Forms & Validation

Bilko Forms Documentation

Current State: Native HTML forms with React state **Validation:** Client-side JavaScript validation (no schema validation yet) **Future State:** Zod schemas for validation, react-hook-form for form management

---

Forms Overview

```
graph TD
    FORMS["Bilko Forms"]

    FORMS --> IW["Invoice Wizard\n6-step multi-page form\n/invoices/new"]
    FORMS --> EF["Expense Form\nDialog (modal)\n/expenses"]
    FORMS --> SF["Settings Forms\n6 sections\n/settings"]
    FORMS --> AUTH["Auth Forms\nLogin + Register\n/login /register"]

    IW --> IW1["Step 1: Customer"]
    IW --> IW2["Step 2: Details"]
    IW --> IW3["Step 3: Line Items"]
    IW --> IW4["Step 4: Customization"]
    IW --> IW5["Step 5: Preview"]
    IW --> IW6["Step 6: Send"]

    EF --> EF1["Amount + Currency"]
    EF --> EF2["Category + Date"]
    EF --> EF3["Vendor + Payment Method"]
    EF --> EF4["Receipt Upload (placeholder)"]
    EF --> EF5["Description (optional)"]

    SF --> SF1["Company Profile"]
    SF --> SF2["Tax & Compliance"]
    SF --> SF3["Notification Preferences"]
```

```
    SF --> SF4["Security Settings"]

    AUTH --> AUTH1["Login\nemail + password"]
    AUTH --> AUTH2["Register\nname, company, email, password, country"]
```

Invoice Creation Wizard (6-Step Multi-Page Form)

Route: `/invoices/new` **File:** `app/(dashboard)/invoices/new/page.tsx`

Wizard State Machine

```
stateDiagram-v2
    [*] --> Step1_Customer : navigate to /invoices/new

    Step1_Customer : Step 1 — Customer Selection
    Step1_Customer : Select existing customer (dropdown)
    Step1_Customer : OR open Add Customer dialog
    Step1_Customer : Validation: customer required

    Step2_Details : Step 2 — Invoice Details
    Step2_Details : invoiceNumber (auto-generated)
    Step2_Details : issueDate, dueDate
    Step2_Details : netTerms (Net 15/30/60 → auto-updates dueDate)
    Step2_Details : currency (EUR/RSD/BAM)

    Step3_LineItems : Step 3 — Line Items
    Step3_LineItems : description (required), qty, unitPrice
    Step3_LineItems : vatRate (0/10/17/20/25%)
    Step3_LineItems : total auto-calculated (read-only)
    Step3_LineItems : Validation: min 1 item with description

    Step4_Customize : Step 4 — Customization
    Step4_Customize : notes (optional textarea)
    Step4_Customize : terms (optional textarea)
    Step4_Customize : No validation required
```

```
    Step5_Preview : Step 5 — Preview
    Step5_Preview : Read-only invoice render
    Step5_Preview : All data from previous steps
    Step5_Preview : No validation

    Step6_Send : Step 6 — Send / Save
    Step6_Send : to (email, pre-filled from customer)
    Step6_Send : subject, message (pre-filled template)
    Step6_Send : sendCopy (checkbox)
    Step6_Send : Save as Draft / Download PDF / Send Invoice

    Step1_Customer --> Step2_Details : Next (customer selected)
    Step1_Customer --> Step1_Customer : Next (no customer) — alert shown

    Step2_Details --> Step3_LineItems : Next
    Step2_Details --> Step1_Customer : Back

    Step3_LineItems --> Step4_Customize : Next (has valid item)
    Step3_LineItems --> Step3_LineItems : Next (no items) — alert shown
    Step3_LineItems --> Step2_Details : Back

    Step4_Customize --> Step5_Preview : Next
    Step4_Customize --> Step3_LineItems : Back

    Step5_Preview --> Step6_Send : Next
    Step5_Preview --> Step4_Customize : Back

    Step6_Send --> [*] : Send Invoice → alert + redirect /invoices
    Step6_Send --> Step5_Preview : Back
    Step6_Send --> [*] : Cancel → confirm dialog → /invoices
```

Form Structure

Step 1: Customer Selection

Fields:

- **Customer** (required)
  - Type: Select (dropdown)

- Options: All customers from contacts (type='customer')
- Can add new customer via dialog
- Validation: Required before proceeding to step 2

Add Customer Dialog:

- **Name** (required)
  - Type: Text
  - Validation: Required
- **Email** (required)
  - Type: Email
  - Validation: Required, valid email format
- **Phone** (optional)
  - Type: Tel
  - Validation: None
- **Tax ID** (optional)
  - Type: Text
  - Validation: None

Validation:

- Alert shown if user tries to proceed without selecting customer
- Form submission triggers inline alert (no schema validation)

Step 2: Invoice Details

Fields:

- **Invoice Number**
  - Type: Text
  - Default: Auto-generated (e.g., "INV-2026-009")
  - Validation: None (can be edited)
- **Issue Date**
  - Type: Date
  - Default: Today's date
  - Validation: None
- **Due Date**
  - Type: Date
  - Default: 30 days from issue date
  - Validation: None
- **Net Terms** (shortcut selector)
  - Type: Select
  - Options: Net 15, Net 30, Net 60
  - Behavior: Auto-calculates due date when selected
  - Validation: None
- **Currency**

- Type: Select
- Options: EUR, RSD, BAM
- Default: EUR
- Validation: None

Behavior:

- Net terms selector auto-updates due date field
- All fields can be manually overridden

Step 3: Line Items

Repeating Fields (Line Items):

Each line item contains:

- **Description** (required)
  - Type: Text
  - Placeholder: "Service or product description"
  - Validation: At least one item must have description
- **Quantity**
  - Type: Number
  - Default: 1
  - Min: 1
  - Validation: Positive number
- **Unit Price**
  - Type: Number
  - Default: 0
  - Min: 0
  - Step: 0.01
  - Validation: Non-negative
- **VAT Rate**
  - Type: Select
  - Options: 0%, 10%, 17%, 20%, 25%
  - Default: 20%
  - Validation: None
- **Total** (calculated, read-only)
  - Type: Text (disabled input)
  - Calculation: `quantity * unitPrice * (1 + vatRate/100)`
  - Display: Formatted currency

Actions:

- **Add Item** button: Appends new empty line item
- **Remove Item** button (X): Removes line item (disabled if only 1 item)

Totals Display (read-only):

- Subtotal (before VAT)
- VAT Total
- Grand Total (with VAT)

Validation:

- Alert shown if user tries to proceed with all empty descriptions
- Form submission requires at least one item with description

---

Step 4: Customization

Fields:

- **Notes** (optional)
  - Type: Textarea
  - Default: "Thank you for your business!"
  - Placeholder: "Add a note for your customer..."
  - Validation: None
- **Terms** (optional)
  - Type: Textarea
  - Default: "Payment due within 30 days."
  - Placeholder: "Payment terms and conditions..."
  - Validation: None

Behavior:

- Both fields are optional
- Default values pre-populated but can be cleared

Step 5: Preview (Read-Only)

No form fields. Displays formatted invoice preview with all data from previous steps.

Preview Elements:

- Invoice title ("INVOICE")
- From/To addresses
- Invoice number, date, due date
- Line items table
- Subtotal, VAT, Total
- Notes (if provided)
- Terms (if provided)

No validation. Step is purely visual review.

Step 6: Send/Save

Email Form:

- **To** (required)
  - Type: Email
  - Default: Pre-filled with customer email
  - Validation: Valid email format (no schema yet)
- **Subject** (required)
  - Type: Text
  - Default: "Invoice {invoiceNumber}"
  - Validation: Required
- **Message** (required)
  - Type: Textarea
  - Default: Pre-filled template
  - Rows: 6
  - Validation: Required
- **Send Me a Copy** (optional)
  - Type: Checkbox
  - Default: Unchecked
  - Validation: None

Action Buttons:

- **Save as Draft** — Alert placeholder (no API)
- **Download PDF** — Alert placeholder (no API)
- **Send Invoice** — Alert + redirect to `/invoices` (no API)

Validation:

- No schema validation
- Form submission triggers alert "Invoice sent!"

Line Item Data Flow

```
flowchart LR
    DESC["description\n(text input)"]
    QTY["quantity\n(number input)"]
    PRICE["unitPrice\n(number input)"]
    VAT["vatRate\n(Select: 0/10/17/20/25%)"]
```

```
    QTY --> CALC["useMemo(totals)\nsubtotal = qty * price\nvatTotal = subtotal * rate\ntotal = subtotal + vatTotal"]
    PRICE --> CALC
    VAT --> CALC

    CALC --> ROW_TOTAL["Row Total\n(read-only display)"]
    CALC --> SUBTOTAL["Subtotal\n(sum of all rows)"]
    CALC --> VAT_TOTAL["VAT Total\n(sum of all VAT)"]
    CALC --> GRAND_TOTAL["Grand Total\n(subtotal + vatTotal)"]

    DESC -->|"required"| VALID["Validation\nAt least 1 item\nwith description"]
```

Form State Management

Local State:

```
const [step, setStep] = useState(1)
const [customer, setCustomer] = useState<Contact | null>(null)
const [showAddCustomer, setShowAddCustomer] = useState(false)
const [invoiceDetails, setInvoiceDetails] = useState<InvoiceDetails>({
  number: "INV-2026-009",
  issueDate: "2026-02-20",
  dueDate: "2026-03-22",
  currency: "EUR"
})
const [lineItems, setLineItems] = useState<LineItem[]>([
  { description: "", quantity: 1, unitPrice: 0, vatRate: 20, total: 0 }
])
const [notes, setNotes] = useState("Thank you for your business!")
const [terms, setTerms] = useState("Payment due within 30 days.")
const [emailData, setEmailData] = useState({
  to: "",
  subject: "",
  message: "",
  sendCopy: false
})
```

No persistence: All state lost on page refresh or navigation away.

No Zod schemas: Validation is inline JavaScript (alert boxes).

Expense Form (Dialog)

Route: `/expenses` **Component:** Dialog triggered by "Add Expense" button

Expense Form Flow

```
flowchart TD
    BTN["'Add Expense' Button\nExpenses Page"]
    BTN --> DLG["Dialog Opens\nisDialogOpen=true"]

    DLG --> AMOUNT["Amount (number, required)"]
    DLG --> CURRENCY["Currency (Select: EUR/RSD/BAM)"]
    DLG --> CATEGORY["Category (Select,
required)\nOffice/Travel/Meals/Utilities\nMarketing/Infrastructure\nSoftware/Professional
Services"]
    DLG --> DATE["Date (date input, required)"]
    DLG --> VENDOR["Vendor (text, optional)"]
    DLG --> PAYMENT["Payment Method (Select, optional)\nCash/Card/Bank Transfer"]
    DLG --> RECEIPT["Receipt Upload\n(placeholder UI — no real upload)"]
    DLG --> DESC["Description (text, optional)"]

    AMOUNT --> SUBMIT["Save Expense button"]
    CATEGORY --> SUBMIT

    SUBMIT -->|"current"| CONSOLE["console.log(formData)\nDialog closes\nForm resets"]
    SUBMIT -->|"future"| API["POST /api/expenses\nRefetch expense list"]

    DLG --> CANCEL["Cancel button\nDialog closes\nForm resets"]
```

Form Fields

- **Amount** (required)
  - Type: Number
  - Placeholder: "0.00"
  - Validation: Required (no schema)
- **Currency** (required)

    - Type: Select
    - Options: EUR, RSD, BAM
    - Default: EUR
    - Width: 24px (narrow select next to amount)
    - Validation: None
- **Category** (required)
    - Type: Select
    - Options: Office, Travel, Meals, Utilities, Marketing, Infrastructure, Software, Professional Services
    - Placeholder: "Select category"
    - Validation: Required
- **Date** (required)
    - Type: Date
    - Default: Today's date
    - Validation: None
- **Vendor** (optional)
    - Type: Text
    - Placeholder: "Search vendor..."
    - Validation: None
    - Note: Not a searchable autocomplete yet — plain text input
- **Payment Method** (optional)
    - Type: Select
    - Options: Cash, Card, Bank Transfer
    - Placeholder: "Select method"
    - Validation: None
- **Receipt** (optional)
    - Type: File upload (placeholder UI only)
    - Display: Dashed border div with "Upload or Drag" text
    - Behavior: No actual upload implemented
    - Validation: None
- **Description** (optional)
    - Type: Text
    - Placeholder: "Additional notes..."
    - Validation: None

Form Actions

- **Cancel** — Closes dialog, resets form
- **Save Expense** — Logs form data to console, closes dialog, resets form

No API submission. Form data not persisted.

No Zod schemas. Validation is JavaScript logic in form submit handler.

Settings Forms

Route: `/settings` **File:** `app/(dashboard)/settings/page.tsx`

Settings Navigation Flow

```
stateDiagram-v2
    [*] --> Company : default section

    Company : Company Profile
    Company : name, legalForm, address, city\npostalCode, country, taxId\nbaseCurrency, fiscalYearStart

    Users : Users & Roles
    Users : User management table\n(name, email, role, status)\nInvite User button

    Tax : Tax & Compliance
    Tax : country (Serbia/BiH/Croatia)\nvatRegistered (checkbox)\nvatNumber (conditional)\nvatRate (conditional)\ncompliance reminder checkboxes

    Integrations : Integrations
    Integrations : Connected: Intesa Bank CSV, Email SMTP\nAvailable: Stripe, Fiken\nGoogle Sheets, Slack, DocuSeal

    Notifications : Notification Preferences
    Notifications : Email: paid, overdue, approved, synced\nIn-App: invoice updates, expense updates\nreconciliation matches

    Security : Security Settings
    Security : Enable 2FA button\nSession Timeout Select\nPassword Policy checkboxes\nAudit Log / Data Export / Delete Company

    Company --> Users : sidebar nav click
    Company --> Tax : sidebar nav click
    Company --> Integrations : sidebar nav click
    Company --> Notifications : sidebar nav click
    Company --> Security : sidebar nav click
```

```
    Users --> Company : sidebar nav click
    Tax --> Company : sidebar nav click
    Integrations --> Company : sidebar nav click
    Notifications --> Company : sidebar nav click
    Security --> Company : sidebar nav click
```

Company Profile Form

Fields:

- **Company Name** (required)
  - Type: Text
  - Default: "SnowIT d.o.o."
- **Legal Form**
  - Type: Select
  - Options: d.o.o., a.d., Preduzetnik
  - Default: "d.o.o."
- **Address**
  - Type: Text
  - Default: "Zmaja od Bosne"
- **City**
  - Type: Text
  - Default: "Sarajevo"
- **Postal Code**
  - Type: Text
  - Default: "71000"
- **Country**
  - Type: Text
  - Default: "BiH"
- **Tax ID / PIB / JIB**
  - Type: Text
  - Default: "4200000000"
- **Base Currency**
  - Type: Select
  - Options: EUR, RSD, BAM
  - Default: "EUR"
- **Fiscal Year Start**
  - Type: Select
  - Options: Jan 1, Apr 1, Jul 1, Oct 1
  - Default: "Jan 1"

Action:

- **Save Changes** — Alert placeholder (no API)

Validation: None (no required fields enforced)

Tax & Compliance Form

Fields:

- **Country**
  - Type: Select
  - Options: Serbia, BiH, Croatia
  - Default: "Serbia"
- **VAT Registered**
  - Type: Checkbox
  - Default: Checked
- **VAT Number** (conditional, shown only if VAT registered)
  - Type: Text
  - Default: "RS123456789"
  - Placeholder: "Enter VAT number"
- **VAT Rate** (conditional, shown only if VAT registered)
  - Type: Select
  - Options: 17% (BiH), 20% (Serbia), 25% (Croatia)
  - Default: "20"

Compliance Reminders:

- **VAT filing deadlines** — Checkbox (default: checked)
- **Annual tax returns** — Checkbox (default: checked)
- **Payroll tax deadlines** — Checkbox (default: unchecked)

Action:

- **Save Settings** — Alert placeholder (no API)

Validation: None

Notification Preferences

Email Notifications:

- Invoice paid — Checkbox (default: checked)
- Invoice overdue — Checkbox (default: checked)
- Expense approved — Checkbox (default: unchecked)
- Bank account synced — Checkbox (default: checked)

In-App Notifications:

- Invoice updates — Checkbox (default: checked)
- Expense updates — Checkbox (default: checked)
- Reconciliation matches — Checkbox (default: unchecked)

Action:

- **Save Preferences** — Alert placeholder (no API)

Validation: None

Security Settings

Two-Factor Authentication:

- **Enable 2FA** button — No functionality yet

Session Timeout:

- Type: Select
- Options: 15 minutes, 30 minutes, 1 hour, 4 hours
- Default: 30 minutes

Password Policy:

- Minimum 12 characters — Checkbox (default: checked)
- Require special characters — Checkbox (default: checked)
- Expire passwords after 90 days — Checkbox (default: unchecked)

Actions:

- **View Audit Log** — No functionality yet
- **Request Data Export** — No functionality yet
- **Delete Company** (Danger Zone) — No functionality yet

Validation: None

Auth Forms

Login Form (`/login`)

```
flowchart TD
    LP["LoginPage\n/login"]
    LP --> EMAIL["Email input\ntype=email, required"]
    LP --> PASS["Password input\ntype=password, show/hide toggle"]
    LP --> SUBMIT["Submit Button\n'Prijavite se'"]

    SUBMIT --> STORE["useAuthStore.login(email, password)"]
    STORE -->|"success"| DASH["router.push('/dashboard')"]
    STORE -->|"error"| ERR["Error banner\n(red bg, border, message from store)"]

    LP --> FORGOT["Forgot password link\n(no functionality yet)"]
    LP --> REG["Link to /register"]
```

Register Form (`/register`)

Fields:

- **First Name** (required)
- **Last Name** (required)
- **Company** (required)
- **Email** (required)
- **Password** (required, show/hide toggle)
- **Country** (required, Select)

Submit: Calls `api.auth.register()` → auto-login via `useAuthStore.login()` → redirect to `/dashboard`

Validation Architecture

```
flowchart LR
    subgraph CURRENT["Current (Phase 1)"]
        INLINE["Inline JS Validation\nalert() on submit\nno schema, no real-time feedback"]
        INLINE --> INVOICE_V["Invoice Wizard\n• Step 1: if(!customer) alert()\n• Step 3:
if(!descriptions) alert()"]
        INLINE --> EXPENSE_V["Expense Form\n• if(!amount || !category) return"]
        INLINE --> AUTH_V["Auth Forms\n• HTML required attribute\n• type=email browser
validation"]
    end
```

```
    subgraph FUTURE["Future (Phase 2)"]
        ZOD["Zod Schemas\ncentralized, type-safe, reusable"]
        RHF["react-hook-form\nuseForm + zodResolver"]
        ZOD --> RHF
        RHF --> RT["Real-time Validation\nfield-level, on-change or on-blur"]
        RHF --> EM["Error Messages\ninline under each field"]
        RHF --> ES["Error State Styling\nred border + error text"]
    end
```

Future Form Enhancements (Phase 2)

Zod Schema Validation

Planned: Replace inline validation with Zod schemas

Example (Invoice Wizard Step 1):

```
import { z } from 'zod'

const customerSchema = z.object({
  id: z.string(),
  name: z.string().min(1, "Customer name required"),
  email: z.string().email("Valid email required"),
  phone: z.string().optional(),
  taxId: z.string().optional()
})
```

Benefits:

- Type-safe validation
- Reusable schemas for API/DB
- Better error messages
- Centralized validation logic

---

react-hook-form Integration

Planned: Replace useState with react-hook-form

Example (Expense Form):

```
import { useForm } from 'react-hook-form'
import { zodResolver } from '@hookform/resolvers/zod'

const expenseSchema = z.object({
  amount: z.number().positive("Amount must be positive"),
  currency: z.enum(['EUR', 'RSD', 'BAM']),
  category: z.string().min(1, "Category required"),
  date: z.string(),
  vendor: z.string().optional(),
  paymentMethod: z.string().optional(),
  description: z.string().optional()
})

const { register, handleSubmit, formState: { errors } } = useForm({
  resolver: zodResolver(expenseSchema)
})
```

Benefits:

- Automatic error handling
- Less boilerplate
- Better performance (no re-renders on every keystroke)
- Built-in dirty/touched state

Field-Level Validation

Planned: Real-time validation as user types

Example (Email Field):

```
<Input
  {...register("email")}
  type="email"
  error={errors.email?.message}
/>
{errors.email && (
  <span className="text-error text-sm">{errors.email.message}</span>
)}
```

Current State: No real-time validation, only on form submit.

Form Persistence

Planned: Save draft forms to localStorage

Use Cases:

- Invoice wizard state saved between page refreshes
- Expense form data saved if user closes dialog accidentally

Implementation:

```
// Save to localStorage on every state change
useEffect(() => {
  localStorage.setItem('invoice-draft', JSON.stringify(invoiceState))
}, [invoiceState])

// Load from localStorage on mount
useEffect(() => {
  const draft = localStorage.getItem('invoice-draft')
  if (draft) setInvoiceState(JSON.parse(draft))
}, [])
```

File Upload (Receipt Attachment)

Current State: Placeholder UI only (dashed border div)

Future Implementation:

- Drag-and-drop file upload
- File size validation (max 5MB)
- File type validation (PDF, JPG, PNG)
- Preview uploaded file
- Remove uploaded file
- Upload to backend API

API Endpoint (planned):

```
POST /api/expenses/:id/receipt
Content-Type: multipart/form-data
```

Autocomplete/Search Fields

Current State: Plain text inputs

Future Enhancement (Vendor Field):

- Searchable dropdown
- Autocomplete from existing vendors
- Create new vendor inline
- Match by partial name

Library: Radix UI Combobox or react-select

Multi-Currency Conversion

Current State: User manually selects currency

Future Enhancement:

- Fetch live exchange rates
- Auto-convert amounts for display
- Store both original currency and base currency
- Show converted amounts in tooltips

Summary

Current Forms:

1. Invoice Wizard (6-step) — Customer, Details, Line Items, Customization, Preview, Send
2. Expense Form (dialog) — Amount, Category, Date, Vendor, Receipt, etc.
3. Company Profile — All company settings
4. Tax & Compliance — VAT settings
5. Notification Preferences — Email/in-app notification toggles
6. Security Settings — 2FA, session timeout, password policy
7. Login Form — Email + password, auth via Zustand store
8. Register Form — Name, company, email, password, country

Validation:

- Inline JavaScript (alert boxes) for wizard and expense form
- HTML `required` attribute + browser validation for auth forms

- No schema validation
- No real-time validation
- No error state styling

State Management:

- React useState for all forms
- No persistence (lost on refresh)
- No form libraries (native HTML forms)
- Auth forms use `useAuthStore` from Zustand (login/register)

Future (Phase 2):

- Zod schemas for validation
- react-hook-form for form management
- Field-level validation
- Form persistence (localStorage)
- File upload functionality
- Autocomplete/search fields
- API integration for submission

Revision #3
Created 2026-02-23 10:48:09 UTC by John
Updated 2026-05-31 20:02:43 UTC by John