

RewaMart – Smart Shopping & Investment Platform

Full System Documentation (Next.js-ready)

1. Introduction

1.1 System Overview

RewaMart is a mobile-first web platform that combines e-commerce, digital wallets, cashback rewards, and Tanzanian investment products. It streamlines shopping, earning, investing, and referrals in one unified system built for the Tanzanian financial and retail ecosystem.

1.2 Purpose of the System

- Provide an intuitive online shopping marketplace
- Offer rewards through cashback and referrals
- Allow users to invest earnings in real Tanzanian financial instruments
- Manage wallet transactions, withdrawals, donations, and financial education

1.3 Intended Audience

- Developers building or extending the platform
- System architects
- Financial tech teams
- Business owners and investors
- End-user documentation

2. System Features

2.1 Shopping & Marketplace

- Category-based product browsing
- Search by product name, vendor, or brand
- Ratings and reviews
- Real-time vendor updates
- QR Code purchasing
- Cart management & checkout simulation

2.2 Wallet & Rewards

- Cashback on every purchase (4%–20%)
- Referral rewards (5% + 10% first-purchase bonus)
- Transaction history with filtering
- Export/print wallet summary
- Auto-Invest feature (10%, 25%, 50%)

2.3 Investment Options

- UTT AMIS funds
- Government Bonds
- Treasury Bills
- M-Wekeza mobile investments
- Stock & Shares via DSE brokers
- All investments include a 1% platform fee

2.4 Referral Program

- Unique referral code per user
- 5% commission on referral's first purchase
- 10% welcome bonus for new users
- Referral earnings dashboard

2.5 Withdrawals

Supports:

- M-Pesa, Tigo Pesa, Airtel Money, Halotel, TTCL
- 23 Tanzanian banks (RTGS, EFT, SWIFT)
- 2% withdrawal fee

2.6 Donations

- Donate to Tanzanian charities
- Multiple categories: Education, Health, Environment, Children
- Donation history tracking

2.7 Financial Education

- Daily financial tips
- Budgeting guides
- Investment tutorials
- Success stories
- Compound interest calculators

3. System Architecture

3.1 Technology Stack

Layer	Technology
Frontend	Next.js 14, Tailwind CSS, React Hooks
UI	Inter Font, Responsive mobile-first design
State	LocalStorage, Context API (recommended), Zustand/Redux (optional)
Backend (future)	Node.js, Express, PostgreSQL/MongoDB
Deployment	Vercel, Netlify, AWS, GitHub Pages (static mode)

Note: Original app is static HTML/JS but this documentation is upgraded for Next.js architecture.

4. Next.js Project Structure (Recommended)

```
rewamart/
|   app/
|     ├── layout.js
|     ├── page.js          # Dashboard
|     ├── shop/
|     |   └── page.js
|     ├── wallet/
|     |   └── page.js
|     ├── invest/
|     |   └── page.js
|     ├── referral/
|     |   └── page.js
|     ├── profile/
|     |   └── page.js
|     └── api/             # Future API routes
|
|   components/
|     ├── Navbar.jsx
|     ├── BottomNav.jsx
|     ├── ProductCard.jsx
|     ├── Modal.jsx
|     ├── QRScanner.jsx
|     └── Toast.jsx
|
|   lib/
|     ├── wallet.js
|     ├── investments.js
|     ├── referrals.js
|     └── transactions.js
|
|   public/
|     ├── images/
|     |   └── icons/
|     └── styles/
|       └── globals.css
|
|   package.json
```

5. Data Models

5.1 User Model

```
{
  walletBalance: Number,
  totalEarned: Number,
  totalInvested: Number,
  cashbackTotal: Number,
  bonusTotal: Number,
  referralEarnings: Number,
  totalReferrals: Number,
  appliedReferralCode: String,
  autoInvestEnabled: Boolean,
  userName: String,
  userPhone: String,
  userLocation: String,
  impactStats: {
    purchases: Number,
    referrals: Number,
    investments: Number
  }
}
```

5.2 Transaction Model

```
{
  id: Number,
  type: "reward" | "withdrawal" | "investment" | "donation",
  amount: Number,
  description: String,
  date: String,
  time: String
}
```

5.3 Product Model

```
{
  id: String,
  name: String,
  price: Number,
  reward: Number,
  vendor: String,
  image: String,
  rating: Number
}
```

6. System Workflows

6.1 Shopping Flow

1. User browses products
2. Selects product
3. Adds to cart
4. Checks out
5. System simulates payment
6. Cashback = price × reward rate
7. Add reward to wallet
8. Trigger Auto-Invest (if enabled)

6.2 Investment Flow

1. Choose investment type
2. Enter required info
3. Validate minimum amount
4. Calculate +1% fee
5. Deduct from wallet
6. Add investment record

6.3 Referral Flow

1. User gets unique code
2. Shares it
3. Invitee uses code before 1st purchase
4. On invitee purchase:
 - Invitee: 10% bonus
 - Referer: 5% commission

6.4 Withdrawal Flow

1. Enter amount
2. Platform adds 2% fee
3. Validate
4. Process transfer
5. Add withdrawal history

7. Fees Structure

Feature	Fee
Shopping	0%
Investments	1%
Withdrawals	2%
Donations	0%
Referral System	Free (pays bonuses)

8. Supported Investment Institutions

8.1 UTT AMIS Funds

- Equity Fund
- Bond Fund
- Money Market
- Balanced Fund
- Islamic Fund
- Dollar Fund
- Income Fund
- Growth Fund

8.2 Brokers (10+)

CRDB, Vertex, Orbit, Hermes, Sterling, TIB, Mkombozi, Azania, African Alliance, Standard

8.3 DSE Stocks

- CRDB
- NMB
- TBL
- Twiga Cement
- Swissport
- TPCC

9. API Integration (Future)

Wallet Balance API

```
GET /api/wallet/balance
```

Purchase API

```
POST /api/payments/process
```

Protected Routes Header

```
"Authorization": "Bearer <JWT TOKEN>"
```

10. Security Considerations

- Enforce JWT auth
- HTTPS only
- CSRF + Rate limiting
- Backend validation
- Tanzanian finance compliance
- KYC / AML paperwork
- Payment encryption & API safe handling

11. Future Enhancements

- Push notifications
- Portfolio charts
- Group investments
- Loans
- Insurance
- Multi-currency
- PWA + offline mode
- Biometric auth
- AI financial advisor
- Gamification

12. Troubleshooting

Issue	Fix
Cart not updating	Clear local storage or refresh
Modal stuck open	Reload or click close icon
Balance insufficient	Confirm you have enough including fees
Web Share API failing	Use clipboard fallback

13. Contact

- Email: support@rewamart.com
- Phone: +255 XXX XXX XXX
- Location: Dar es Salaam, Tanzania

14. License

Demo license — not for production finance operations.

15. Version History

v2.1.0 (Dec 2024)

- First stable version with full features
- 5 investment types
- Rewards + referrals
- Wallet management
- Donations
- Financial education

16. Detailed Feature Specifications

16.1 Product Categories

Electronics

- Smartphones, Laptops, TVs, Audio Equipment
- Cashback: 5-10%
- Featured Vendors: TechHub TZ, iStore Tanzania, Electronics Plus

Fashion & Clothing

- Footwear, Sportswear, Accessories, Formal Wear
- Cashback: 8-15%
- Featured Vendors: SportZone TZ, Fashion Hub, Executive Wear

Home & Garden

- Furniture, Appliances, Home Decor
- Cashback: 4-7%
- Featured Vendors: Furniture Palace, Home Appliances TZ, ElectroMart

Health & Beauty

- Skincare, Supplements, Fitness Equipment
- Cashback: 15-20% (highest category)
- Featured Vendors: Beauty Corner, FitLife Tanzania, HealthPlus Pharmacy

16.2 Investment Minimum Amounts

Investment Type	Minimum Amount	Term	Expected Return
UTT AMIS	TZS 10,000	Flexible	Varies by fund
Government Bonds	TZS 1,000,000	2-20 years	12.5%-18.2%
Treasury Bills	TZS 100,000	35-364 days	8.5%-11.8%
M-Wekeza	TZS 1,000	Flexible	Market-based
Stocks & Shares	10 shares	Indefinite	Market-based

16.3 Mobile Money Operators

Operator	Prefix Codes	Network
Vodacom M-Pesa	074, 075, 076	Vodacom
Tigo Pesa	071, 065, 067	Tigo
Airtel Money	068, 069, 078	Airtel
Halotel	062, 063	Halotel
TTCL Mobile	073	TTCL

16.4 Supported Banks (23 Total)

- CRDB Bank PLC
- National Microfinance Bank (NMB)
- Stanbic Bank Tanzania
- Standard Chartered Bank
- Exim Bank Tanzania
- NCBA Bank Tanzania
- Azania Bank
- Bank of Baroda
- Bank of India
- Citibank Tanzania
- Diamond Trust Bank (DTB)
- Equity Bank Tanzania
- First National Bank (FNB)
- Guaranty Trust Bank (GTBank)
- I&M Bank Tanzania
- KCB Bank Tanzania
- Mkombozi Commercial Bank
- PBZ Bank
- Peoples Bank of Zanzibar

- Tanzania Commercial Bank (TCB)
- Tanzania Investment Bank (TIB)
- United Bank for Africa (UBA)
- Access Bank Tanzania

17. Component Architecture (Next.js)

17.1 Core Components

Navbar Component

```
// components/Navbar.jsx
export default function Navbar({ walletBalance, userName }) {
  return (
    <header className="gradient-bg text-white p-4 sticky top-0 z-50">
      <div className="max-w-md mx-auto flex items-center justify-between">
        <div className="flex items-center space-x-3">
          <div className="w-10 h-10 bg-white rounded-full flex items-center justify-center">
            <span className="text-purple-600 font-bold text-lg">R</span>
          </div>
        </div>
        <h1 className="font-bold text-lg">RewaMart</h1>
        <p className="text-xs opacity-90">Shop • Earn • Invest</p>
      </div>
      <div className="text-right">
        <p className="text-xs opacity-90">Wallet Balance</p>
        <p className="font-bold text-lg">TZS {walletBalance.toLocaleString()}</p>
      </div>
    </header>
  );
}
```

Product Card Component

```
// components/ProductCard.jsx
export default function ProductCard({ product, onAddToCart, onViewDetails }) {
  return (
    <div className="card-shadow bg-white p-4 rounded-xl border">
      <div className="flex items-center space-x-4">
        <div className="w-16 h-16 bg-gradient-to-br from-purple-400 to-pink-400 rounded-xl flex items-center justify-center text-2xl">
          {product.image}
        </div>
        <div className="flex-1">
          <h4 className="font-medium text-gray-800">{product.name}</h4>
          <p className="text-sm text-gray-500">{product.vendor}</p>
          <div className="flex items-center space-x-2 mt-1">
            <span className="text-lg font-bold text-purple-600">
              TZS {product.price.toLocaleString()}
            </span>
            <span className="text-xs text-green-600">
              +(Math.round(product.reward * 100))%
            </span>
          </div>
        </div>
        <div className="flex flex-col space-y-2">
          <button
            onClick={() => onViewDetails(product.id)}
            className="bg-purple-600 hover:bg-purple-700 text-white px-3 py-2 rounded-lg text-xs font-medium transition-colors">
            View
          </button>
          <button
            onClick={() => onAddToCart(product.id)}
            className="bg-green-600 hover:bg-green-700 text-white px-3 py-2 rounded-lg text-xs font-medium transition-colors">
            Add
          </button>
        </div>
      </div>
    </div>
  );
}
```

17.2 State Management Utilities

Wallet Management

```
// lib/wallet.js
export class WalletManager {
  constructor() {
    this.balance = 0;
    this.transactions = [];
  }

  addFunds(amount, description, type = 'reward') {
    this.balance += amount;
    this.addTransaction(type, amount, description);
  }

  deductFunds(amount, description, type = 'withdrawal') {
    if (this.balance >= amount) {
      this.balance -= amount;
      this.addTransaction(type, amount, description);
      return true;
    }
    return false;
  }

  addTransaction(type, amount, description) {
    this.transactions.unshift({
      id: Date.now(),
      type,
      amount,
      description,
      date: new Date().toLocaleDateString(),
      time: new Date().toLocaleTimeString()
    });
  }

  getBalance() {
    return this.balance;
  }

  getTransactions(filterType = 'all') {
    if (filterType === 'all') return this.transactions;
    return this.transactions.filter(t => t.type === filterType);
  }
}
```

Investment Manager

```
// lib/investments.js
export const INVESTMENT_TYPES = {
  UTT: {
    name: 'UTT AMIS',
    minAmount: 10000,
    fee: 0.01,
    funds: [
      'UTT Equity Fund',
      'UTT Bond Fund',
      'UTT Money Market Fund',
      'UTT Balanced Fund',
      'UTT Islamic Fund',
      'UTT Dollar Fund',
      'UTT Growth Fund',
      'UTT Income Fund'
    ]
  },
  BONDS: {
    name: 'Government Bonds',
    minAmount: 1000000,
    fee: 0.01,
    options: [
      { name: '2-Year Treasury Bond', years: 2, rate: '12.5%' },
      { name: '5-Year Treasury Bond', years: 5, rate: '14.2%' },
      { name: '7-Year Treasury Bond', years: 7, rate: '15.1%' },
      { name: '10-Year Treasury Bond', years: 10, rate: '16.8%' },
      { name: '15-Year Treasury Bond', years: 15, rate: '17.5%' },
      { name: '20-Year Treasury Bond', years: 20, rate: '18.2%' }
    ]
  },
  TBILLS: {
    name: 'Treasury Bills',
    minAmount: 100000,
    fee: 0.01,
    options: [
      { name: '35-Day Treasury Bill', days: 35, rate: '8.5%' },
      { name: '91-Day Treasury Bill', days: 91, rate: '9.2%' },
      { name: '182-Day Treasury Bill', days: 182, rate: '10.1%' },
      { name: '364-Day Treasury Bill', days: 364, rate: '11.8%' }
    ]
  },
  MWEKEZA: {
    name: 'M-Wekeza',
    minAmount: 1000,
    fee: 0.01
  },
  STOCKS: {
    name: 'Stocks & Shares',
    minShares: 10,
    fee: 0.01,
    companies: {
      crdb: { name: 'CRDB Bank', price: 1200 },
      nmb: { name: 'NMB Bank', price: 2800 },
      tbl: { name: 'Tanzania Breweries', price: 4500 },
      twiga: { name: 'Twiga Cement', price: 850 },
      tpc: { name: 'Tanzania Portland Cement', price: 1650 },
      swissport: { name: 'Swissport Tanzania', price: 950 }
    }
  }
};

export function calculateInvestmentCost(type, amount) {
  const investment = INVESTMENT_TYPES[type];
  const fee = Math.floor(amount * investment.fee);
  return {
    amount,
    fee,
    total: amount + fee
  };
}

export function validateInvestment(type, amount) {
  const investment = INVESTMENT_TYPES[type];
  if (!investment) return { valid: false, error: 'Invalid investment type' };

  if (amount < investment.minAmount) {
    return {
      valid: false,
      error: `Minimum investment is TZS ${investment.minAmount.toLocaleString()}`
    };
  }

  return { valid: true };
}
```

18. Deployment Guide

18.1 Vercel Deployment (Recommended)

```
# Install Vercel CLI
npm i -g vercel

# Login to Vercel
vercel login

# Deploy
vercel --prod
```

18.2 Environment Variables

```
# .env.local
NEXT_PUBLIC_API_URL=https://api.rewamart.com
NEXT_PUBLIC_MPESA_API_KEY=your_mpesa_key
DATABASE_URL=postgresql://user:password@host:5432/rewamart
JWT_SECRET=your_jwt_secret
```

18.3 Build Configuration

```
// next.config.js
module.exports = {
  reactStrictMode: true,
  images: {
    domains: ['rewamart.com'],
  },
  env: {
    CUSTOM_API_URL: process.env.NEXT_PUBLIC_API_URL,
  },
}
```

19. Testing Strategy

19.1 Unit Tests

```
// __tests__/wallet.test.js
import { WalletManager } from '../lib/wallet';

describe('WalletManager', () => {
  let wallet;

  beforeEach(() => {
    wallet = new WalletManager();
  });

  test('adds funds correctly', () => {
    wallet.addFunds(1000, 'Test deposit');
    expect(wallet.getBalance()).toBe(1000);
  });

  test('deducts funds when sufficient balance', () => {
    wallet.addFunds(1000, 'Deposit');
    const result = wallet.deductFunds(500, 'Withdrawal');
    expect(result).toBe(true);
    expect(wallet.getBalance()).toBe(500);
  });

  test('fails to deduct when insufficient balance', () => {
    wallet.addFunds(100, 'Deposit');
    const result = wallet.deductFunds(500, 'Withdrawal');
    expect(result).toBe(false);
    expect(wallet.getBalance()).toBe(100);
  });
});
```

19.2 Integration Tests

```
// __tests__/purchase-flow.test.js
import { render, screen, fireEvent } from '@testing-library/react';
import PurchasePage from '../app/purchase/page';

describe('Purchase Flow', () => {
  test('completes purchase and adds cashback', async () => {
    render();
    // Select product
    const product = screen.getByText('Samsung Galaxy S24');
    fireEvent.click(product);

    // Add to cart
    const addButton = screen.getByText('Add to Cart');
    fireEvent.click(addButton);

    // Checkout
    const checkoutButton = screen.getByText('Checkout');
    fireEvent.click(checkoutButton);

    // Verify cashback added
    await screen.findByText(/TZS .* added to your wallet/);
  });
});
```

20. Performance Optimization

20.1 Code Splitting

```
// app/layout.js
import dynamic from 'next/dynamic';

const DynamicModal = dynamic(() => import('@/components/Modal'), {
  loading: () => <p>Loading...</p>,
  ssr: false
});
```

20.2 Image Optimization

```
import Image from 'next/image';

<Image
  src="/products/phone.jpg"
  alt="Product"
  width={200}
  height={200}
  priority
/>
```

20.3 Caching Strategy

```
// app/api/products/route.js
export const revalidate = 3600; // Revalidate every hour

export async function GET() {
  const products = await fetchProducts();
  return Response.json(products);
}
```

21. Accessibility (WCAG 2.1 AA)

- Semantic HTML elements
- ARIA labels on interactive elements
- Keyboard navigation support
- Color contrast ratio ≥ 4.5:1
- Focus indicators
- Screen reader compatibility
- Alt text for all images

22. Analytics & Monitoring

22.1 Google Analytics Setup

```
// app/layout.js
import Script from 'next/script';

export default function RootLayout({ children }) {
  return (
    <html>
      <head>
        <Script
          src="https://www.googletagmanager.com/gtag/js?id=GA_MEASUREMENT_ID"
          strategy="afterInteractive"
        />
      </head>
      <body>{children}</body>
    </html>
  );
}
```

22.2 Error Tracking (Sentry)

```
// sentry.client.config.js
import * as Sentry from '@sentry/nextjs';

Sentry.init({
  dsn: process.env.NEXT_PUBLIC_SENTRY_DSN,
  tracesSampleRate: 1.0,
});
```

23. Backup & Recovery

- Daily database backups
- Transaction logs retention: 7 years
- User data encrypted at rest
- Disaster recovery plan with RTO < 4 hours
- Multi-region redundancy

24. Compliance & Regulations

- Tanzania Communications Regulatory Authority (TCRA) compliance
- Bank of Tanzania (BoT) financial service guidelines
- Data Protection Act adherence
- AML/CFT regulations compliance
- Consumer Protection Act compliance

25. Support & Maintenance

25.1 Support Channels

- Email: support@rewamart.com (24/7)
- Phone: +255 XXX XXX XXX (9 AM - 6 PM EAT)
- Live Chat: Available in app
- FAQ: In-app help section

25.2 SLA Commitments

Issue Severity	Response Time	Resolution Time
Critical	< 1 hour	< 4 hours
High	< 4 hours	< 24 hours
Medium	< 24 hours	< 72 hours
Low	< 48 hours	< 1 week

End of Documentation

Built with ❤️ for Tanzania

For technical inquiries, contact: dev@rewamart.com

Last Updated: December 2024