

ING Demo App - Complete Feature Documentation

Version: 1.0

Date: October 15, 2025

Project: ING Banking Demo with Optimizely Integration

Table of Contents

1. [Overview](#)
 2. [Core Banking Features](#)
 3. [Dynamic Content Features](#)
 4. [Optimizely A/B Testing Features](#)
 5. [Technical Architecture](#)
 6. [User Interface Components](#)
 7. [Data & Analytics](#)
-

Overview

The ING Demo App is a comprehensive banking application built with React, TypeScript, and Optimizely Feature Experimentation. It showcases modern banking features with integrated A/B testing capabilities for optimizing user experience and conversion rates.

Key Technologies:

- React 19.2.0
- TypeScript 4.9.5
- Optimizely React SDK 3.3.0
- Styled Components 6.1.19
- React Router 7.9.4

Live URL: <http://localhost:3000>

GitHub Repository: <https://github.com/nacfigueiredoopti/ing-demo-app>

Core Banking Features

1. Dashboard

The main landing page featuring an overview of user accounts and financial status.

Features:

- Welcome hero banner with personalized greeting
- Account overview cards (Checking, Savings, Investment)
- Real-time balance display
- Quick action buttons
- Feature flag controlled layout options

Account Cards:

- **Primary Checking:** €4,532.50
- **Emergency Fund Savings:** €12,450.00 (2.5% interest rate)
- **Investment Portfolio:** €28,750.00 (+12.3% this year)

2. Accounts Page

Detailed view of all user accounts with transaction capabilities.

Features:

- Complete account listing
- Account type indicators
- Balance information
- Interest rates display
- Quick transfer functionality (feature flagged)

3. Transfer Functionality

Secure money transfer system between accounts.

Features:

- From/To account selection
- Amount input with validation
- Transfer confirmation
- Event tracking for completed transfers
- Real-time balance updates

Tracked Data:

- Transfer amount
- Source account
- Destination account
- Timestamp

4. Savings Goals

Goal-based savings tracking system (feature flagged).

Features:

- Multiple savings goals support
- Progress visualization
- Target amount tracking
- Visual progress bars
- Goal creation tracking

Example Goals:

- Vacation: €3,200 / €5,000 (64% complete)
- New Car: €8,500 / €20,000 (42.5% complete)

Dynamic Content Features

1. Spending Insights

Comprehensive spending analytics and budget tracking.

Features:

- Monthly spending overview
- Budget remaining calculator
- Average daily spending
- Category breakdown with percentages
- Visual progress bars

Metrics Displayed:

- Total spending: €3,560
- Budget: €4,000
- Budget remaining: €440
- Savings rate: 11%
- Average daily spending: €254.29

Spending Categories:

- Housing: €1,250 (35%)
- Food & Dining: €680 (19%)
- Transportation: €450 (13%)
- Entertainment: €320 (9%)
- Shopping: €580 (16%)
- Utilities: €280 (8%)

2. Transaction History

Complete transaction log with filtering capabilities.

Features:

- 14 realistic transactions
- Filter by type (All, Income, Expenses, Transfers)
- Transaction icons and categories
- Date and time stamps
- Account attribution
- Color-coded amounts (green for income, red for expenses)

Transaction Types:

- Income transactions (salary, freelance)
- Expense transactions (rent, utilities, shopping)
- Transfer transactions (savings transfers)

Sample Transactions:

- Salary Deposit: +€3,500
- Rent Payment: -€1,250
- Electricity Bill: -€120
- Freelance Project: +€800
- Netflix Subscription: -€15

3. Market Overview

Real-time financial market data and stock prices.

Features:

- Major market indices tracking
- Currency exchange rates
- Commodity prices
- Top Dutch stocks listing
- Price change indicators
- Color-coded performance (green/red)

Indices & Commodities:

- AEX Index: 789.45 (+1.24%)
- EUR/USD: 1.0895 (+0.32%)
- Gold: €1,987 (-0.15%)
- Oil (Brent): €78.50 (+2.10%)

Top Dutch Stocks:

- ASML Holding: €745.30 (+2.45%)
- Shell PLC: €29.85 (+1.20%)
- ING Group: €14.92 (+0.85%)
- Philips: €23.10 (-0.45%)
- Heineken: €89.50 (+1.15%)
- Unilever: €48.75 (-0.30%)

4. Smart Money Tips

Personalized financial advice and recommendations.

Features:

- 6 actionable financial tips
- Personalized insights based on spending
- Goal-oriented recommendations
- Educational content
- Special product recommendations

Tips Included:

1. **Build Your Emergency Fund** - Currently at 2.8 months of expenses
2. **Review Your Spending** - Dining optimization suggestion
3. **Set Clear Goals** - Goal-setting increases savings by 42%
4. **Automate Your Savings** - €50/week = €2,600/year
5. **Invest for Long-term** - Retirement planning advice
6. **Track Subscriptions** - Average €75/month in unused subscriptions

Personalized Recommendation:

Premium Savings Account offer with 2.5% cashback and potential €245 extra interest annually.

5. Financial News & Insights

Curated financial news feed with categorized articles.

Features:

- 6 current financial articles
- Category badges (Markets, Investing, Economy, Crypto)
- Article excerpts
- Source attribution
- Timestamp information
- Color-coded categories

News Categories:

- Markets (blue)
- Investing (green)
- Economy (orange)
- Crypto (purple)

Sample Articles:

- "European Markets Rally as ECB Holds Rates Steady"
- "Sustainable Investing Reaches New Heights in 2025"
- "Inflation Shows Signs of Cooling in Eurozone"
- "Digital Euro Pilot Program Expands"

Optimizely A/B Testing Features

Feature Flags Configuration

All feature flags are configured in `src/optimizely.config.ts`

1. checkout_button_color

Purpose: Test different CTA button colors for conversion optimization

Flag Key: checkout_button_color

Variable: color (String)

Default Value: white

Variations:

- White (control)
- Green (#00A755)
- Blue (#1976D2)
- Purple (#7B1FA2)

Location: Dashboard hero banner "Get Started" button

Primary Metric: cta_clicked event

Use Case: Optimize call-to-action button color to maximize click-through rates and user engagement.

2. new_dashboard_layout

Purpose: Test different dashboard grid layouts

Flag Key: new_dashboard_layout

Variable: layout (String)

Default Value: two-column

Variations:

- Two-column (control)
- Three-column
- Sidebar layout

Location: Main dashboard account cards grid

Primary Metric: dashboard_view event

Use Case: Determine which layout provides better information hierarchy and user engagement.

3. hero_banner_variant

Purpose: Test different hero banner styles and sizes

Flag Key: hero_banner_variant

Variables:

- variant (String): default, gradient, dark
- size (String): default, large

Default Values:

- variant: default
- size: default

Variations:

- Control: Default orange background, default size
- Gradient: Orange gradient background, large title
- Dark: Dark background, default size

Location: Dashboard hero banner

Primary Metric: cta_clicked event

Use Case: Optimize hero banner visual design for maximum impact and conversion.

4. show_savings_goals

Purpose: Feature rollout for savings goals functionality

Flag Key: show_savings_goals

Type: Boolean flag

Default: Disabled

Location: Dashboard - appears below account cards

Primary Metric: savings_goal_created event

Use Case: Gradually roll out savings goals feature to measure adoption and engagement before full release.

5. enable_quick_transfer

Purpose: Test quick transfer button on account cards

Flag Key: enable_quick_transfer

Type: Boolean flag

Default: Disabled

Location: Account cards (Quick Transfer button)

Primary Metric: transfer_completed event

Use Case: Test if adding a quick transfer button increases transfer usage and user convenience.

6. navigation_style

Purpose: Test different navigation bar styles

Flag Key: navigation_style

Variable: style (String)

Default Value: default

Variations:

- Default: Standard padding and font size
- Compact: Reduced padding and smaller font
- Bold: Increased font weight

Location: Header navigation bar

Primary Metric: page_view event

Use Case: Optimize navigation style for better usability and visual hierarchy.

Event Tracking

All events are configured in src/optimizely.config.ts

1. dashboard_view

Event Key: dashboard_view

Description: Triggered when users view the dashboard

Location: Dashboard component (on mount)

Use Case: Measure dashboard engagement and layout effectiveness

2. cta_clicked

Event Key: cta_clicked

Description: Triggered when users click call-to-action buttons

Location: Dashboard "Get Started" button

Tracked Properties:

- buttonColor: Color variant of the button
- location: Where the button was clicked (hero_banner)

Use Case: Primary metric for button color and hero banner experiments

3. transfer_completed

Event Key: transfer_completed

Description: Triggered when users complete a transfer

Location: Transfer page

Tracked Properties:

- fromAccount: Source account
- toAccount: Destination account
- amount: Transfer amount

Use Case: Measure transfer feature usage and optimize transfer flow

4. savings_goal_created

Event Key: savings_goal_created

Description: Triggered when users create a savings goal

Location: Savings page

Tracked Properties:

- goalName: Name of the goal
- targetAmount: Target savings amount

Use Case: Measure savings goals feature adoption

5. page_view

Event Key: page_view

Description: General page view tracking

Location: Available for all pages

Use Case: General engagement and navigation tracking

Technical Architecture

Tech Stack

Frontend Framework:

- React 19.2.0
- TypeScript 4.9.5
- React Router DOM 7.9.4

Styling:

- Styled Components 6.1.19
- Custom theme system
- Responsive design
- Mobile-first approach

A/B Testing:

- Optimizely React SDK 3.3.0
- Optimizely SDK 6.1.0
- Feature flag management
- Real-time event tracking

Development Tools:

- React Scripts 5.0.1
- TypeScript compiler
- ESLint
- Testing Library

Project Structure

```
ing-demo-app/
├── public/
│   ├── ing-logo.svg
│   ├── index.html
│   └── _redirects
├── src/
│   ├── components/
│   │   ├── Header.tsx
│   │   ├── Dashboard.tsx
│   │   ├── Accounts.tsx
│   │   ├── Transfer.tsx
│   │   ├── Savings.tsx
│   │   ├── FinancialNews.tsx
│   │   ├── MarketTrends.tsx
│   │   ├── SpendingInsights.tsx
│   │   ├── TransactionHistory.tsx
│   │   └── FinancialTips.tsx
│   ├── styles/
│   │   ├── theme.ts
│   │   └── GlobalStyles.ts
│   ├── optimizely.config.ts
│   ├── App.tsx
│   └── index.tsx
├── .env
├── netlify.toml
├── package.json
└── tsconfig.json
```

Theme Configuration

Primary Colors:

- Primary (ING Orange): #FF6200
- Secondary: #333333
- Background: #F5F5F5
- Border: #E0E0E0

Text Colors:

- Primary: #212121
- Secondary: #757575
- Disabled: #BDBDBD

Semantic Colors:

- Success/Income: #00A755
- Error/Expense: #D32F2F
- Warning: #F57C00
- Info: #1976D2

Spacing Scale:

- xs: 4px
- sm: 8px

- md: 16px
- lg: 24px
- xl: 32px

Border Radius:

- sm: 4px
- md: 8px
- lg: 12px
- full: 9999px

Shadows:

- sm: 0 1px 2px rgba(0,0,0,0.05)
- md: 0 4px 6px rgba(0,0,0,0.1)
- lg: 0 10px 15px rgba(0,0,0,0.1)

Breakpoints:

- mobile: 768px
- tablet: 1024px
- desktop: 1280px

User Interface Components

Header Component

Features:

- ING logo (clickable, returns to home)
- Navigation links (Home, Accounts, Transfer, Savings)
- Sticky positioning
- Responsive design
- Optimizely-controlled navigation style

Navigation Style Variants:

- Default: Standard padding
- Compact: Reduced padding
- Bold: Increased font weight

Card Components

Reusable card component used throughout the app

Features:

- White background
- Rounded corners
- Shadow effect
- Hover animations
- Responsive padding

Usage:

- Account cards
- Transaction items
- News articles
- Tip cards

Button Components

CTA Button:

- Color variants (white, green, blue, purple)
- Hover effects (translateY, shadow)
- Active states
- Responsive sizing
- Optimizely-controlled colors

Form Components

Input Fields:

- Account selection dropdowns
- Amount inputs
- Validation
- Error states

Progress Bars

Used in:

- Spending category breakdown
- Savings goals progress

Features:

- Animated width transitions
- Color-coded by category
- Percentage display
- Smooth animations

Data & Analytics

Dummy Data Sources

All data in the demo is realistic but simulated.

Account Data:

- 3 main accounts (Checking, Savings, Investment)
- Realistic balances
- Interest rates
- Account types

Transaction Data:

- 14 realistic transactions

- Multiple transaction types
- Categorized expenses
- Realistic dates and amounts

Market Data:

- 4 major indices/commodities
- 6 Dutch stocks
- Realistic prices
- Percentage changes

Spending Data:

- 6 spending categories
- Monthly totals
- Budget tracking
- Category percentages

News Data:

- 6 financial articles
- Multiple sources
- Realistic headlines
- Timestamps

Analytics Integration**Optimizely Event Tracking:**

- User interactions
- Feature usage
- Conversion events
- Page views

Debug Logging:

Console logs for development:

- SDK key status
- Feature flag decisions
- Variable values
- Event tracking

Deployment**Local Development**

```
# Install dependencies
npm install

# Set up environment
cp .env.example .env
# Add your Optimizely SDK key to .env

# Start development server
npm start

# Build for production
npm run build
```

Netlify Deployment

Configuration: netlify.toml

Build Settings:

- Build command: npm run build
- Publish directory: build
- Node version: Latest LTS

Environment Variables Required:

- REACT_APP_OPTIMIZEY_SDK_KEY

Features:

- Automatic deployments from GitHub
- Preview deployments for pull requests
- Custom domain support
- HTTPS enabled
- React Router redirects configured

Deployment URL: TBD (after deployment)

Optimizely Setup Guide

Prerequisites

1. Optimizely Feature Experimentation account
2. Project created in Optimizely
3. SDK key obtained

Setup Steps

1. **Create Feature Flags** (6 flags listed above)
2. **Create Events** (5 events listed above)
3. **Get SDK Key** from Settings → Environments
4. **Configure .env** file with SDK key
5. **Create Experiments** in Optimizely dashboard
6. **Set Traffic Allocation** (typically 100% for demos)

7. **Define Audiences** (optional)
8. **Set Primary Metrics** for each experiment

Testing Experiments

Multiple Users:

- Use incognito windows
- Clear cookies between tests
- Change user ID in App.tsx:23

Real-time Changes:

- Toggle flags in Optimizely dashboard
- Refresh app to see changes
- Monitor events in real-time

Debug Mode:

- Check browser console
 - Look for Optimizely logs
 - Verify SDK initialization
 - Monitor network requests
-

Feature Roadmap

Planned Features

- User authentication
- Real-time balance updates
- Mobile app version
- Investment portfolio details
- Credit card management
- Bill payment system
- Budget planning tools
- Financial advisor chat
- Document uploads
- Multi-language support

Optimization Opportunities

- Additional A/B tests for color schemes
 - Layout experiments for mobile
 - Onboarding flow optimization
 - Navigation structure tests
 - Content personalization
 - AI-powered recommendations
-

Support & Documentation

Resources

- **Optimizely Setup Guide:** OPTIMIZEZY_SETUP.md
- **Quick Start Guide:** QUICKSTART.md
- **Feature Flags Reference:** FEATURE_FLAGS_REFERENCE.md
- **Project Summary:** PROJECT_SUMMARY.md
- **GitHub Repository:** <https://github.com/nacfigueiredoopti/ing-demo-app>

Browser Console Debugging

Open browser DevTools (F12) to see:

- Optimizely initialization logs
- Feature flag decisions
- Event tracking
- SDK key status
- Variable values

Common Issues

Flags not working:

- Verify SDK key in .env
- Check flag names (case-sensitive)
- Ensure experiment is running
- Check browser console for errors

Events not tracking:

- Verify event names match exactly
- Check network tab for requests
- Ensure user ID is set
- Look for JavaScript errors

Build errors:

- Clear node_modules and reinstall
- Check for TypeScript errors
- Verify all imports

Credits

Built with:

- React
- TypeScript
- Optimizely Feature Experimentation
- Styled Components
- React Router

Designed for:

- ING Bank demo purposes
- A/B testing showcase
- Feature experimentation

- User experience optimization

Created: October 2025

Version: 1.0.0

Appendix

Environment Variables

```
# Required
REACT_APP_OPTIMIZELY_SDK_KEY=your_sdk_key_here
```

NPM Scripts

```
{
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
}
```

Key Dependencies

```
{
  "@optimizely/optimizely-sdk": "^6.1.0",
  "@optimizely/react-sdk": "^3.3.0",
  "react": "^19.2.0",
  "react-dom": "^19.2.0",
  "react-router-dom": "^7.9.4",
  "styled-components": "^6.1.19",
  "typescript": "^4.9.5"
}
```

End of Documentation

For questions or support, please refer to the documentation files in the project repository or contact the development team.