarheid en functionaliteit van de API. Moq en xUnit zijn veelgebruikte hulpmiddelen waarmee we deze tests kunnen schrijven en uitvoeren.

Om de belangrijke request handlers van onze web API te testen, maken we gebruik van het Moq-framework. Moq stelt ons in staat om nepobjecten te creëren die afhankelijkheden vervangen en ons in staat stellen om specifieke scenario's te simuleren. We combineren Moq met het xUnit-testframework om onze tests te schrijven en uit te voeren.

Het doel van het testen van onze request handlers is om ervoor te zorgen dat ze correct reageren op inkomende verzoeken en de verwachte resultaten retourneren. Door het gebruik van mocks kunnen we externe afhankelijkheden isoleren en onafhankelijk van externe services of databases testen. Dit zorgt voor een gecontroleerde omgeving waarin we onze request handlers kunnen valideren en fouten kunnen identificeren.



### 3 OUTPUT CACHING

Ik heb output caching toegevoegd aan ons project omdat het van essentieel belang is voor de verbetering van de prestaties en de schaalbaarheid van mijn webapplicatie. Met output caching kan ik de responstijd van mijn applicatie aanzienlijk verkorten.

Door het toepassen van output caching sla ik het resultaat van bewerkingen op in een cache. Dit betekent dat wanneer een vergelijkbaar verzoek wordt ontvangen, het antwoord direct uit de cache kan worden gehaald, zonder dat de bewerking opnieuw hoeft te worden uitgevoerd. Hierdoor worden de verwerkings- en reistijd van gegevens drastisch verminderd, waardoor mijn applicatie sneller en responsiever wordt.

```
[HttpGet("{id}")]
[OutputCache(PolicyName = "1_Min_Policy")]
/*    [Authorize]*/
[ProducesResponseType(StatusCodes.Status400BadRequest)]
[ProducesResponseType(StatusCodes.Status401Unauthorized)]
[ProducesResponseType(StatusCodes.Status404NotFound)]
[ProducesResponseType(typeof(Response<OrganisationDto>), 200)]
0 references
public async Task<IActionResult> GetOne(Guid id)
{
    return await ExecuteRequest(new GetUserRequest(id));
}
```

## **4        SOFT SKILLS**

In dit project was het van cruciaal belang om mijn softskills optimaal te benutten. We werkten aan een uitdagende taak waarbij dagelijkse stand-up meetings een essentieel onderdeel waren van ons werkproces. Deze korte bijeenkomsten boden de gelegenheid om onze voortgang te bespreken, eventuele obstakels te identificeren en onze doelen voor de dag te bepalen. Door deel te nemen aan deze meetings kon ik mijn communicatieve vaardigheden verbeteren door beknopt en duidelijk te rapporteren over mijn werkzaamheden en eventuele knelpunten te bespreken met mijn teamleden.

Een ander hoogtepunt van dit project was de vereiste om regelmatig demo-presentaties aan de klant te geven. Deze presentaties gaven ons de mogelijkheid om onze voortgang en resultaten aan de klant te tonen. Tijdens deze presentaties kon ik mijn presentatievaardigheden aanscherpen door complexe technische concepten begrijpelijk en boeiend over te brengen aan de klant. Ik maakte gebruik van visuele hulpmiddelen, zoals grafieken en diagrammen, om de informatie helder te presenteren en de klant te betrekken bij het project.

Binnen dit project waren samenwerkingsvaardigheden van onschatbare waarde. Samenwerken met mijn teamleden was van essentieel belang om onze gemeenschappelijke doelen te bereiken. Ik heb geleerd om effectief te communiceren en te luisteren naar de ideeën en standpunten van anderen. We moesten als team beslissingen nemen en ik was actief betrokken bij het proces, waarbij ik constructieve feedback gaf en bijdroeg aan een positieve teamdynamiek.