**Loan Calculator Application Documentation**

**Project Overview**

**Project Title:** Novel Loan Calculator with Currency Conversion  
**Purpose:** A responsive web application for calculating loan EMIs with amortization schedules and real-time currency conversion

**Github Link**: <https://github.com/sounar97/Novel-Loan-Calculator>

**Deployment link (View)** : <https://sounovel-loan-calculator.netlify.app/>

**Technical Specifications**

**Architecture**

└── src/

├── components/ # Reusable UI components

├── context/ # Global state management

├── hooks/ # Custom business logic

├── pages/ # Application views

└── utils/ # Constants and helpers

**Technology Stack**

| **Category** | **Technologies Used** |
| --- | --- |
| Core Framework | React |
| UI Library | Material-UI |
| State Management | Context API |
| API Handling | Axios |
| Hosting | Netlify |

**Implementation Details**

*1. EMI Calculation System*

**Key Components:**

* useEmiCalculator.js custom hook
* Amortization table generator

**Formula Implementation:**

// In useEmiCalculator.js

const calculateEmi = (principal, annualRate, months) => {

const monthlyRate = annualRate / 12 / 100;

const emi = principal \* monthlyRate \*

Math.pow(1 + monthlyRate, months) /

(Math.pow(1 + monthlyRate, months) - 1);

return emi.toFixed(2);

};

*2. Currency Conversion Module*

**API Integration:**

// In useExchangeRate.js

const fetchRates = async (baseCurrency) => {

try {

const response = await axios.get(

`https://v6.exchangerate-api.com/v6/${API\_KEY}/latest/${baseCurrency}`

);

setRates(response.data.conversion\_rates);

} catch (err) {

setError(err.message);

} finally {

setLoading(false);

}

};

*3. Responsive Design Implementation*

**Breakpoint Handling:**

// In Navbar.js

const theme = useTheme();

const isMobile = useMediaQuery(theme.breakpoints.down('md'));

{isMobile ? (

<MobileMenu />

) : (

<DesktopNavigation />

)}

**Features Delivered**

Core Functionality

| **Feature** | **Implementation Status** | **Key Components Used** |
| --- | --- | --- |
| EMI Calculation | ✅ Completed | useEmiCalculator |
| Amortization Schedule | ✅ Completed | AmortizationTable |
| Currency Conversion | ✅ Completed | useExchangeRate |
| Theme Toggling | ✅ Completed | AppContext |

**Error Handling**

* API error boundaries
* 404 page for invalid routes
* Network error fallbacks

// Error handling example

{error && (

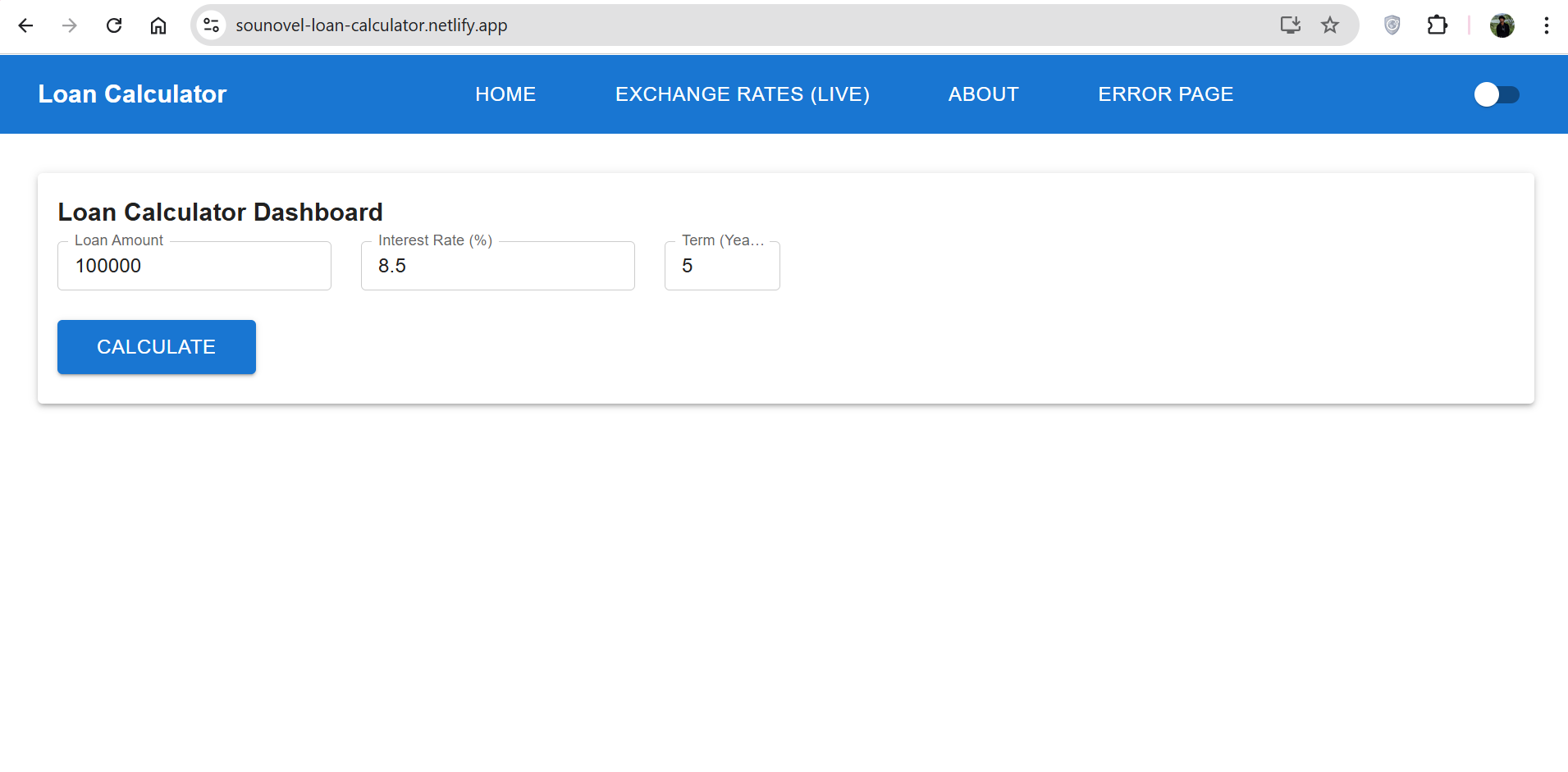
<Alert severity="error" sx={{ mb: 3 }}>

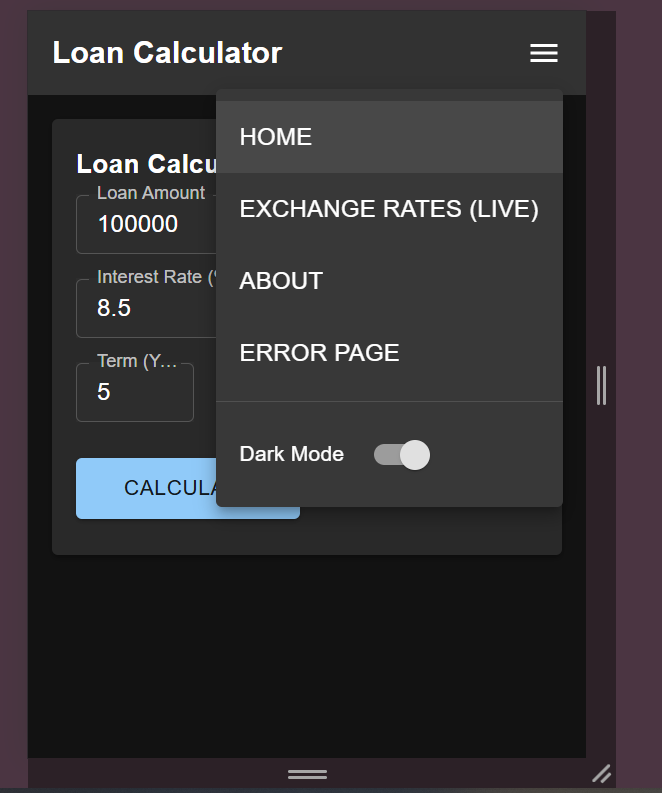
Failed to load rates: {error}

</Alert>

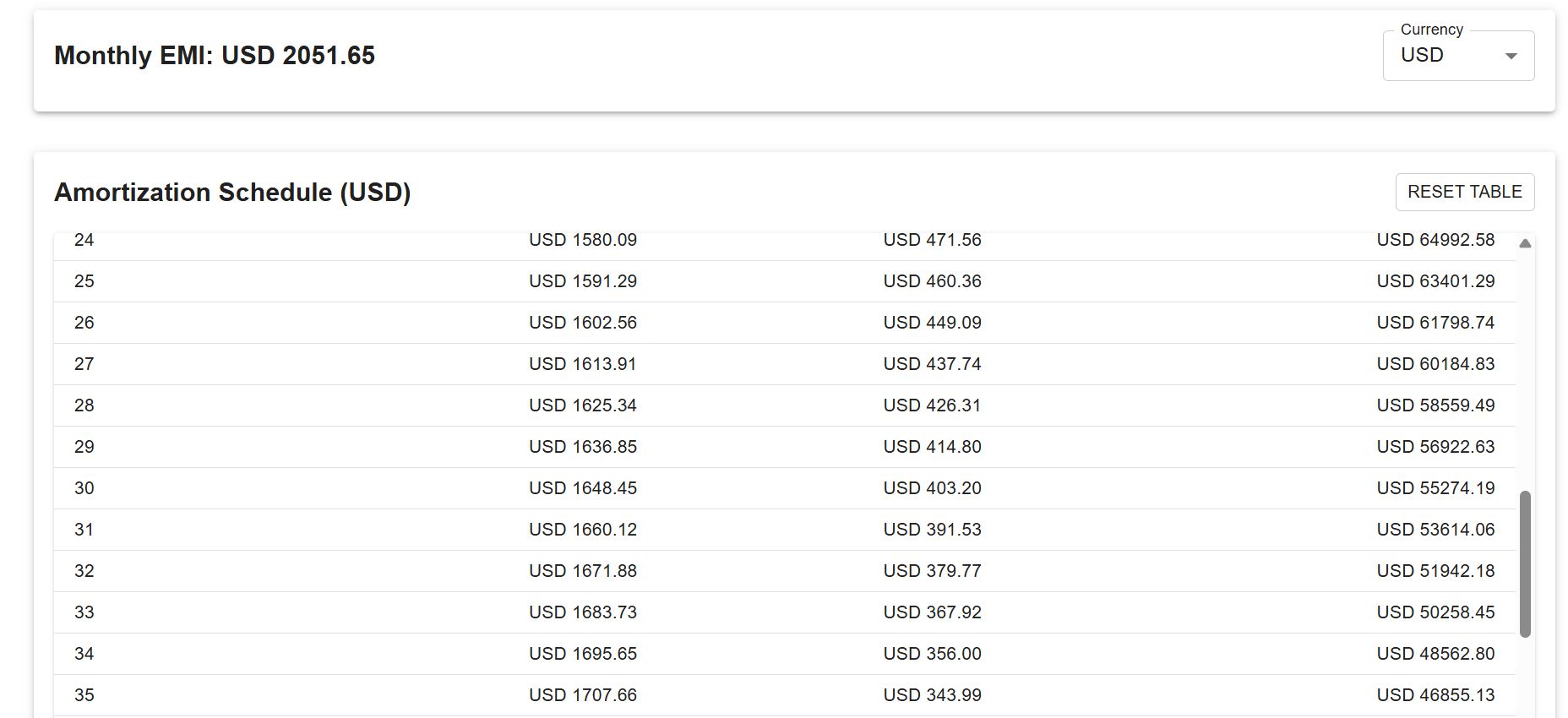
)}

**Screenshots**

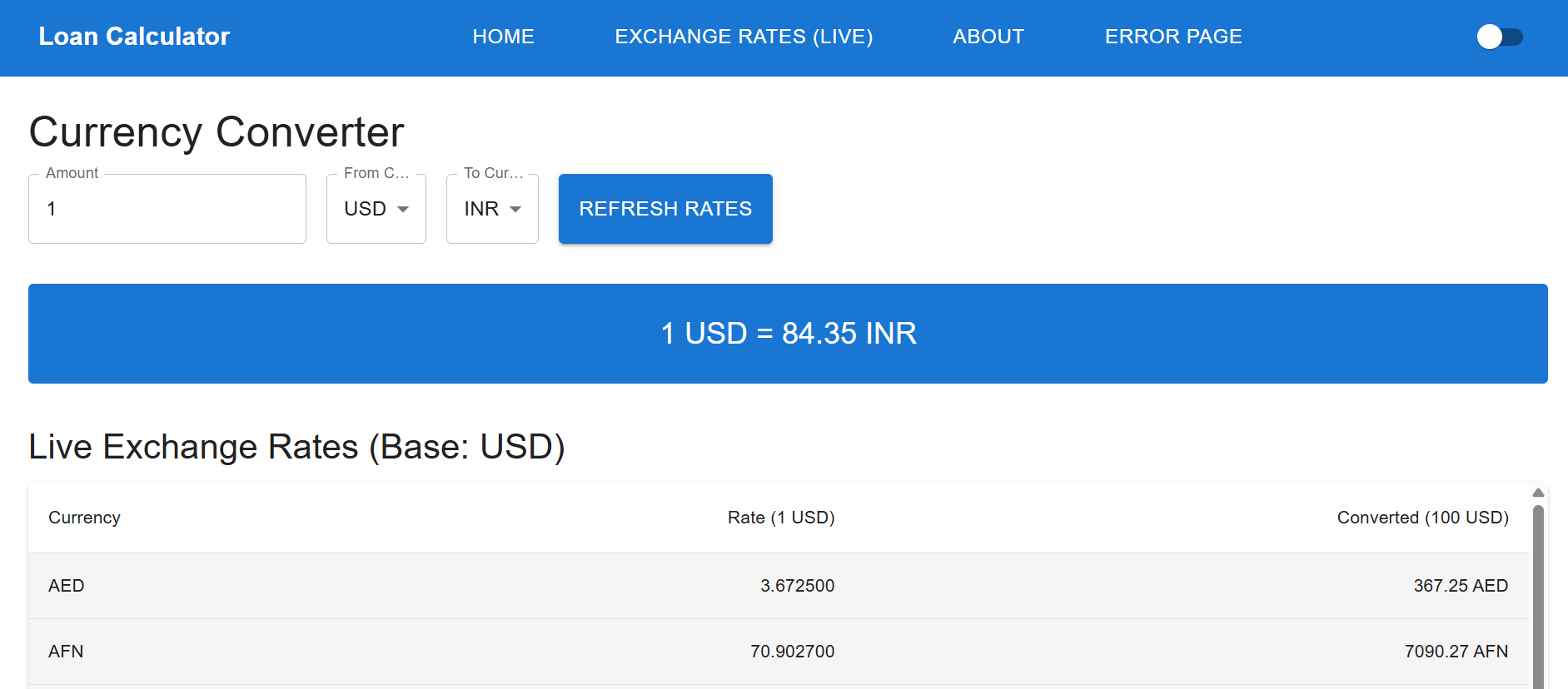
**Desktop view - Light mode**



**Mobile view - Dark mode**



**Amortization table**



**Currency converter**

**Testing Methodology**

1. **Calculation Validation:**
   * Verified EMI results against bank calculators
   * Edge cases: Zero interest, 1-month term
2. **Responsiveness Testing:**
   * Chrome DevTools device emulation
   * Physical devices: Mobile, tablet, desktop
3. **API Testing:**
   * Success scenarios
   * Network failure simulation
   * Invalid API key test

