CryptoLock - GPG Encryption UI

A super cool, modern, and eye-catching React frontend for GPG encryption/decryption operations. Built with stunning animations, glassmorphism effects, and beautiful gradient designs.

Features

- **Modern UI Design** - Beautiful gradient backgrounds with glassmorphism effects
- **Smooth Animations** - Framer Motion animations for seamless transitions
- **User Management** - Create users and generate cryptographic keys
- **Message Encryption** - Encrypt messages with recipient's public key
- **Message Decryption** - Decrypt messages using private keys (client-side only)
- **Responsive Design** - Works perfectly on desktop and mobile devices
- **Real-time Notifications** - Toast notifications for all operations
- **Beautiful Gradients** - Cyan, purple, and pink gradient color scheme
- **Icon Integration** - Lucide icons for beautiful UI elements

Quick Start

Prerequisites

- Node.js 16+ installed
- npm or yarn package manager
- Backend API running on http://127.0.0.1:8000

Installation

1. Navigate to the frontend directory:

```
cd frontend/webapp
```

2. Install dependencies:

```
npm install
```

3. Start the development server:

```
npm run dev
```

4. Open your browser and visit:

```
http://localhost:5173
```

Dependencies

```
{
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "axios": "^1.6.2",
  "framer-motion": "^10.16.4",
  "lucide-react": "^0.292.0",
  "react-toastify": "^9.1.3",
  "tailwindcss": "^3.3.5"
}
```

Design Features

Color Scheme

- **Primary**: Cyan (#06b6d4)
- **Secondary**: Purple (#8b5cf6)
- **Accent**: Pink (#ec4899)
- **Background**: Dark gradient with animated transitions

Effects

- **Glassmorphism**: Frosted glass effect on cards
- **Gradients**: Beautiful linear gradients on buttons and text
- **Animations**: Smooth fade-in, slide-in, and hover animations
- **Glow Effects**: Cyan and purple glow on hover

Pages

1. Home

- Welcome screen with feature showcase
- Call-to-action button to get started

2. Users

- Create new users
- Generate and download cryptographic key pairs
- Beautiful card-based interface

3. Encrypt

- Select recipient username

- Enter message to encrypt
- Copy encrypted message to clipboard
- Real-time encryption with visual feedback

4. Decrypt

- Upload private key file
- Enter passphrase if protected
- Paste ciphertext
- View decrypted message
- Copy plaintext to clipboard

API Integration

The frontend communicates with the backend API at `http://127.0.0.1:8000`:

- `POST /users/` - Create user
- `POST /users/{username}/generate_keys` - Generate and download keys
- `POST /encrypt` - Encrypt message
- `POST /decrypt` - Decrypt message
- `GET /users/{username}/public_key` - Get public key

Usage Workflow

1. **Create User**: Go to Users page → Create New User
2. **Generate Keys**: Enter username → Click "Generate & Download"
3. **Send Message**:
    - Go to Encrypt page
    - Enter recipient username
    - Type message
    - Click "Encrypt Message"
    - Copy ciphertext
4. **Receive Message**:
    - Go to Decrypt page
    - Upload your private key file
    - Enter passphrase if needed
    - Paste ciphertext
    - Click "Decrypt Message"

Development

Build for Production

```
npm run build
```

Preview Production Build

```
npm run preview
```

Code Structure

```
src/
├── App.jsx          # Main component with all pages
├── index.css        # Global styles and animations
├── main.jsx         # Entry point
└── assets/          # Static assets
```

Technologies Used

- **React 18** - UI library
- **Vite** - Build tool & dev server
- **Framer Motion** - Animation library
- **Lucide React** - Icon library
- **Axios** - HTTP client
- **React Toastify** - Notification system
- **Tailwind CSS** - Utility-first CSS framework

Security Notes

- Private keys are **never** sent to the server
- Private keys are uploaded locally from user's file system
- All encryption/decryption happens via the backend's secure API
- Passphrase protection for private keys is optional
- Temporary directories are cleaned up after operations

Tips

- **Save Private Keys**: Always backup your downloaded private key files
- **Remember Passphrases**: Private keys are passphrase-protected for security
- **Share Public Keys**: Your public key can be shared freely to receive encrypted messages
- **Copy Encrypted Messages**: Use the copy button to easily share encrypted messages

Troubleshooting

API Connection Error

- Ensure backend is running on `http://127.0.0.1:8000`
- Check CORS settings on backend

Decryption Failed

- Verify the private key matches the encrypted message

- Check if passphrase is correct
- Ensure ciphertext is complete and properly formatted

File Upload Issues

- Ensure private key file is in `.asc` or `.pgp` format
- Check file size (should be < 10MB)
- Try uploading again

CryptoLock - Making encryption beautiful and accessible!