

Pages & Routing

Bilko Frontend Pages

Framework: Next.js 15 (App Router) **Current State:** Fully implemented with mock data **Future State:** Will require API integration to replace mock data

Route Architecture Overview

```
graph TD
  ROOT["app/layout.tsx\n(Root HTML, Inter font, metadata)"]
  LANDING["/ - Landing Page\napp/page.tsx"]
  AUTH["(auth) group\nNo shared layout"]
  DASH["(dashboard) group\napp/(dashboard)/layout.tsx\nSidebar + TopBar"]
  LOGIN["/login\nLoginPage"]
  REGISTER["/register\nRegisterPage"]
  DASHBOARD["/dashboard\nDashboard Home"]
  INVOICES["/invoices\nInvoice List"]
  INV_NEW["/invoices/new\nInvoice Wizard"]
  EXPENSES["/expenses\nExpense List"]
  EXPENSES_NEW["/expenses/new\nAdd Expense"]
  PURCHASES["/purchases\nPurchases (alias → expenses)"]
  BANKING["/banking\nBanking Hub"]
  REPORTS["/reports\nReports Hub"]
  REPORTS_PL["/reports/profit-loss\nP&L Report"]
  REPORTS_VAT["/reports/vat\nVAT Report"]
  SETTINGS["/settings\nSettings"]
  LOGIN --> LANDING
  REGISTER --> LANDING
  LOGIN --> DASHBOARD
  REGISTER --> DASHBOARD
```

```
REGISTER --> DASHBOARD
```

```
classDef layout fill:#1e1e2e,color:#cdd6f4,stroke:#89b4fa
classDef auth fill:#313244,color:#cba6f7,stroke:#cba6f7
classDef dashboard fill:#1e1e2e,color:#a6e3a1,stroke:#a6e3a1
classDef reports fill:#313244,color:#89dceb,stroke:#89dceb
class ROOT,DASH layout
class AUTH,LOGIN,REGISTER auth
class DASHBOARD,INVOICES,INV_NEW,EXPENSES,EXPENSES_NEW,PURCHASES,BANKING,SETTINGS
dashboard
class REPORTS,REPORTS_PL,REPORTS_VAT reports
```

Layouts

Root Layout

- **Route:** `/`
- **File:** `app/layout.tsx`
- **Purpose:** Root HTML wrapper, applies Inter font, sets metadata
- **Metadata:**
 - Title: "Bilko - Accounting Made Simple"
 - Description: "Modern accounting software for Balkan businesses"
- **Dependencies:** Inter font from Google Fonts
- **Current State:** Fully implemented

Dashboard Layout

- **Route:** `/dashboard/*`
- **File:** `app/dashboard/layout.tsx`
- **Purpose:** Layout wrapper for all authenticated pages
- **Key Components:**
 - Sidebar (desktop persistent, mobile overlay)
 - TopBar (header with search, notifications, user menu)
- **State Management:** Local state for mobile sidebar toggle
- **Mobile Responsiveness:** Responsive sidebar with overlay
- **Current State:** Fully implemented

```
graph LR
  DL["DashboardLayout\napp/dashboard/layout.tsx"]
```

```
DL --> SB["Sidebar\ncomponents/sidebar.tsx\n• Logo\n• Main nav (5 items)\n• Bottom nav (Settings)\n• Active state via usePathname"]
```

```
DL --> TB["TopBar\ncomponents/top-bar.tsx\n• Mobile menu button\n• Search (Cmd+K)\n• Notifications bell\n• User dropdown menu"]
```

```
DL --> SLOT["Page Slot\n{children}"]
```

```
SB -->|"dark sidebar #111113"| SCREEN["Rendered Screen"]
```

```
TB -->|"white top bar"| SCREEN
```

```
SLOT -->|"light content #FAFAFA"| SCREEN
```

Pages

Dashboard (Home)

- **Route:** `/dashboard`
- **File:** `app/(dashboard)/dashboard/page.tsx`
- **Purpose:** Financial overview and quick actions
- **Key Components:**
 - Metric cards (Cash Balance, Revenue MTD, Unpaid Invoices)
 - P&L Bar Chart (6-month trend)
 - Receivables Aging (stacked bar chart)
 - Expenses by Category (donut chart)
 - Recent Transactions table
 - Quick Actions card
- **Data Requirements (Future API):**
 - `GET /api/metrics` — cashBalance, revenueMTD, unpaidInvoices, expensesMTD, profitMTD, cashFlowChange
 - `GET /api/pl/monthly` — monthly P&L data (revenue, expenses, profit)
 - `GET /api/receivables/aging` — receivables breakdown (current, 30d, 60d, 90d+)
 - `GET /api/expenses/by-category` — expenses grouped by category
 - `GET /api/transactions/recent?limit=5` — recent transactions
- **Charts:** Recharts (BarChart, PieChart) with responsive containers
- **Current State:** Mock data from `lib/mock-data.ts`
- **Mobile Responsive:** Grid adapts 1/2/3 columns based on breakpoint

```
graph TD
```

```
DP["Dashboard Page\n/dashboard"]
```

```

DP --> ROW1["Row 1 – Metric Cards\ngrid-cols-1 md:grid-cols-3"]
DP --> ROW2["Row 2 – Charts\ngrid-cols-1 md:grid-cols-3"]
DP --> ROW3["Row 3 – Tables + Actions\ngrid-cols-1 lg:grid-cols-3"]

ROW1 --> MC1["Cash Balance\nwith trend arrow"]
ROW1 --> MC2["Revenue MTD\ncurrent month total"]
ROW1 --> MC3["Unpaid Invoices\nwith warning badge"]

ROW2 --> CH1["P&L Bar Chart\nRecharts BarChart\n6-month revenue/expenses/profit"]
ROW2 --> CH2["Receivables Aging\nRecharts StackedBarChart\ncurrent/30d/60d/90d+"]
ROW2 --> CH3["Expenses by Category\nRecharts PieChart (donut)\nper-category breakdown"]

ROW3 --> TBL["Recent Transactions\nRecharts Table\nlast 5 transactions"]
ROW3 --> QA["Quick Actions\nNew Invoice\nNew Expense\nImport Bank Statement"]

```

Invoices List

- **Route:** `/invoices`
- **File:** `app/(dashboard)/invoices/page.tsx`
- **Purpose:** Invoice management with search, filter, sort
- **Key Features:**
 - Status filter (all/draft/sent/paid/overdue)
 - Search by customer name or invoice number
 - Date range filter (this month, last month, quarter, all)
 - Sortable columns (number, customer, date, due date, amount)
 - Row actions (view, edit, send, download PDF, delete)
 - Summary row with totals by status
- **Data Requirements (Future API):**
 - `GET /api/invoices?status=X&search=Y&dateRange=Z&sort=field&dir=asc` — filtered/sorted invoice list
 - `PATCH /api/invoices/:id` — update invoice
 - `DELETE /api/invoices/:id` — delete invoice
 - `POST /api/invoices/:id/send` — send invoice via email
 - `GET /api/invoices/:id/pdf` — generate PDF
- **Current State:** Mock data, client-side filtering/sorting
- **Mobile Responsive:** Table scrolls horizontally, filters stack vertically

Invoice Creation Wizard

- **Route:** `/invoices/new`
- **File:** `app/(dashboard)/invoices/new/page.tsx`
- **Purpose:** 6-step wizard to create and send invoices

- **Steps:**
 1. **Customer Selection** — Select existing customer or add new (dialog)
 2. **Invoice Details** — Invoice number, issue/due dates, net terms, currency
 3. **Line Items** — Add/remove items (description, qty, unit price, VAT rate), auto-calculate totals
 4. **Customization** — Optional notes and payment terms
 5. **Preview** — Visual preview of generated invoice
 6. **Send** — Email form (to, subject, message, copy to self) + save/download options
- **Data Requirements (Future API):**
 - GET /api/customers — customer list for dropdown
 - POST /api/customers — create new customer
 - POST /api/invoices — create invoice (draft)
 - POST /api/invoices/:id/send — send invoice via email
 - GET /api/invoices/:id/pdf — download PDF
- **Form Validation:**
 - Step 1: Customer required
 - Step 3: At least one line item with description required
 - Step 6: Email address required
- **Current State:** Mock data, local state, no persistence
- **Mobile Responsive:** Wizard steps adapt, form fields stack on mobile

Expenses List

- **Route:** /expenses
- **File:** app/(dashboard)/expenses/page.tsx
- **Purpose:** Expense tracking with categorization and approval workflow
- **Key Features:**
 - Period filter (this month, last month, quarter, year)
 - Category filter (Office, Travel, Meals, Utilities, Marketing, Infrastructure, Software, Professional Services)
 - Search by description or vendor
 - Status badges (pending, approved, paid)
 - Receipt attachment indicator
 - Add Expense dialog (with upload placeholder)
 - Summary stats (total, pending, approved, paid counts)
- **Data Requirements (Future API):**
 - GET /api/expenses?period=X&category=Y&search=Z — filtered expense list
 - POST /api/expenses — create expense
 - PATCH /api/expenses/:id — update expense status
 - POST /api/expenses/:id/receipt — upload receipt (multipart)
 - GET /api/expenses/:id/receipt — download receipt
- **Form Fields:**
 - Amount + currency (EUR, RSD, BAM)
 - Category (dropdown)
 - Date

- Vendor (searchable input)
- Payment method (Cash, Card, Bank Transfer)
- Receipt upload (placeholder UI)
- Description (optional)
- **Current State:** Mock data, form submits to console
- **Mobile Responsive:** Filters stack, table scrolls

Purchases (Alias)

- **Route:** `/purchases`
- **File:** `app/(dashboard)/purchases/page.tsx`
- **Purpose:** Alias to expenses page
- **Current State:** Same component as `/expenses`

Banking

- **Route:** `/banking`
- **File:** `app/(dashboard)/banking/page.tsx`
- **Purpose:** Bank account management and reconciliation
- **Key Features:**
 - 3 tabs: Accounts, Reconcile, Transactions
 - **Accounts Tab:** List of bank accounts with balances (multiple currencies)
 - **Reconcile Tab:** Unreconciled transactions with match confidence, period selector
 - **Transactions Tab:** All bank transactions (chronological)
 - Import Transactions button (placeholder)
 - Match actions (approve, link, create new)
- **Data Requirements (Future API):**
 - `GET /api/bank-accounts` — list accounts
 - `POST /api/bank-accounts` — add account
 - `GET /api/bank-accounts/:id/transactions?reconciled=false` — unreconciled transactions
 - `POST /api/bank-accounts/:id/import` — CSV import
 - `POST /api/bank-accounts/:id/transactions/:txId/reconcile` — mark reconciled
 - `POST /api/bank-accounts/:id/transactions/:txId/link?invoiceId=X` — link to invoice/expense
- **Match Confidence Logic:** Visual indicators for 0%, <50%, 50-89%, 90%+
- **Current State:** Mock data, no reconciliation logic
- **Mobile Responsive:** Tabs stack, tables scroll

Reports Hub

- **Route:** `/reports`
- **File:** `app/(dashboard)/reports/page.tsx`
- **Purpose:** Financial report selection and P&L preview

- **Report Cards:**

- Profit & Loss (implemented at `/reports/profit-loss`)
- Balance Sheet (coming soon)
- Cash Flow Statement (coming soon)
- VAT/PDV Report (live at `/reports/vat`)
- Trial Balance (coming soon)
- General Ledger (coming soon)

- **P&L Preview:**

- Expandable Revenue/Expenses sections
- Current month detailed breakdown
- Export buttons (PDF, Excel) — placeholders

- **Data Requirements (Future API):**

- `GET /api/reports/pl?month=YYYY-MM` — P&L report data
- `GET /api/reports/balance-sheet?date=YYYY-MM-DD` — balance sheet
- `GET /api/reports/cash-flow?period=X` — cash flow statement

- **Current State:** Only P&L and VAT reports implemented, others show "Coming Soon" badge

- **Mobile Responsive:** Report cards grid adapts

```
graph TD
    RH["Reports Hub\n/reports"]

    RH --> PL["Profit & Loss\n/reports/profit-loss\nLIVE – expandable  
revenue/expenses\nfetches from API with mock fallback"]
    RH --> BS["Balance Sheet\nCOMING SOON"]
    RH --> CF["Cash Flow Statement\nCOMING SOON"]
    RH --> VAT["VAT/PDV Report\n/reports/vat\nLIVE – 3-step wizard"]
    RH --> TB["Trial Balance\nCOMING SOON"]
    RH --> GL["General Ledger\nCOMING SOON"]

    PL --> PL_R["Revenue Section\n(expandable, breakdown by category)"]
    PL --> PL_E["Expenses Section\n(expandable, breakdown by category)"]
    PL --> PL_NP["Net Profit + Margin"]
    PL --> PL_EX["Export PDF / Export Excel"]

    VAT --> VAT_S1["Step 1: Reconciliation Check\n(warn if unreconciled transactions)"]
    VAT --> VAT_S2["Step 2: VAT Audit\n(all VAT transactions table)"]
    VAT --> VAT_S3["Step 3: Return Summary\nBox 1 / Box 2 / Box 3"]

    classDef live fill:#1e3a2f,color:#a6e3a1,stroke:#a6e3a1
    classDef soon fill:#2a1e2e,color:#888,stroke:#555
    class PL,VAT live
```

VAT Report

- **Route:** `/reports/vat`
- **File:** `app/(dashboard)/reports/vat/page.tsx`
- **Purpose:** 3-step VAT return preparation
- **Steps:**
 1. **Reconciliation Check** — Warns if unreconciled bank transactions exist
 2. **VAT Audit** — Table of all VAT transactions (invoices + expenses) with net/VAT amounts
 3. **Return Summary** — VAT return boxes (Box 1: collected, Box 2: paid, Box 3: net due)
- **Data Requirements (Future API):**
 - `GET /api/bank-accounts/unreconciled-count` — reconciliation status
 - `GET /api/vat/transactions?period=YYYY-MM` — all VAT transactions
 - `GET /api/vat/return?period=YYYY-MM` — calculated VAT return data
 - `POST /api/vat/submit` — e-filing (Phase 2)
 - `GET /api/vat/export/pdf` — PDF export
 - `GET /api/vat/export/xml` — XML for e-filing
- **Calculations:** Assumes 20% standard VAT rate, converts all currencies to EUR equivalent
- **Current State:** Mock data, no submission, export placeholders
- **Mobile Responsive:** Tabs adapt, tables scroll, boxes stack

Settings

- **Route:** `/settings`
- **File:** `app/(dashboard)/settings/page.tsx`
- **Purpose:** Multi-section settings page
- **Sections:**
 1. **Company** — Company profile (name, legal form, address, tax ID, currency, fiscal year)
 2. **Users** — User management table (name, email, role, status), invite button
 3. **Tax & Compliance** — Country, VAT registration, VAT number/rate, compliance reminders
 4. **Integrations** — Connected integrations (Intesa Bank CSV, Email SMTP), available integrations (Stripe, Fiken, Google Sheets, Slack, DocuSeal)
 5. **Notifications** — Email and in-app notification preferences
 6. **Security** — 2FA, session timeout, password policy, audit log, data export, delete company
- **Data Requirements (Future API):**
 - `GET /api/settings/company` — company data
 - `PATCH /api/settings/company` — update company
 - `GET /api/users` — user list

- `POST /api/users/invite` — invite user
 - `GET /api/settings/tax` — tax settings
 - `PATCH /api/settings/tax` — update tax settings
 - `GET /api/integrations` — integration list
 - `POST /api/integrations/:id/connect` — connect integration
 - `GET /api/settings/notifications` — notification preferences
 - `PATCH /api/settings/notifications` — update preferences
 - `GET /api/security/audit-log` — audit log
 - `POST /api/security/data-export` — request data export
 - `DELETE /api/company` — delete company
- **Current State:** Mock data, form submits to console
 - **Mobile Responsive:** Sidebar nav stacks, forms adapt
-

Authentication Flow

```
sequenceDiagram
```

```
    participant U as User
```

```
    participant AP as AuthProvider
```

```
    participant AS as useAuthStore
```

```
    participant R as Router
```

```
    participant API as Backend API
```

```
    U->>AP: Navigate to /dashboard
```

```
    AP->>AS: checkAuth()
```

```
    AS->>API: GET /api/auth/me (Bearer token)
```

```
    alt API not configured (demo mode)
```

```
        AP-->>U: Render page directly (demoFallback=true)
```

```
    else Token valid
```

```
        API-->>AS: 200 OK – user data
```

```
        AS-->>AP: isAuthenticated=true
```

```
        AP-->>U: Render protected page
```

```
    else Token invalid / no token
```

```
        API-->>AS: 401 Unauthorized
```

```
        AS-->>AP: isAuthenticated=false
```

```
        AP->>R: router.replace('/login')
```

```
        R-->>U: Redirect to /login
```

```
        U->>U: Enter email + password
```

```
        U->>AS: login(email, password)
```

```
AS->>API: POST /api/auth/login
API-->>AS: 200 OK – JWT token
AS->>R: router.push('/dashboard')
R-->>U: Redirect to /dashboard
```

end

Screenshot Descriptions

Dashboard

User sees:

- 3 metric cards at top (cash balance with green arrow, revenue MTD, unpaid invoices with warning badge)
- 3 charts in row below (P&L bar chart, receivables aging stacked bar, expenses donut)
- Bottom row: recent transactions table on left, quick action buttons on right

Invoices List

User sees:

- Header with "Invoices" title and "New Invoice" button
- Filter row (status dropdown, search input, date range dropdown)
- Table with all invoices (sortable columns, status badges, action menu per row)
- Summary bar at bottom showing totals by status

Invoice Wizard

User sees:

- Progress bar with 6 steps at top
- Current step content in card below
- Back/Next buttons at bottom (Send on final step)
- Step 3 shows line items with add/remove, running total calculation
- Step 5 shows formatted invoice preview

Expenses

User sees:

- "Expenses" header with "Add Expense" button
- Filters (period, category, search)
- Table with date, description, category, amount, vendor, status badge, receipt icon
- Summary bar with totals (total €, pending count, approved count, paid count)
- Add dialog: form with amount+currency, category, date, vendor, payment method, receipt upload, description

Banking

User sees:

- 3 tabs: Accounts, Reconcile, Transactions
- Accounts tab: table of bank accounts with type, currency, balance
- Reconcile tab: account selector, period, unreconciled transaction table with match confidence indicators, action buttons (✓ approve, link, create)
- Transactions tab: full transaction history with reconciled status

VAT Report

User sees:

- 3-step tabs at top
- Step 1: Warning card if unreconciled transactions exist, "Reconcile Now" and "Continue" buttons
- Step 2: VAT transaction table (type badge, net amount, VAT rate, VAT amount), summary boxes (collected, paid, net due)
- Step 3: VAT return boxes (Box 1, Box 2, Box 3 highlighted), export buttons (PDF, XML), submit button (disabled, "Coming in Phase 2")

Settings

User sees:

- Sidebar nav on left (6 sections)
- Content area on right showing active section
- Company section: form fields for company data
- Users section: table with invite button
- Integrations section: connected integrations cards, available integrations grid
- Security section: danger zone at bottom (delete company, red border)

Notes

- **All pages use mock data** from `lib/mock-data.ts`
 - **Authentication** implemented via `AuthProvider` + `useAuthStore` — demo mode active when `NEXT_PUBLIC_API_URL` not set
 - **No persistence** — refreshing page loses all changes
 - **Mobile-first design** — all pages tested at mobile/tablet/desktop breakpoints
 - **Dark sidebar + light content area** — consistent layout across all pages
-

Revision #3

Created 2026-02-23 10:48:08 UTC by John

Updated 2026-05-31 20:02:40 UTC by John