

Stripe Payment Integration - Implementatie Samenvatting

✓ Wat is er geïmplementeerd?

1. Database Wijzigingen

- **Nieuwe velden in `users` tabel:**
- `stripe_customer_id` - Stripe klant ID
- `stripe_subscription_id` - Stripe abonnement ID
- `subscription_status` - Status: inactive/active/past_due/cancelled
- `subscription_start_date` - Start datum abonnement
- `subscription_end_date` - Eind datum abonnement
- **Migratie script:** `migrate_stripe.py` (voegt velden toe aan bestaande database)

2. Nieuwe Pagina's

- `/payment` - Betaalpagina met moderne UI
- Toont abonnement details (€99/maand)
- Lijst met features
- Stripe Checkout integratie
- Ondersteuning voor iDEAL en creditcard
- `/payment/success` - Success pagina
- Bevestiging van succesvolle betaling
- Automatische redirect naar dashboard (5 sec)
- `/payment/cancel` - Cancel pagina
- Melding dat betaling is geannuleerd
- Opties om opnieuw te proberen of terug te gaan

3. API Endpoints

Betalings Endpoints

- **POST** `/api/create-checkout-session`
- Maakt Stripe Checkout sessie aan
- Gebruikt bestaande of nieuwe Stripe customer
- Haalt of maakt price aan voor product
- Ondersteunt iDEAL en creditcard
- **POST** `/api/stripe/webhook`
- Ontvangt Stripe webhook events
- Verwerkt subscription updates
- Handelt betalingen af

Aangepaste Endpoints

- **POST** `/api/auth/login`
- Check subscription status
- Blokkeert login als status != 'active'
- Redirect naar `/payment` als betaling vereist
- Superusers bypassen payment check
- **POST** `/api/auth/register`
- Maakt user aan met status 'inactive'
- Zet `pending_user_id` in session
- Frontend redirect naar `/payment`

4. Webhook Handlers

Automatische verwerking van Stripe events:

- **`checkout.session.completed`**
- Activeert subscription na succesvolle betaling
- Update `subscription_status` naar 'active'
- Slaat `customer_id` en `subscription_id` op
- **`customer.subscription.updated`**
- Update subscription status bij wijzigingen
- Handelt status changes af (active/past_due/etc)
- **`customer.subscription.deleted`**
- Zet status op 'cancelled'
- Registreert `end_date`
- **`invoice.payment_succeeded`**
- Bevestigt succesvolle maandelijkse betaling
- Houdt subscription actief
- **`invoice.payment_failed`**
- Zet status op 'past_due'
- Gebruiker krijgt grace period

5. Frontend Wijzigingen

Login Pagina (`templates/login.html`)

- Detecteert `payment_required` error
- Toont melding over betaling
- Automatische redirect naar `/payment` (2 sec)

Registratie Flow

- Na registratie: direct naar `/payment`
- Melding: "Account aangemaakt! Doorverwijzen naar betaling..."

6. Configuratie

Environment Variables (Vereist)

```
STRIPE_SECRET_KEY=sk_test_... of sk_live_...
STRIPE_PUBLISHABLE_KEY=pk_test_... of pk_live_...
STRIPE_WEBHOOK_SECRET=whsec_...
```

Product Configuratie

- Product ID: `prod_TBxZgD1ASfSHPc`
- Prijs: €99.00 per maand
- Recurring: Maandelijks

7. Security Features

- ☒ Webhook signature verificatie
- ☒ Session-based user tracking
- ☒ Secure payment flow via Stripe
- ☒ PCI compliance (Stripe handled)
- ☒ Superuser bypass voor admins

8. User Experience Flow

Nieuwe Gebruiker:

1. Registreer → 2. Redirect /payment → 3. Betaal → 4. Webhook → 5. Dashboard

Bestaande Gebruiker (geen subscription):

1. Login → 2. Check subscription → 3. Redirect /payment → 4. Betaal → 5. Dashboard

Bestaande Gebruiker (actief subscription):

1. Login → 2. Check subscription → 3. Dashboard ☒

Superuser:

1. Login → 2. Bypass payment check → 3. Dashboard ☒

Deployment Checklist

- ☐ Code gepushed naar GitHub
- ☐ Stripe account aangemaakt
- ☐ Product aangemaakt in Stripe (of bestaande gebruikt)
- ☐ API keys toegevoegd aan Render environment
- ☐ Webhook endpoint geconfigureerd in Stripe
- ☐ Webhook secret toegevoegd aan Render
- ☐ Test betaling uitgevoerd
- ☐ Database automatisch gemigreerd
- ☐ Superuser account aangemaakt (optioneel)

Testing

Test Mode

```
Card: 4242 4242 4242 4242
Expiry: 12/25 (elke toekomstige datum)
CVC: 123
Postcode: 1234AB
```

Verificatie

1. Maak test account aan
2. Wordt doorgestuurd naar /payment
3. Vul test card in
4. Betaling succesvol
5. Redirect naar dashboard
6. Check database: `subscription_status = 'active'`

Documentatie

- **STRIPE_SETUP.md** - Gedetailleerde Stripe configuratie
- **DEPLOYMENT_INSTRUCTIONS.md** - Stap-voor-stap deployment guide
- **migrate_stripe.py** - Database migratie script

Volgende Stappen

1. **Push code naar GitHub** (handmatig via je lokale machine)
2. **Configureer Stripe keys** op Render
3. **Test de betaling** in test mode
4. **Ga live** met live keys

Tips

- Start altijd met test mode
- Test alle webhook events
- Controleer logs in Stripe Dashboard
- Monitor subscription statussen in database
- Maak een superuser voor jezelf

Support

Bij problemen:

1. Check Render logs
 2. Bekijk Stripe webhook logs
 3. Controleer database `subscription_status`
 4. Lees `DEPLOYMENT_INSTRUCTIONS.md`
-

Implementatie voltooid! 🚀

Alle code is klaar en gecommitt. Push naar GitHub en configureer Stripe keys om te starten.