

```
const express = require('express');
const router = express.Router();
const stripe = require('stripe')(process.env.STRIPE_SECRET_KEY);
const fs = require('fs');
const path = require('path');
const sendEmail = require('../mailer');
const generateLeads = require('../generateLeads');
const config = require('../config');
```

```
// Create a PaymentIntent
```

```
router.post('/create-payment-intent', async (req, res) => {
  try {
    const { niche, amount } = req.body;
    if (!niche || !amount) return res.status(400).json({ message:
'Missing' });
```

```
    const paymentIntent = await stripe.paymentIntents.create({
      amount,
      currency: 'usd',
      metadata: { niche }
    });
```

```
    res.json({ clientSecret: paymentIntent.client_secret,
publishableKey: process.env.STRIPE_PUBLISHABLE_KEY });
  } catch (err) {
    console.error(err);
    res.status(500).json({ message: err.message });
```

```
}  
});
```

```
// Finalize order after client-side confirmed payment
```

```
router.post('/finalize-order', async (req, res) => {  
  try {  
    const { paymentIntentId, email, niche } = req.body;  
    if (!paymentIntentId || !email || !niche) return  
res.status(400).json({ message: 'Missing' });
```

```
    // Validate the PaymentIntent status  
    const paymentIntent = await  
stripe.paymentIntents.retrieve(paymentIntentId);  
    if (paymentIntent.status !== 'succeeded') return  
res.status(400).json({ message: 'Payment not successful' });
```

```
    // Generate leads (placeholder): returns path to CSV  
    const filePath = await generateLeads(niche, 500);
```

```
    // Email CSV to customer  
    await sendEmail(email, `Your ${niche} leads — LeadsIn72`,  
`Thanks — attached are your ${niche} leads.`, filePath);
```

```
    // Persist order to simple JSON file  
    const ordersFile = config.dbPath;  
    let orders = [];
```

```
if (fs.existsSync(ordersFile)) orders =
JSON.parse(fs.readFileSync(ordersFile));
const order = {
  id: `ord_${Date.now()}`,
  paymentIntentId,
  email,
  niche,
  amount: paymentIntent.amount,
  status: 'delivered',
  file: filePath,
  createdAt: new Date().toISOString()
};
orders.unshift(order);
fs.writeFileSync(ordersFile, JSON.stringify(orders, null, 2));

res.json({ success: true });
} catch (err) {
  console.error(err);
  res.status(500).json({ message: err.message });
}
});

module.exports = router;
```