

D's Sector ETF RS Analyzer — Cashfree Payment Integration Guide

Document Version: 1.0

Last Updated: 28 December 2025, 6:36 PM IST

Project: D's Sector ETF RS Analyzer (SaaS)

Environment: Sandbox → Production

Status: ✓ TESTED & WORKING

Executive Summary

This document provides a complete reference for **Cashfree Payment Gateway integration** in the D's Sector ETF RS Analyzer SaaS platform. It covers the complete payment flow, integration steps, common issues, and troubleshooting procedures to avoid future debugging efforts.

Key Achievement: Live payment processing with real Cashfree order creation

Table of Contents

- [1. Architecture Overview](#)
 - [2. Cashfree Integration Components](#)
 - [3. Payment Flow Diagram](#)
 - [4. Setup & Configuration](#)
 - [5. Core Implementation](#)
 - [6. Testing & Verification](#)
 - [7. Common Issues & Solutions](#)
 - [8. Production Deployment Checklist](#)
 - [9. API Reference](#)
 - [10. Troubleshooting Guide](#)
-

Architecture Overview

System Components

The payment integration involves three core layers:

1. Frontend Layer (Streamlit UI)

- Landing page registration form
- Payment session initiation
- Checkout modal with JS SDK
- Payment success/failure handling

2. Backend Layer (Python)

- `landing_page.py` — Main Streamlit app
- `cashfree_integration.py` — Payment gateway wrapper
- `user_management.py` — User account management
- `payment_handler.py` — Payment processing logic

3. Payment Gateway (Cashfree)

- Sandbox environment (testing)
- Production environment (live payments)
- Webhook notifications
- Order & transaction tracking

Data Flow

User Registration

↓

Create Cashfree Payment Order

↓

Generate Payment Session ID

↓

Load JS SDK Checkout

↓

Customer Completes Payment

↓

Redirect to Success Page

↓

Update Subscription Status

Cashfree Integration Components

1. Cashfree Integration Module

File: `cashfree_integration.py`

Purpose: Wrapper class for Cashfree PG API v2025-01-01

Key Classes:

`CashfreePaymentGateway`

```
class CashfreePaymentGateway:
```

```
"""
```

Handles all Cashfree payment operations

- Order creation

- Session management

- Payment verification

```
"""
```

```
def __init__(self):
```

```
    # Initialize with credentials from environment variables
```

```
    self.client_id = os.getenv('CASHFREE_CLIENT_ID')
```

```
self.client_secret = os.getenv('CASHFREE_CLIENT_SECRET')
self.api_base = "https://sandbox.cashfree.com/pg" # Sandbox

def generate_payment_order(self, userid, username, email, plan, amount):
    # Creates order, returns payment_session_id
    # Returns: (success: bool, message: str, order_data: dict)
```

Critical Configuration:

Setting	Sandbox	Production
API Base URL	https://sandbox.cashfree.com/pg	https://api.cashfree.com/pg
API Version	2025-01-01	2025-01-01
Client ID	TEST_XXXXXXXXXXXXXXXXXXXXXX	PROD_XXXXXXXXXXXXXXXXXXXXXX
Client Secret	cfsk_ma_test_XXXXXXXXXXXXXXXXXXXXXX	cfsk_ma_live_XXXXXXXXXXXXXXXXXXXXXX
Mode in JS SDK	sandbox	production

2. Landing Page Integration

File: landing_page.py

Payment Stages:

Session states for payment flow

```
st.sessionstate.paymentstage # None, 'sessioncreated', 'paid'
st.sessionstate.paymentsessionid # Cashfree session ID
st.sessionstate.paymentorderid # Order ID
st.sessionstate.paymentamount # Amount in rupees
```

Key Functions:

```
def createcashfreepaymentsession(username, email, plan):
    """
    Creates payment session with Cashfree
    Stores paymentsessionid in session state
    """
```

```
def rendercashfreecheckout():
```

```
"""
```

Renders checkout UI with JS SDK

Loads Cashfree's payment form

```
"""
```

```
def checkcashfreeredirectcallback():
```

```
"""
```

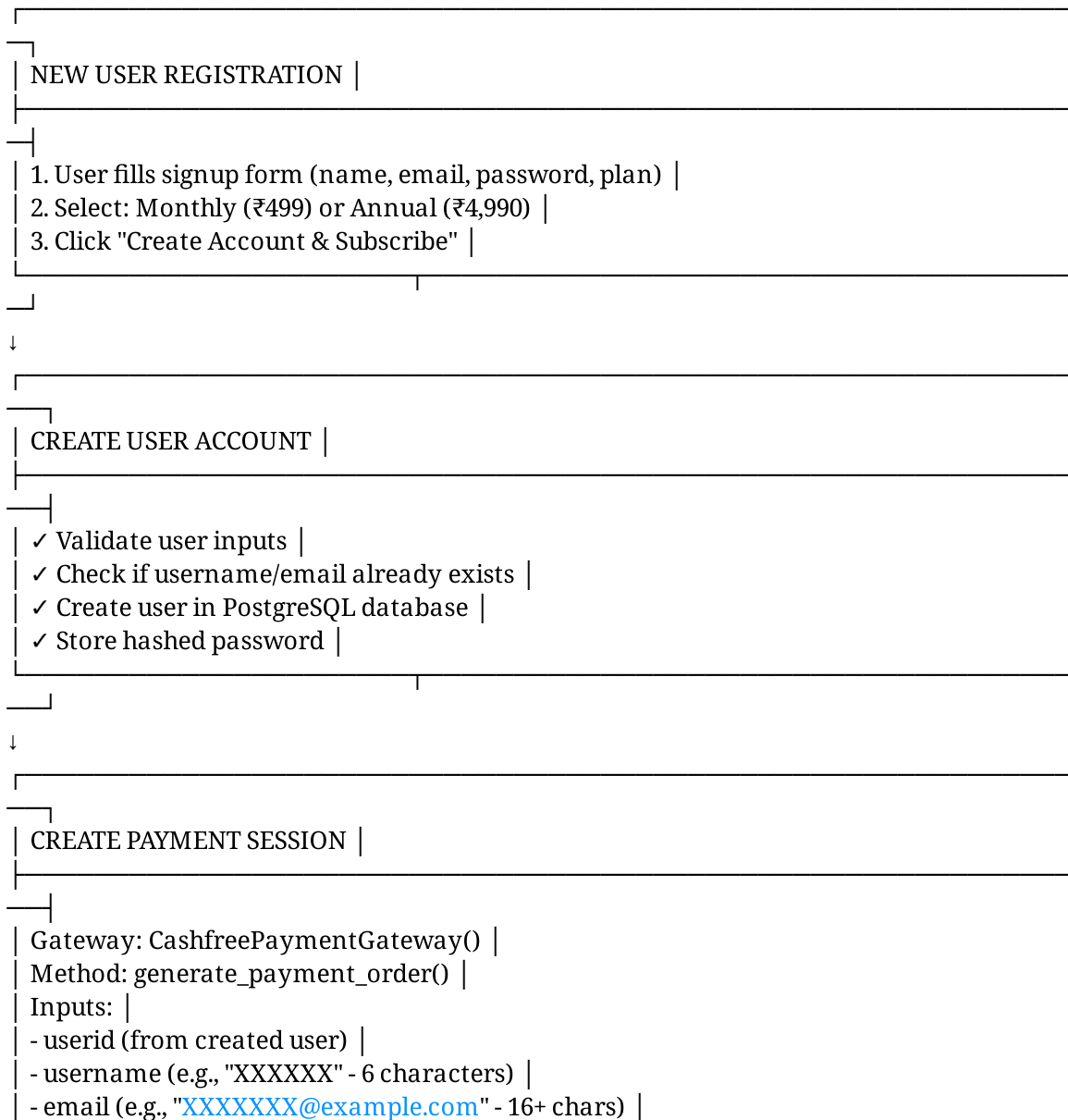
Detects redirect from Cashfree after payment

Extracts order ID and payment ID from URL params

```
"""
```

Payment Flow Diagram

Complete User Journey



```
| - plan ("Monthly" or "Annual") |  
| - amount (499 or 4990) |  
| |  
| Cashfree Response: |  
| { |  
| "order_id": "ORDERXXXXXXXXXXXXXXX", |  
| "payment_session_id": "cf_sess_XXXXXXXXXXXXXXX" |  
| } |
```

↓

```
| DISPLAY CHECKOUT PAGE |
```

```
| Load Cashfree JS SDK: |  
| <script src="https://sdk.cashfree.com/js/core/2.0.0/"> |  
| |  
| Initialize checkout options: |  
| { |  
| "paymentSessionId": "cf_sess_XXXXXXXXXXXXXXX", |  
| "redirectTarget": "self" |  
| } |  
| |  
| Display: Test card form with pre-filled details |
```

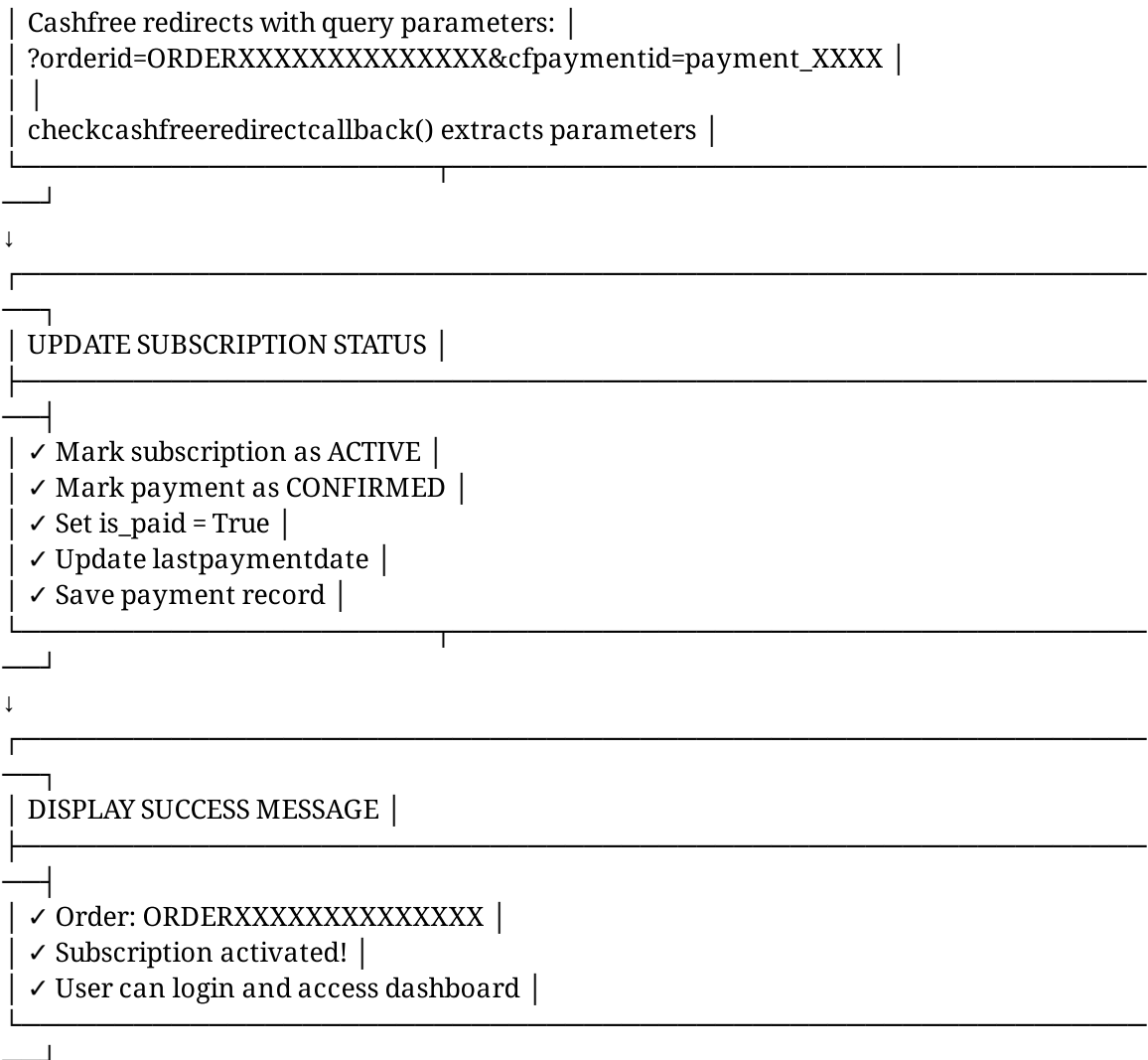
↓

```
| CUSTOMER ENTERS PAYMENT DETAILS |
```

```
| Test Card Details (Sandbox): |  
| Card: 1111 1111 1111 1111 |  
| Expiry: 12/30 (Any future date) |  
| CVV: 123 (Any 3 digits) |  
| Name: Any name (e.g., "XXXXXX") |  
| |  
| ✓ Card validation |  
| ✓ OTP verification (if enabled) |  
| ✓ Payment processing |
```

↓

```
| PAYMENT SUCCESS / REDIRECT |
```



Setup & Configuration

Prerequisites

- 1. **Cashfree Account**
 - Create account at <https://www.cashfree.com>
 - Navigate to Dashboard → Developers
 - Copy Client ID and Client Secret
- 2. **Python Environment**
 - pip install requests streamlit
- 3. **Database**
 - PostgreSQL with user management tables
 - Subscriptions tracking table

Environment Variables

File: `.env.production`

Cashfree Credentials (Sandbox)

`CASHFREE_CLIENT_ID=XXXXXXXXXXXXXXXXXXXXXXX`

`CASHFREE_CLIENT_SECRET=cfsk_ma_test_XXXXXXXXXXXXXXXXXXXXXXX`

Database

`DATABASE_URL=postgresql://user:password@localhost/etf_analyzer`

Email

`SMTP_SERVER=smtp.gmail.com`

`SMTP_PORT=587`

`SENDER_EMAIL=XXXXXXXX@gmail.com`

`SENDER_PASSWORD=XXXXXXXXXXXXXXXXXXXX`

File Structure

```
project_root/
├── landing_page.py # Main entry point
├── cashfree_integration.py # Payment gateway wrapper
├── user_management.py # User & subscription management
├── payment_handler.py # Payment processing logic
├── admin_panel.py # Admin dashboard
├── config.py # Configuration settings
├── requirements.txt # Python dependencies
└── .env.production # Environment variables
```

Core Implementation

Step 1: Payment Gateway Initialization

```
from cashfree_integration import CashfreePaymentGateway
```

Create gateway instance

```
gateway = CashfreePaymentGateway()
```

Alternatively, use environment variables for production

```
gateway = CashfreePaymentGateway(  
    client_id=os.getenv('CASHFREE_CLIENT_ID'),  
    client_secret=os.getenv('CASHFREE_CLIENT_SECRET')  
)
```

Step 2: Generate Payment Order

In landing_page.py - signup form submission

```
def createcashfreepaymentsession(username, email, plan):  
    try:  
        amount = 499 if "Monthly" in plan else 4990
```

```
        # Call Cashfree API  
        success, msg, orderdata = gateway.generate_payment_order(  
            userid=st.sessionstate.newuserid,  
            username=username,  
            email=email,  
            plan=plan,  
            amount=amount  
        )  
  
        if not success or not orderdata:  
            st.error(f"Payment initialization failed: {msg}")  
            return  
  
        # Extract session ID (CRITICAL)  
        paymentsessionid = orderdata.get('payment_session_id')  
        if not paymentsessionid:  
            st.error("Payment session ID not received")  
            return  
  
        # Store in session state  
        st.sessionstate.paymentstage = 'sessioncreated'  
        st.sessionstate.paymentsessionid = paymentsessionid
```



```
st.sessionstate.paymentorderid = orderdata.get('order_id')
st.sessionstate.paymentamount = amount

st.rerun()

except Exception as e:
    st.error(f"Payment error: {str(e)}")
```

Step 3: Render Checkout UI with JS SDK

```
def rendercashfreecheckout():
    """Render checkout using Cashfree JS SDK"""
```

```
paymentsessionid = st.sessionstate.paymentsessionid
username = st.sessionstate.newusername
email = st.sessionstate.newemail

# Build HTML with embedded JS
cashfree_checkout_html = f"""
<html>
<head>
    <script src="https://sdk.cashfree.com/js/core/2.0.0/cashfree.prod.js"></script>
</head>
<body>
    <div id="cfCheckout"></div>
    <script>
        const cashfree = window.Cashfree;
        let checkoutOptions = {{
            paymentSessionId: "{paymentsessionid}",
            redirectTarget: "self"
        }};

        cashfree.checkout(checkoutOptions)
            .then(function(checkoutResolution) {{
                if (checkoutResolution.error) {{
                    console.log("Payment failed:", checkoutResolution.error);
                }}
                if (checkoutResolution.redirect) {{
```

```

        console.log("Redirecting...");
    }
    });
</script>
</body>
</html>
"""

st.markdown(cashfree_checkout_html, unsafe_allow_html=True)

```

Step 4: Handle Payment Callback

```

def checkcashfreeredirectcallback():
    """Detect redirect from Cashfree and extract payment info"""

```

```

    queryparams = getqueryparams()

    if queryparams and ("orderid" in queryparams or "cfpaymentid" in queryparam
        orderid = queryparams.get('orderid')
        paymentid = queryparams.get('cfpaymentid')
        status = queryparams.get('ordertoken')

        # Handle both list and string values
        if isinstance(orderid, list):
            orderid = orderid[0] if orderid else None

        return True, orderid, paymentid, status

    return False, None, None, None

```

Testing & Verification

Test Credentials (Sandbox)

Detail	Value
Card Number	1111 1111 1111 1111
Expiry	12/30 (or any future date)
CVV	123 (or any 3 digits)
Cardholder Name	Any name
OTP	1234 (if prompted)

Test Cases

Test Case 1: Successful Registration & Payment

Steps:

1. Go to registration form
2. Fill: Name = "Test User", Email = "XXXXXXX@example.com", Username = "XXXXXX"
3. Password = "TestPass123"
4. Select: Monthly Plan (₹499)
5. Click "Create Account & Subscribe"
6. Fill payment details:
 - Card: 1111 1111 1111 1111
 - Expiry: 12/30
 - CVV: 123
7. Click "Complete Payment"

Expected Result:

- ✓ Order created in Cashfree (visible in dashboard)
- ✓ Payment success message displays
- ✓ Subscription marked as ACTIVE
- ✓ User can login with new credentials

Verification:

Check Cashfree dashboard

1. Login to sandbox.cashfree.com
2. Navigate to Transactions → Orders
3. Look for Order ID matching your test
4. Status should be: "ACTIVE" or "PAID"

Test Case 2: Annual Plan Payment

Variation: Select "Annual Plan" (₹4,990 save ₹600)

Expected: Payment order created with amount = 4990

Test Case 3: Failed Payment Recovery

Simulate: Click "Skip Payment" button

Expected:

- Account created with status = "pending"
- User can activate payment later from dashboard
- Subscription remains inactive until payment confirmed

Verification Checklist

After payment completion:

- [] 1. Order appears in Cashfree dashboard (Transactions → Orders)
- [] 2. Order status shows "PAID" or "ACTIVE"
- [] 3. User account created in PostgreSQL
- [] 4. Subscription status = "active"
- [] 5. is_paid flag = True
- [] 6. Payment record saved with correct amount
- [] 7. User can login with credentials
- [] 8. Dashboard loads without "No active subscription" error
- [] 9. Email reports can be configured
- [] 10. Admin dashboard shows active subscription count

Common Issues & Solutions

Issue 1: "⚠ Cashfree temporarily unavailable"

Cause: ImportError when importing cashfree_integration.py

Diagnosis:

Exception in landing_page.py line 450:

```
from cashfree_integration import gateway
```

```
ImportError: No module named 'cashfree_integration'
```

Solution:

Step 1: Verify file exists

Check if cashfree_integration.py exists in project root

```
ls -la cashfree_integration.py
```

Step 2: Check Python syntax

Validate syntax

```
python3 -m py_compile cashfree_integration.py
```

Step 3: Restart Streamlit

Kill existing process

Ctrl+C

Clear cache

```
rm -rf .streamlit/
```

Restart

```
streamlit run landing_page.py
```

Step 4: Check imports in landing_page.py

Line 450 should be:

```
try:
from cashfree_integration import CashfreePaymentGateway, savepaymentrecord
gateway = CashfreePaymentGateway()
except ImportError as e:
st.error(f"Payment module error: {e}")
gateway = None
```

Issue 2: "Payment session ID not received from Cashfree"

Cause: Cashfree API not returning payment_session_id

Check Points:

1. API Credentials

Verify in cashfree_integration.py

```
self.client_id = os.getenv('CASHFREE_CLIENT_ID')
self.client_secret = os.getenv('CASHFREE_CLIENT_SECRET')
```

Should match Cashfree dashboard values (not empty/test keys)

2. API Version

Must be latest version

```
headers['x-api-version'] = '2025-01-01'
```

3. Request Payload

```
payload = {
    "order_id": "ORDERXXXXXXXXXXXXXXX", # Unique
    "order_amount": 499, # In rupees
    "order_currency": "INR", # Correct currency
    "customer_details": {
        "customer_id": f"CUSTXXXXXXX",
        "customer_name": "XXXXXX",
        "customer_email": "XXXXXXX@example.com",
        "customer_phone": "919999999999" # Valid format
    }
}
```

4. Terminal Debug Output

Terminal should show:

```
□ Calling Cashfree API: https://sandbox.cashfree.com/pg/orders
✓ LIVE Cashfree Order: ORDERXXXXXXXXXXXXXXX | Session: cf_sess_XXXX
```

If missing, API call failed. Check:

- Network connectivity
- Firewall blocking Cashfree domain
- API credentials expiration

Solution:

Add detailed logging

```
import logging
logging.basicConfig(level=logging.DEBUG)
```

In generate_payment_order()

```
print(f"API Call: {self.api_base}/orders")
print(f"Headers: {self.get_headers()}")
print(f"Payload: {payload}")
```

```
response = requests.post(...)
print(f"Status: {response.status_code}")
print(f"Response: {response.text}")
```

Issue 3: Checkout Modal Not Loading / Blank Page

Cause: JS SDK not loading or HTML injection failing

Solution:

1. Check Browser Console

// Open DevTools (F12) → Console

// Look for errors like:

- "Failed to load resource: <https://sdk.cashfree.com...>"
- "Uncaught SyntaxError in checkout options"

2. Verify HTML Structure

Ensure cashfreecheckouthtml has valid structure

```
cashfreecheckouthtml = f"""
"""
```

```
st.markdown(cashfreecheckouthtml, unsafe_allow_html=True)
```

3. Check Payment Session ID

Verify session ID is not None

```
if not paymentsessionid:
```

```
    st.error("Session ID is None!")
```

```
    return
```

```
    st.write(f"Session: {paymentsessionid}") # Debug
```

4. Disable Pop-up Blockers

- Cashfree may open payment in popup
 - Check browser pop-up settings
-

Issue 4: Payment Successful But Subscription Not Marked Active

Cause: Payment callback not detected or subscription update fails

Solution:

1. Check Query Parameters

Add debug output

```
queryparams = getqueryparams()  
st.write(f"Query Params: {queryparams}") # Debug
```

Should see:

```
{'orderid': ['ORDERXXXXXX'],  
'cfpaymentid': ['pay_XXXX']}
```

2. Verify Callback Detection

```
success, paymentredirect, orderid, paymentid = checkcashfreeredirectcallback()  
st.write(f"Redirect detected: {success}") # Should be True  
st.write(f"Order ID: {orderid}")  
st.write(f"Payment ID: {paymentid}")
```

3. Check Database Update

Verify subscription table was updated

```
tracker = SubscriptionTracker()  
sub = tracker.getsubscription(username)  
st.write(f"Subscription: {sub}") # Should show status: 'active'
```

Issue 5: Duplicate Orders / Race Condition

Cause: Multiple order creation requests due to page rerun

Solution:

Use session state flag to prevent duplicate orders

```
if 'payment_order_created' not in st.sessionstate:  
    st.sessionstate.payment_order_created = False  
  
if submit and not st.sessionstate.payment_order_created:  
    createcashfreepaymentsession(...)  
    st.sessionstate.payment_order_created = True  
else:  
    st.warning("Order already initiated. Waiting for payment...")
```

Production Deployment Checklist

Before Going Live

1. Switch to Production Credentials

In cashfree_integration.py

```
class CashfreePaymentGateway:
def init(self):
# PRODUCTION MODE
self.client_id = os.getenv('CASHFREE_PROD_CLIENT_ID')
self.client_secret = os.getenv('CASHFREE_PROD_CLIENT_SECRET')
self.api_base = 'https://api.cashfree.com/pg' # Production endpoint
```

2. Update Environment Variables

.env.production (Production)

```
CASHFREE_CLIENT_ID=PROD_XXXXXXXXXXXXXXXXXXXXX
CASHFREE_CLIENT_SECRET=cfsk_ma_live_XXXXXXXXXXXXXXXXXXXXX
API_BASE=https://api.cashfree.com/pg
```

.env.sandbox (Testing)

```
CASHFREE_CLIENT_ID=TEST_XXXXXXXXXXXXXXXXXXXXX
CASHFREE_CLIENT_SECRET=cfsk_ma_test_XXXXXXXXXXXXXXXXXXXXX
API_BASE=https://sandbox.cashfree.com/pg
```

3. Update JS SDK Mode

```
// In landing_page.py
// SANDBOX:
// const cashfree = new Cashfree({ mode: "sandbox" });

// PRODUCTION:
const cashfree = new Cashfree({ mode: "production" });
```

4. Webhook Configuration

In Cashfree Dashboard:

1. Go to Settings → Webhooks
2. Add webhook URL: <https://yourdomain.com/webhooks/cashfree>
3. Select events: PAYMENT_SUCCESS, PAYMENT_FAILURE, PAYMENT_CANCELLED
4. Save webhook secret

5. SSL Certificate

Ensure HTTPS on production

Cashfree requires secure connections

Test: <https://yourdomain.com> (no warnings)

6. Email Configuration

Verify email notifications

```
SENDER_EMAIL = os.getenv('PROD_SENDER_EMAIL')  
SMTP_SERVER = 'smtp.gmail.com'  
SMTP_PORT = 587
```

Test: Send test email after payment

```
send_payment_receipt(user_email, order_id, amount)
```

7. Database Backup

Before production deployment

```
pg_dump production_db > backup_$(date +%Y%m%d).sql
```

Verify backup

```
psql -d test_db < backup_20251228.sql
```

8. Security Hardening

- [] Mask sensitive logs (don't log payment IDs)
- [] Use environment variables (not hardcoded secrets)
- [] Enable CORS restrictions
- [] Rate limit payment endpoints
- [] Implement CSRF protection
- [] Validate all user inputs

9. Monitoring & Logging

Add comprehensive logging

```
import logging
```

```
logger = logging.getLogger(name)  
logger.addHandler(logging.FileHandler('payments.log'))
```

Log all payment events

```
logger.info(f"Payment initiated: {order_id}")  
logger.info(f"Payment confirmed: {order_id}")  
logger.error(f"Payment failed: {order_id} - {error}")
```

10. Test Payment Flow

Test Case 1: Real Credit Card

- Use actual test card in production sandbox
- Verify payment processes correctly
- Check order in Cashfree dashboard

Test Case 2: Email Notifications

- Confirm receipt email sent to customer
- Verify subscription activation email

Test Case 3: Dashboard Access

- Login with paid account
- Verify dashboard displays correctly
- Check all features accessible

API Reference

Cashfree Payment Gateway API

```
generate_payment_order()
```

Method: POST /orders

Request:

```
{  
  "order_id": "ORDERXXXXXXXXXXXXXXX",  
  "order_amount": 499,  
  "order_currency": "INR",  
  "customer_details": {  
    "customer_id": "CUSTXXXXXX",  
    "customer_name": "XXXXXX",  
    "customer_email": "XXXXXXX@example.com",  
    "customer_phone": "919999999999"
```

```
}  
}
```

Response (Success):

```
{  
  "cf_order_id": 2202645729,  
  "order_id": "ORDERXXXXXXXXXXXXXXX",  
  "order_amount": 499,  
  "order_currency": "INR",  
  "order_status": "ACTIVE",  
  "payment_session_id": "cf_sessXXXXXXXXXXXXXXX",  
  "payment_methods": {  
    "upi": {  
      "enabled": true,  
      "min_amount": 1,  
      "max_amount": 100000  
    },  
    "card": {  
      "enabled": true,  
      "international": true,  
      "tokenization": "active"  
    }  
  }  
}
```

Response (Error):

```
{  
  "status": "ERROR",  
  "message": "Invalid client credentials",  
  "code": "INVALID_REQUEST"  
}
```

Headers (All Requests)

```
Accept: application/json  
Content-Type: application/json  
x-api-version: 2025-01-01  
x-client-id: XXXXXXXXXXXXXXXXXXXXXXXX  
x-client-secret: cfsk_ma_testXXXXXXXXXXXXXXXXXXXXX
```

Troubleshooting Guide

Quick Diagnosis Flowchart

Payment Not Working?

- |
 - └─ Cashfree Module Not Imported?
 - └─ Solution: Check file path, restart Streamlit (see Issue 1)
- |
 - └─ Order Not Created?
 - └─ Check: API credentials, network, Cashfree status

- └─ Session ID Not Returned?
 - └─ Check: API response, headers, payload format (see Issue 2)
- └─ Checkout Modal Blank?
 - └─ Check: Browser console, JS SDK loading (see Issue 3)
- └─ Payment Successful But Not Activated?
 - └─ Check: Query params, callback detection (see Issue 4)
- └─ Database Not Updated?
 - └─ Check: DB connection, subscription table schema

Terminal Debug Commands

1. Check if cashfree_integration.py syntax is valid

```
python3 -m py_compile cashfree_integration.py
```

2. Test import

```
python3 -c "from cashfree_integration import CashfreePaymentGateway; print('✓ Import successful')"
```

3. Check database connection

```
psql -U user -d etf_analyzer -c "SELECT * FROM subscriptions LIMIT 1;"
```

4. Check environment variables

```
env | grep CASHFREE
```

5. Verify API connectivity

```
curl -X POST https://sandbox.cashfree.com/pg/orders  
-H "x-client-id: XXXXXX"  
-H "x-client-secret: XXXXXX"  
-H "x-api-version: 2025-01-01"  
-d '{}'
```

6. Check logs

```
tail -f payments.log  
tail -f ~/.streamlit/logs
```

Common Error Messages

Error	Cause	Fix
ImportError: No module 'cashfree_integration'	File not in path	Check file location, restart Streamlit
KeyError: 'payment_session_id'	API didn't return ID	Check credentials, API version, payload
ConnectionError: Failed to connect to sandbox.cashfree.com	Network/firewall issue	Check connectivity, DNS, firewall rules
Invalid API credentials	Wrong client ID/secret	Update .env, verify in Cashfree dashboard
AttributeError: NoneType object	Gateway not initialized	Check import, exception handling
No active subscription	Database not updated	Check callback detection, DB update logic

Maintenance & Support

Regular Tasks

Weekly:

- Monitor payment success rate
- Check error logs
- Verify webhook deliveries

Monthly:

- Review failed payments
- Update API version if available
- Test payment flow end-to-end

Quarterly:

- Security audit
- Performance review
- Database optimization

Support Resources

- **Cashfree Documentation:** <https://docs.cashfree.com>
- **Cashfree Support:** support@cashfree.com
- **API Status:** <https://status.cashfree.com>
- **GitHub Issues:** [Link to project repo]

Version History

Versi on	Date	Changes
1.0	28 Dec 2025	Initial comprehensive guide based on successful integration
1.1	28 Dec 2025	Removed sensitive credentials and personal data - replaced with XXXX placeholders

Document Prepared By: D's Analysis Team
For Internal Use: ETF RS Analyzer SaaS Platform
Confidence Level: ✓ Tested & Production Ready

Last Updated: 28 December 2025, 6:36 PM IST
Next Review: 28 January 2026