



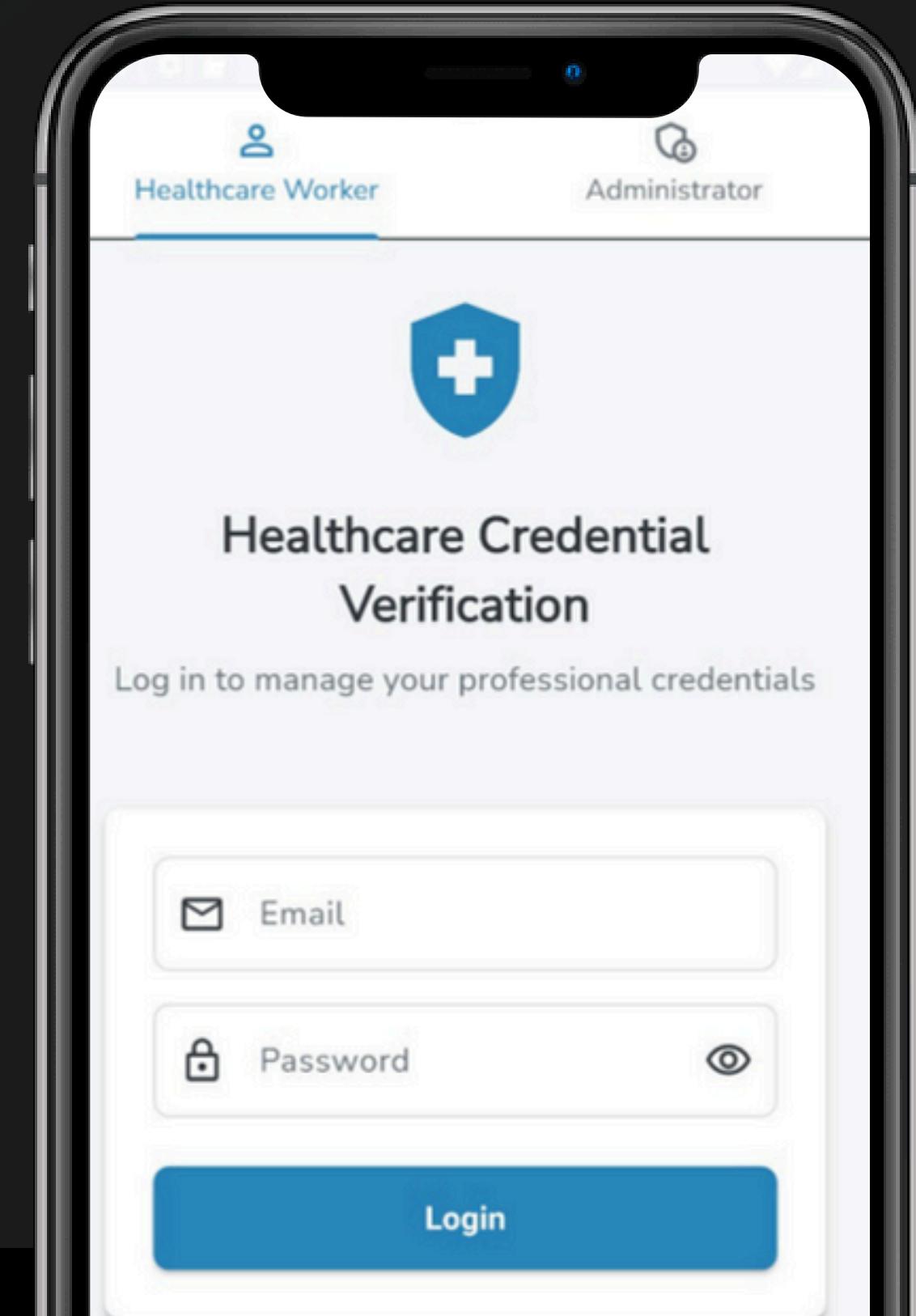
Healthcare Credential Verification

Senior Project Presentation

Project Advisor: Nazeeruddin Mohammed



Lamyia Bakhurji 202001976 - IT
Norah Alsubaie 202001750 - IT
Norah Alghubari 202001373 - IT
Zaynab O. Taha 201802184 - IT



...



Table of Contents

TOPICS COVERED

- 01 • Introduction
- 02 • Objectives
- 03 • Blockchain (overview)
- 04 • System overview
- 05 • Conceptual Design
- 06 • Functional Requirements
- 07 • Non-Functional Requirements
- 08 • UI Screenshots
- 09 • Challenges & Solutions
- 10 • Benefits of the applications
- 11 • Conclusion



Introduction

Healthcare credentialing verifies healthcare providers' qualifications to ensure they meet the standards for safe, effective care.

The growing concerns of identity theft and fake qualifications jeopardize patient safety and trust. Traditional verification methods are error-prone and time-consuming.

The Need:

A secure, transparent, and efficient system is essential to restore trust and reduce risks. Our solution advances digital credential verification to meet today's security and efficiency needs.

2. Objectives

1. Secure Mobile Platform

Create a secure mobile platform for healthcare workers and administrators.

2. Credential Upload & Review

Create a feature for credential uploads and reviewing credentials, such as certificates and licenses.

3. Notifications & Activity Tracking

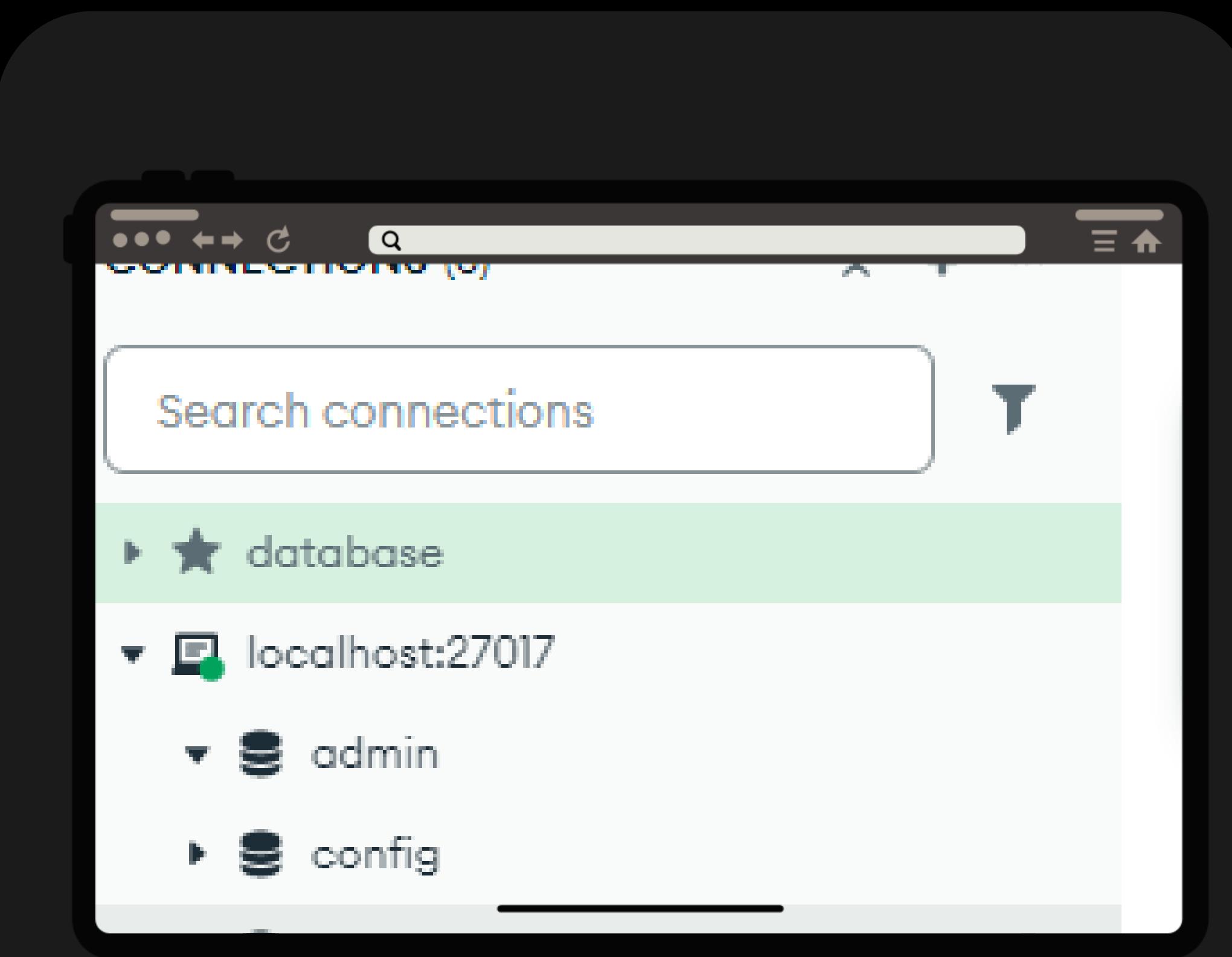
Notifications and activity tracking to keep users informed of important updates and actions.

4. AI-Based Credential Verification

To implement AI for credential verification.

5. Security & Authentication

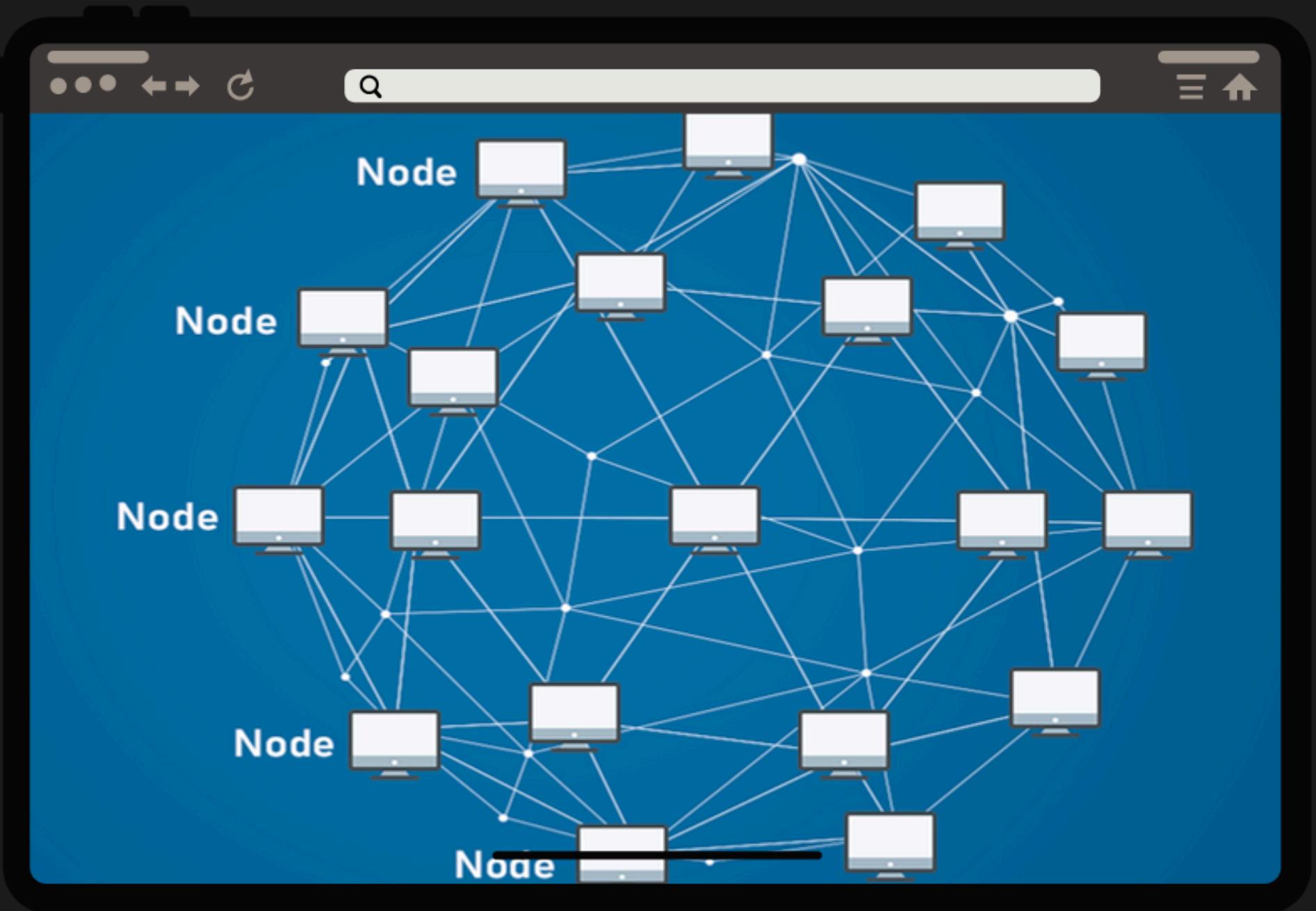
Security, privacy, and user authentication.



...

Blockchain

- Blockchain is a decentralized digital ledger technology that records data across multiple systems in a secure, transparent, and tamper-resistant way.
- Each entry (or "block") is linked to the previous one, forming a chain that makes it nearly impossible to alter past records without detection.
- This makes blockchain ideal for use cases that require data integrity, trust, and transparency, such as healthcare credential verification



[Go Back to Table of Contents Page](#)

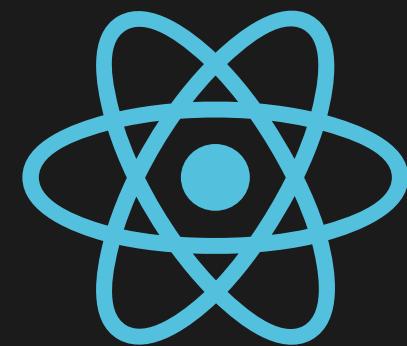




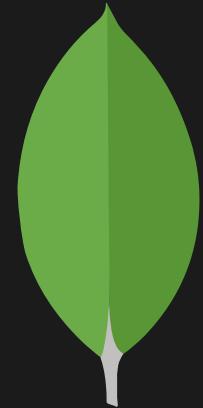
System Overview



Mobile-first
approach
using
Flutter.



Backend
developed
with
Node.js.



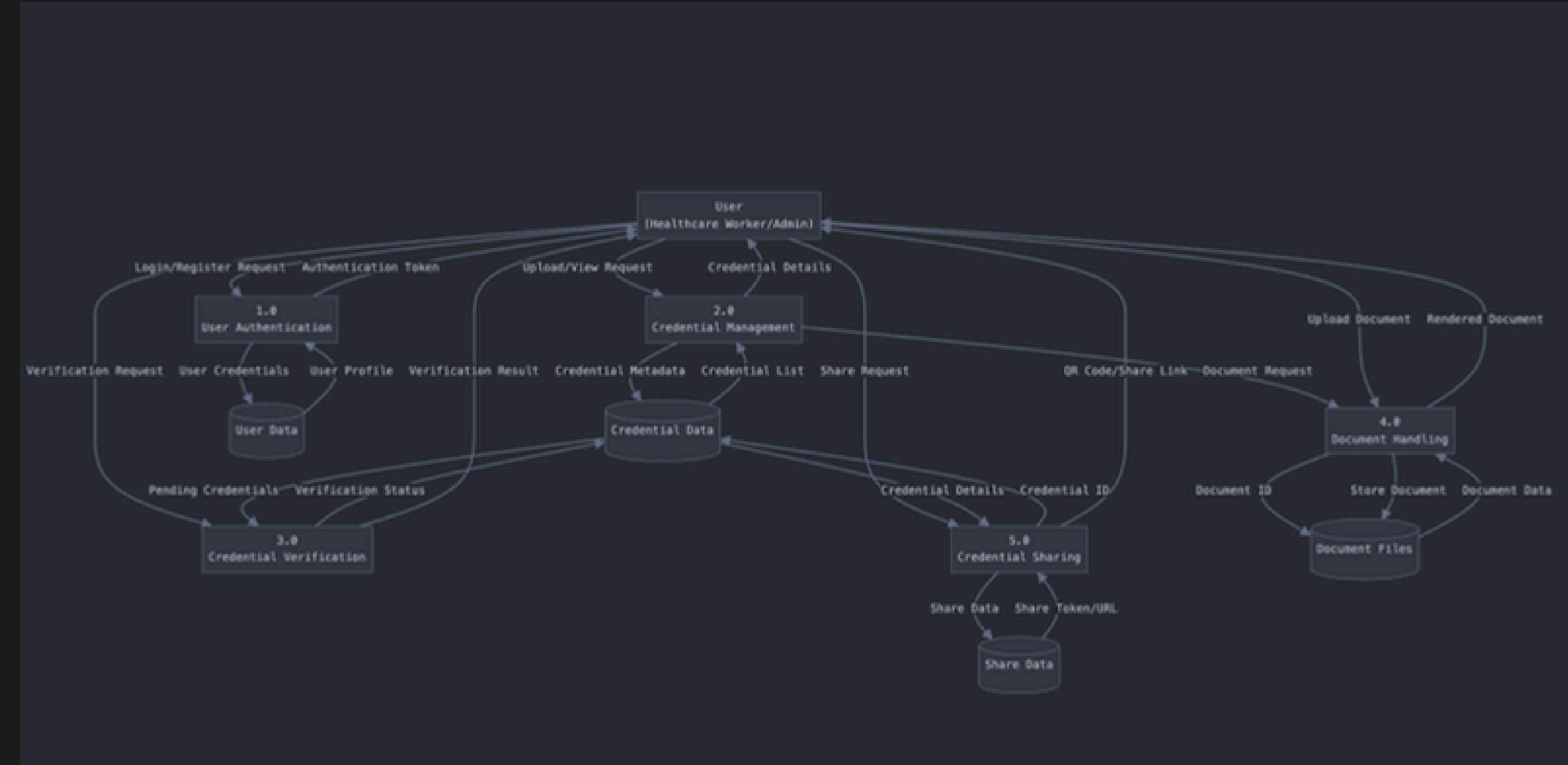
MongoDB
used for
flexible
healthcare
Database.



JWT for user authentication and role-based access control.

...

Conceptual Design



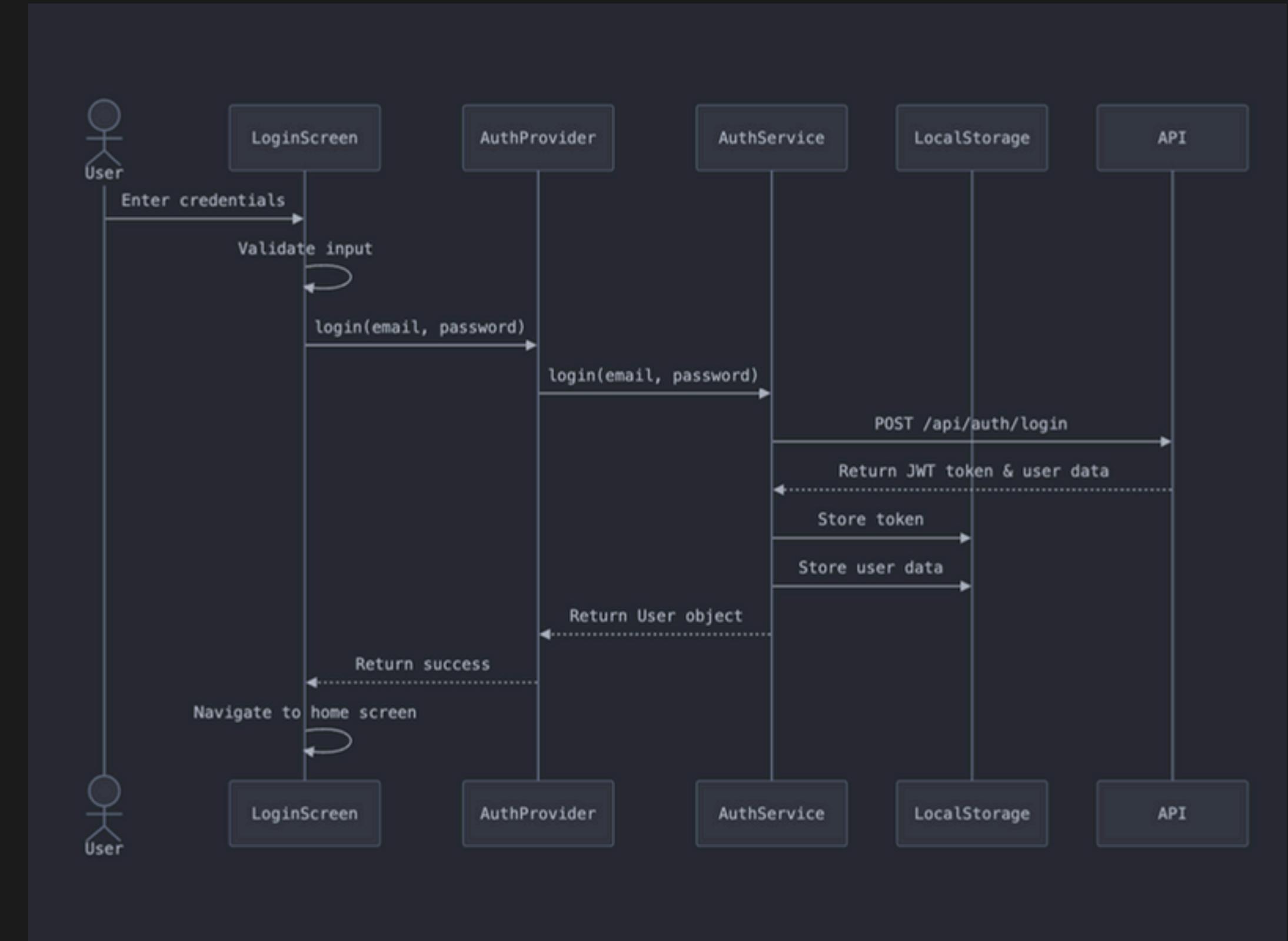
[Go Back to Table of Contents Page](#)



...

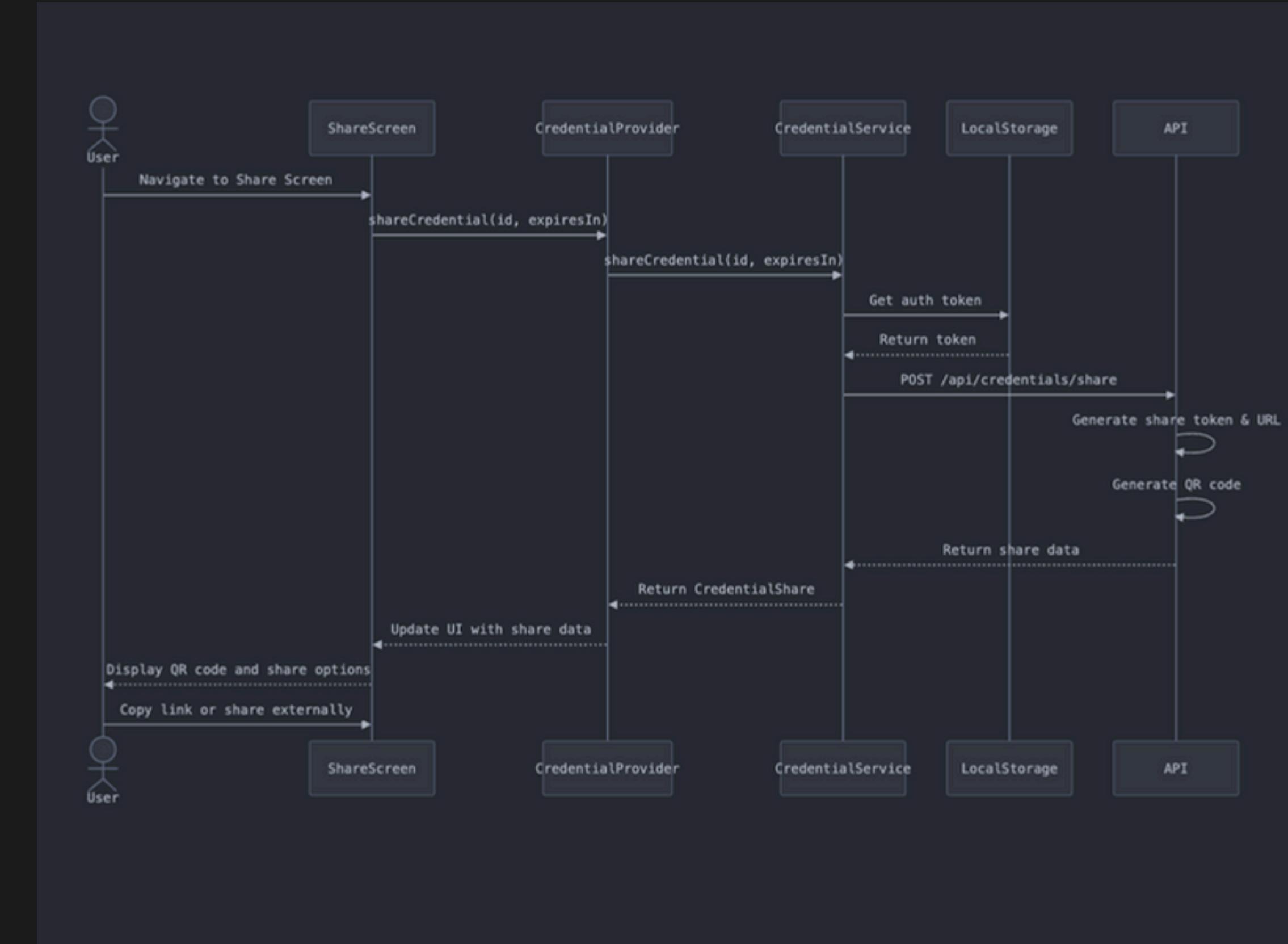
Authentication Flow Diagram

1. User enters email/password
2. Validation performed on input fields
3. Login request sent to API with credentials
4. On success, JWT token received and stored
5. User profile data fetched and stored
6. Navigation to appropriate home screen based on role
7. Error handling for failed login attempts



...

Credential Management Flow Diagram



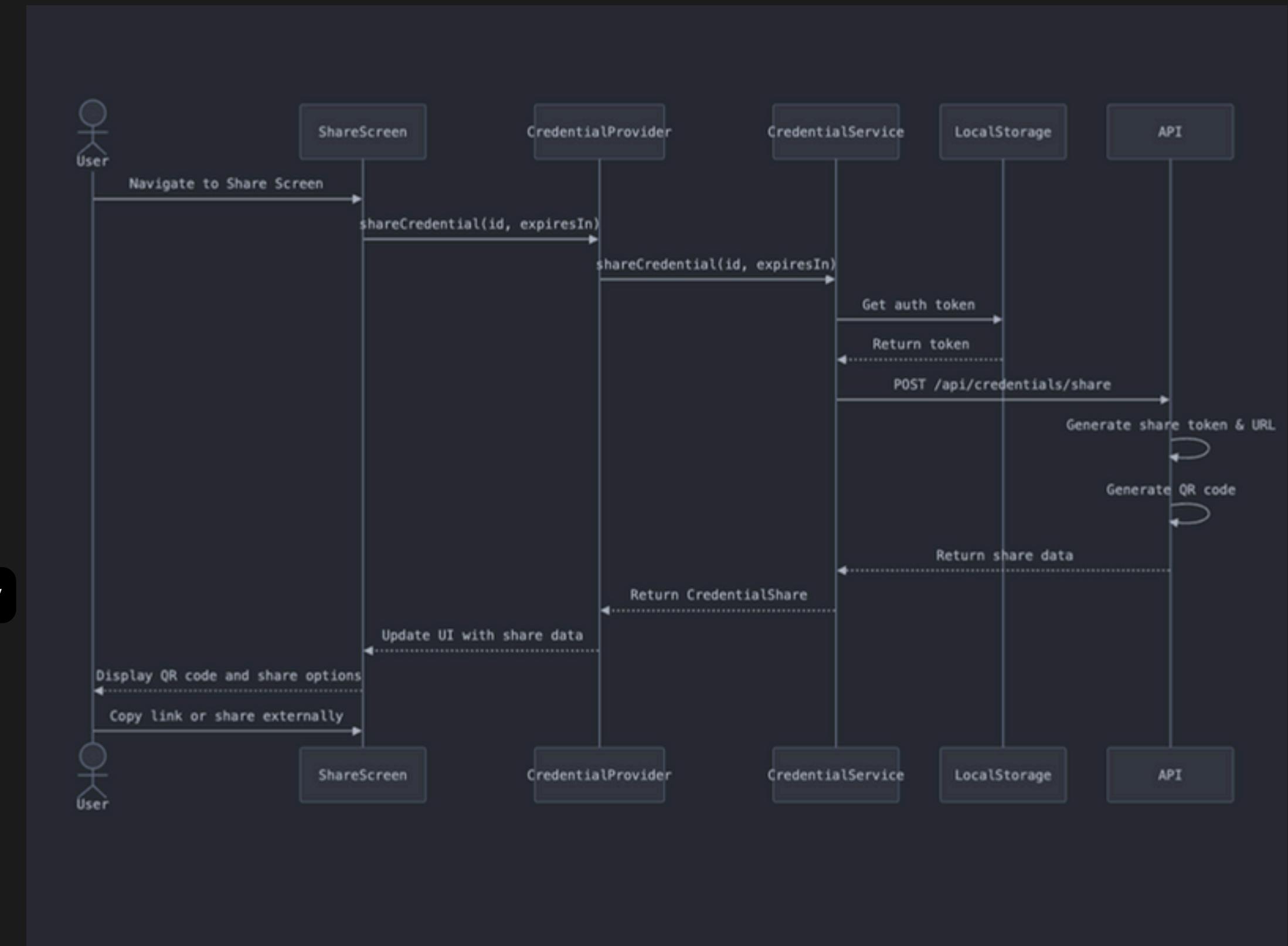
[Go Back to Table of Contents Page](#)



...

Credential Sharing Flow Diagram

1. User selects credential to share
2. Share request sent to API
3. Server generates unique share token with expiry
4. QR code generated from share URL
5. User can set expiration time and regenerate
6. Share options (copy link, share via system)
7. Expiry validation before displaying



[Go Back to Agenda Page](#)



Functional Requirements:

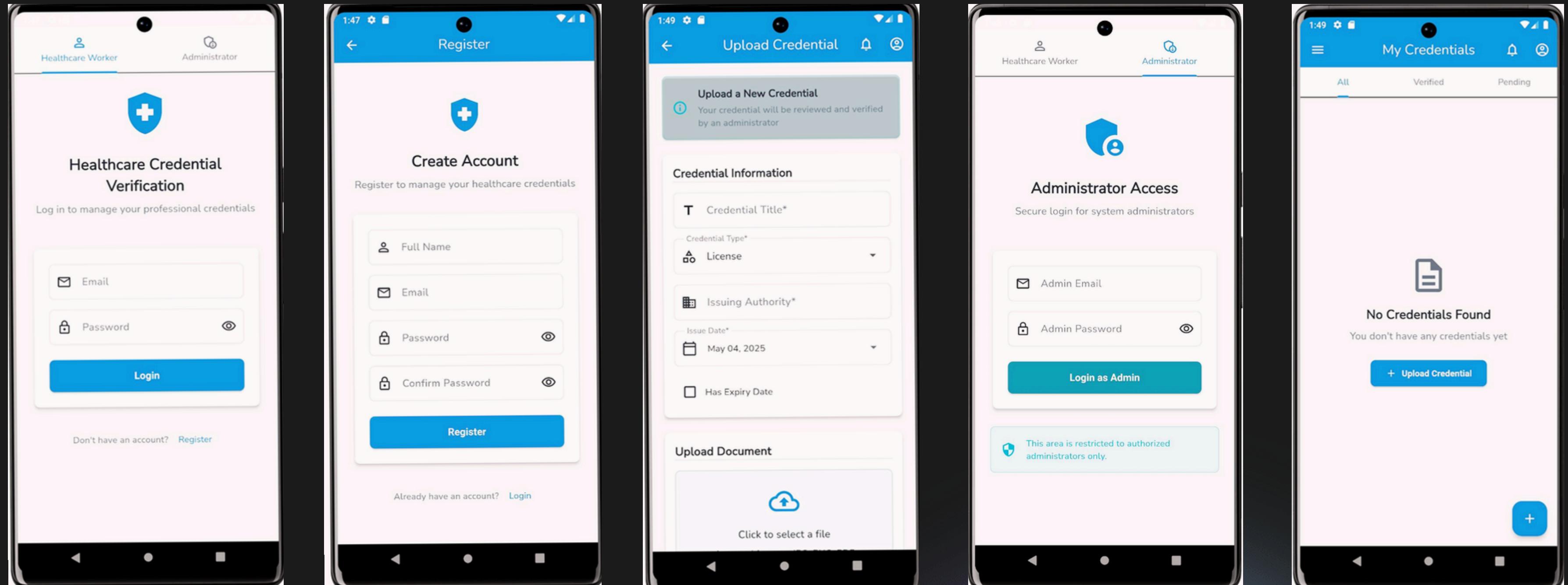
- 1. User Registration and Authentication**
- 2. Credential Verification**
- 3. Credential Management**
- 4. Hospital Admin Access**
- 5. Fraud Prevention**
- 6. User-Friendly Interface**
- 7. Data Security**
- 8. Audit Logs and Reports**
- 9. Cost Reduction and Automation**

Non-functional Requirements:

- Performance
- Scalability
- Security
- Availability & Recovery
- Usability & Accessibility
- Maintainability & Compliance

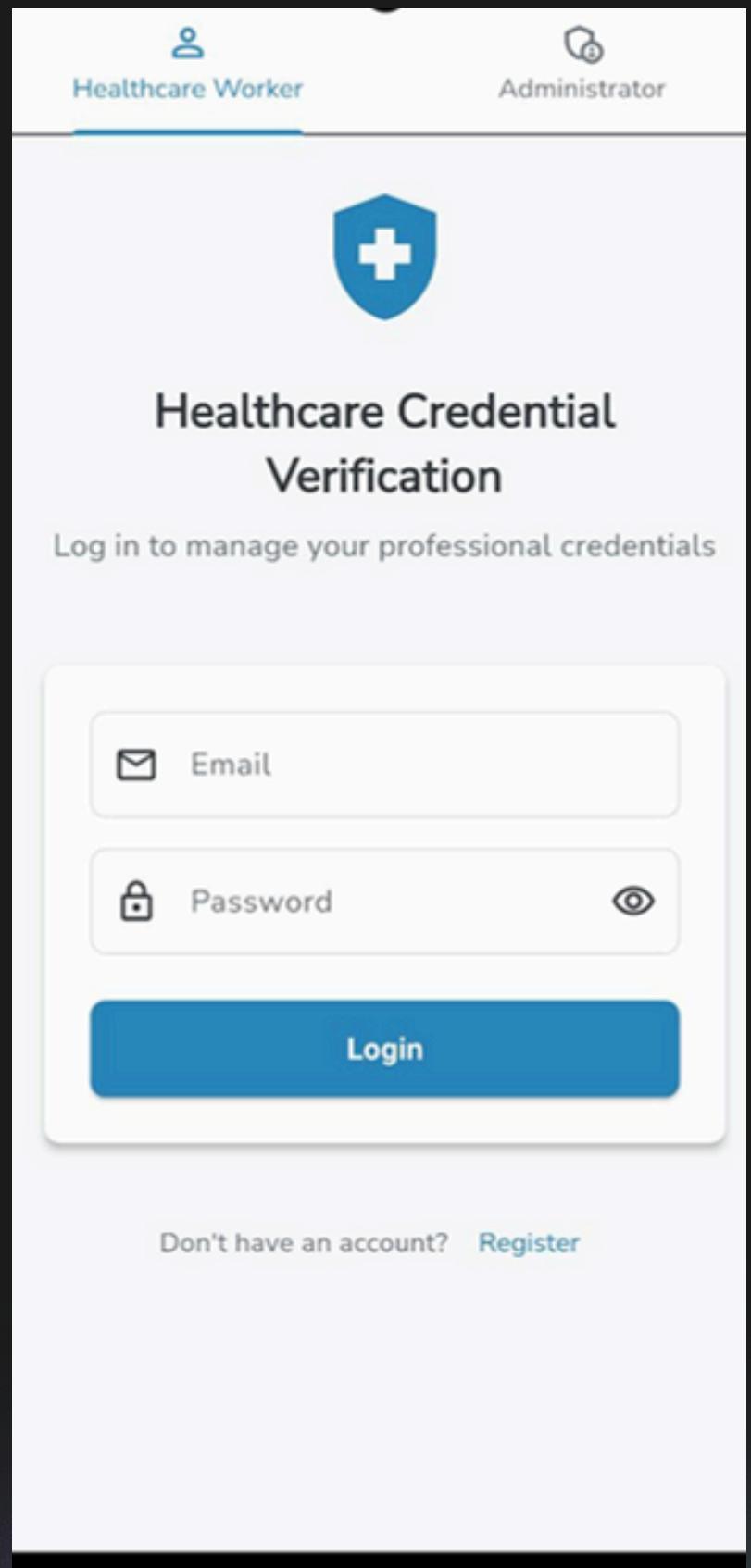
...

User Interface Design

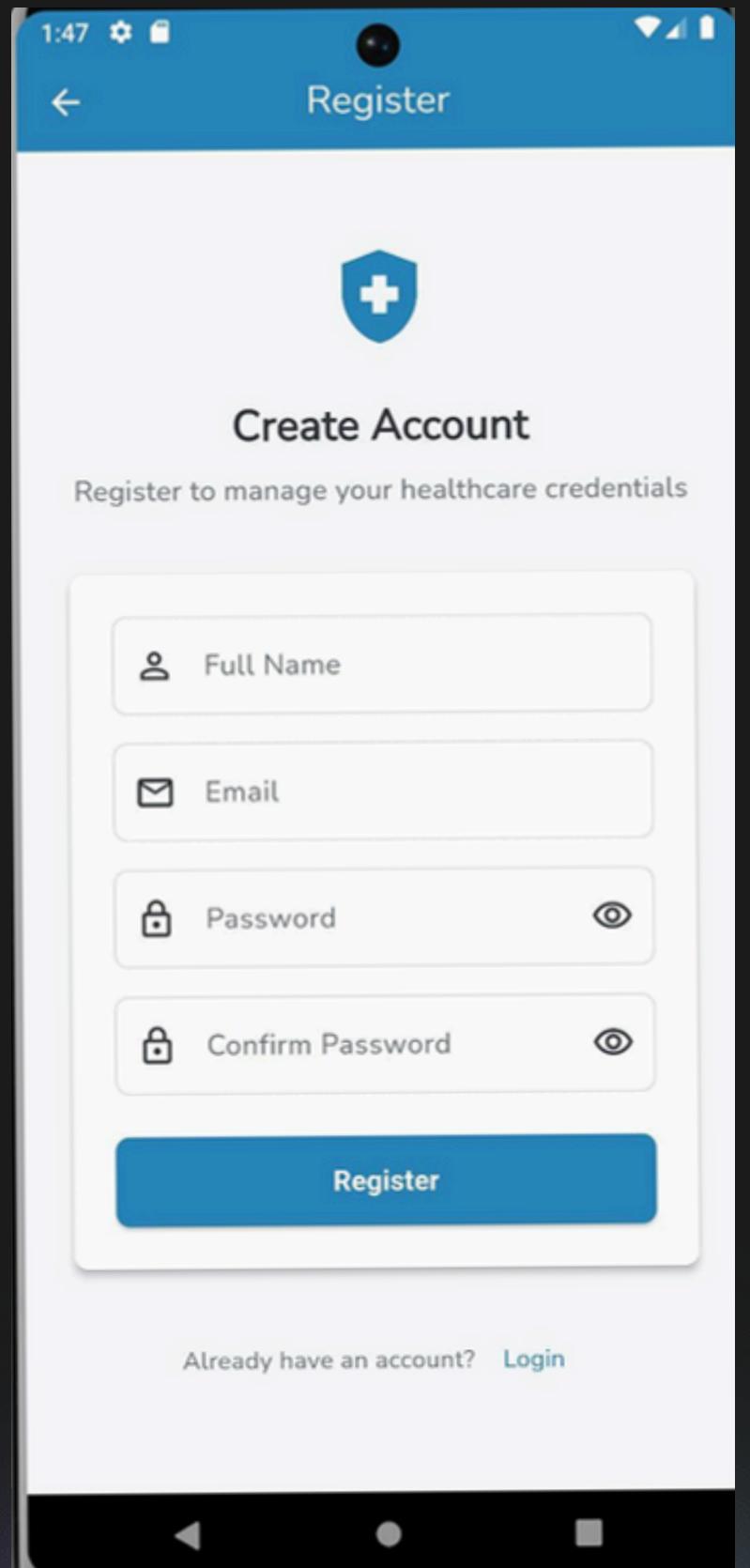


[Go Back to Table of Contents Page](#)

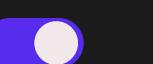


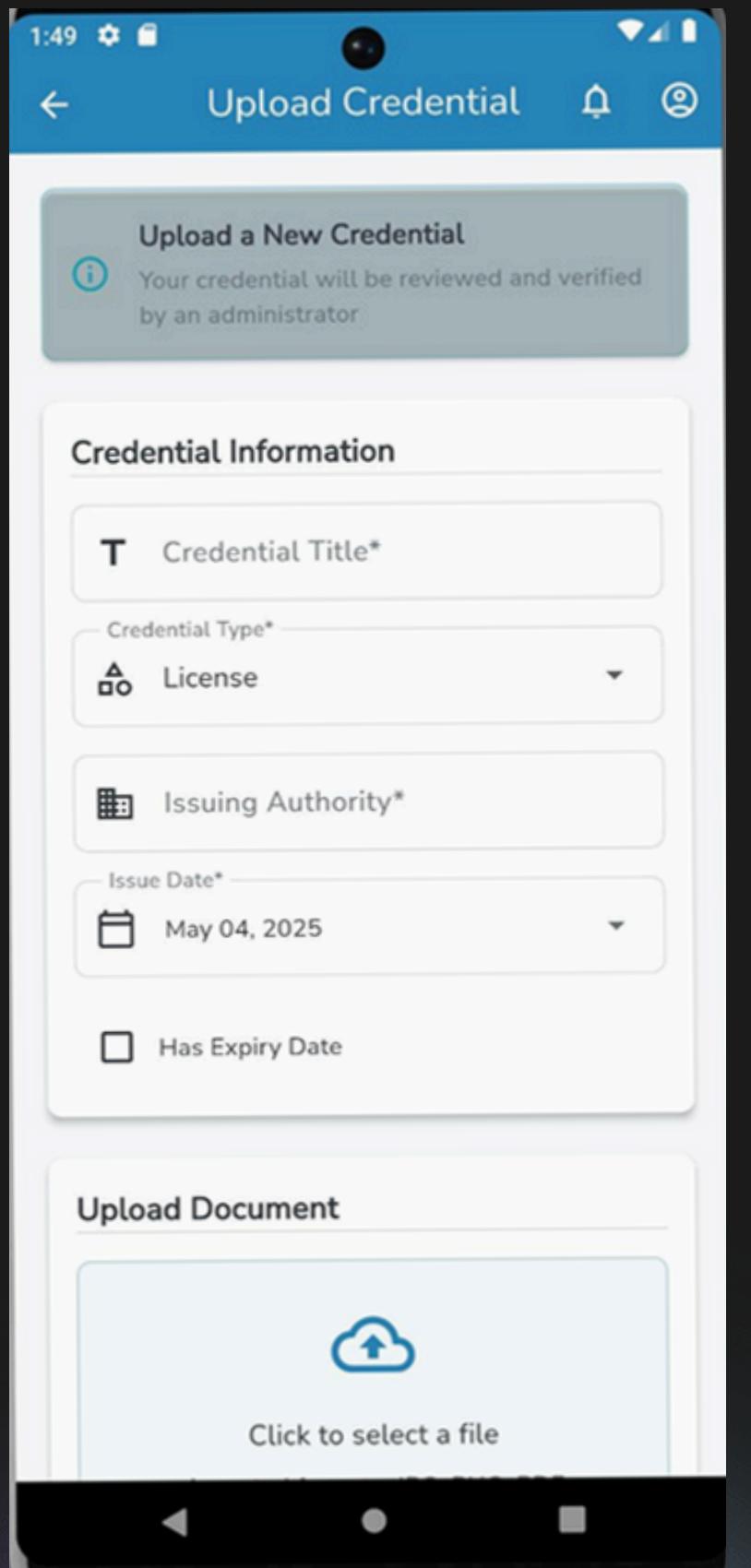


- UI Diagram Login Page 

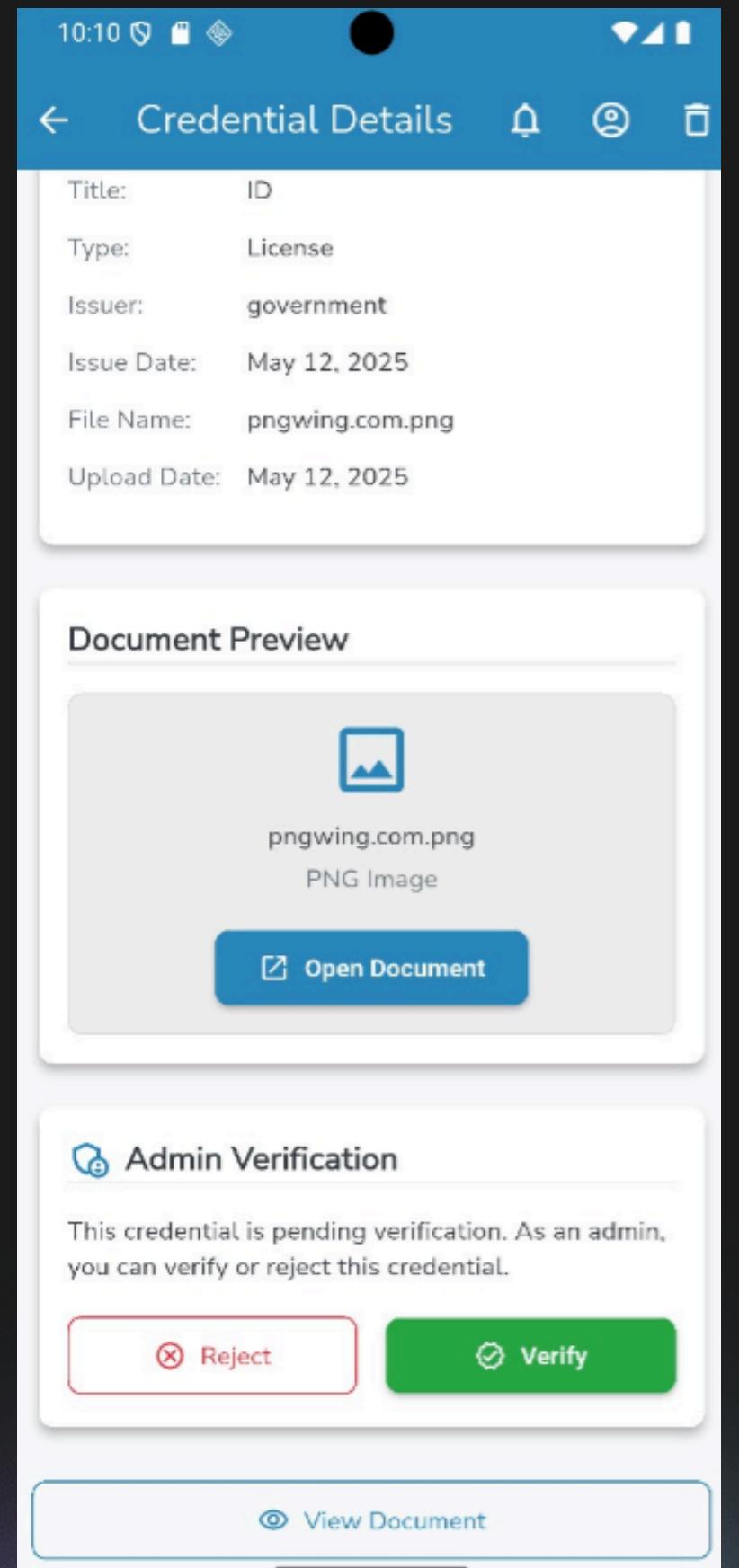


- UI Diagram Register Page

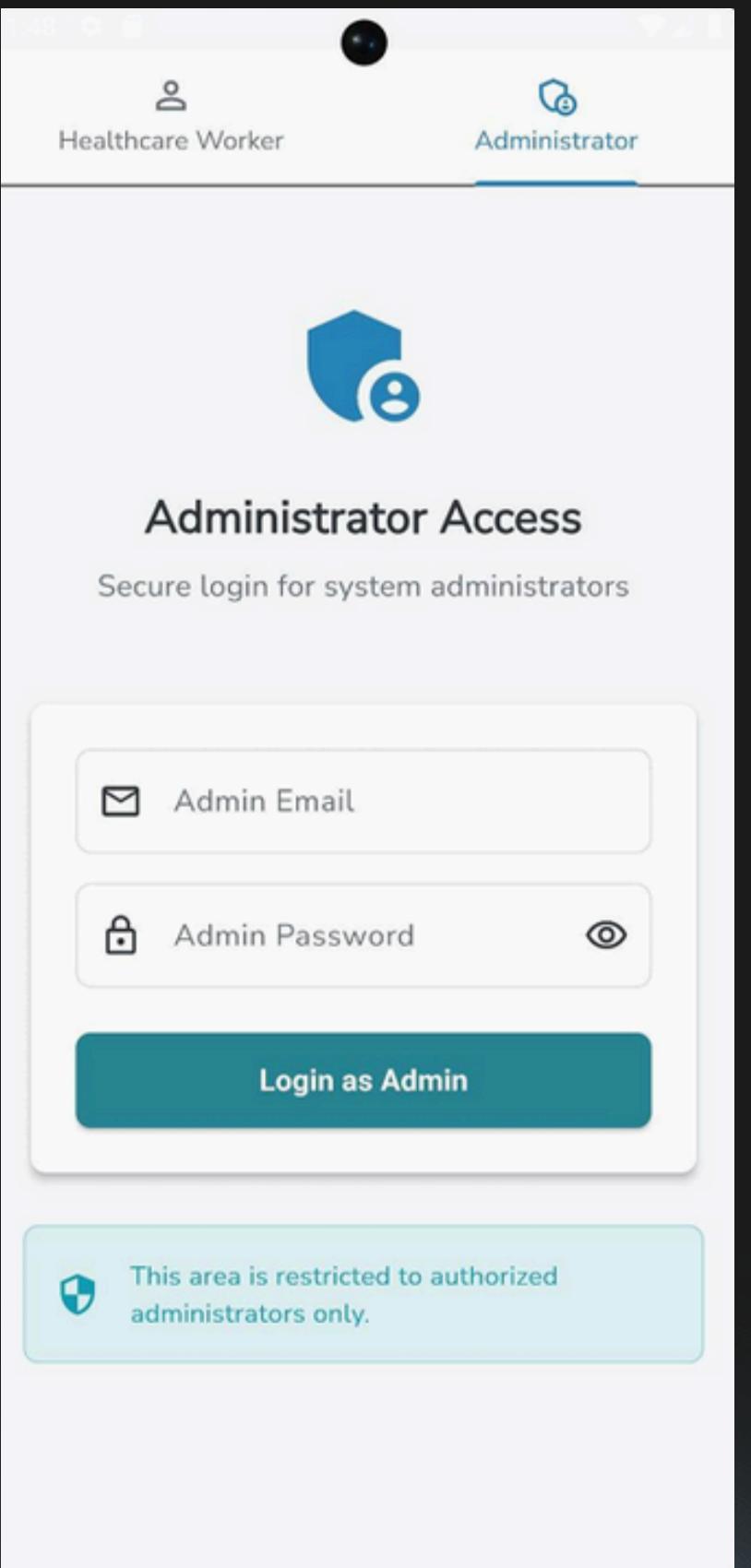




- UI Diagram Credentials Info

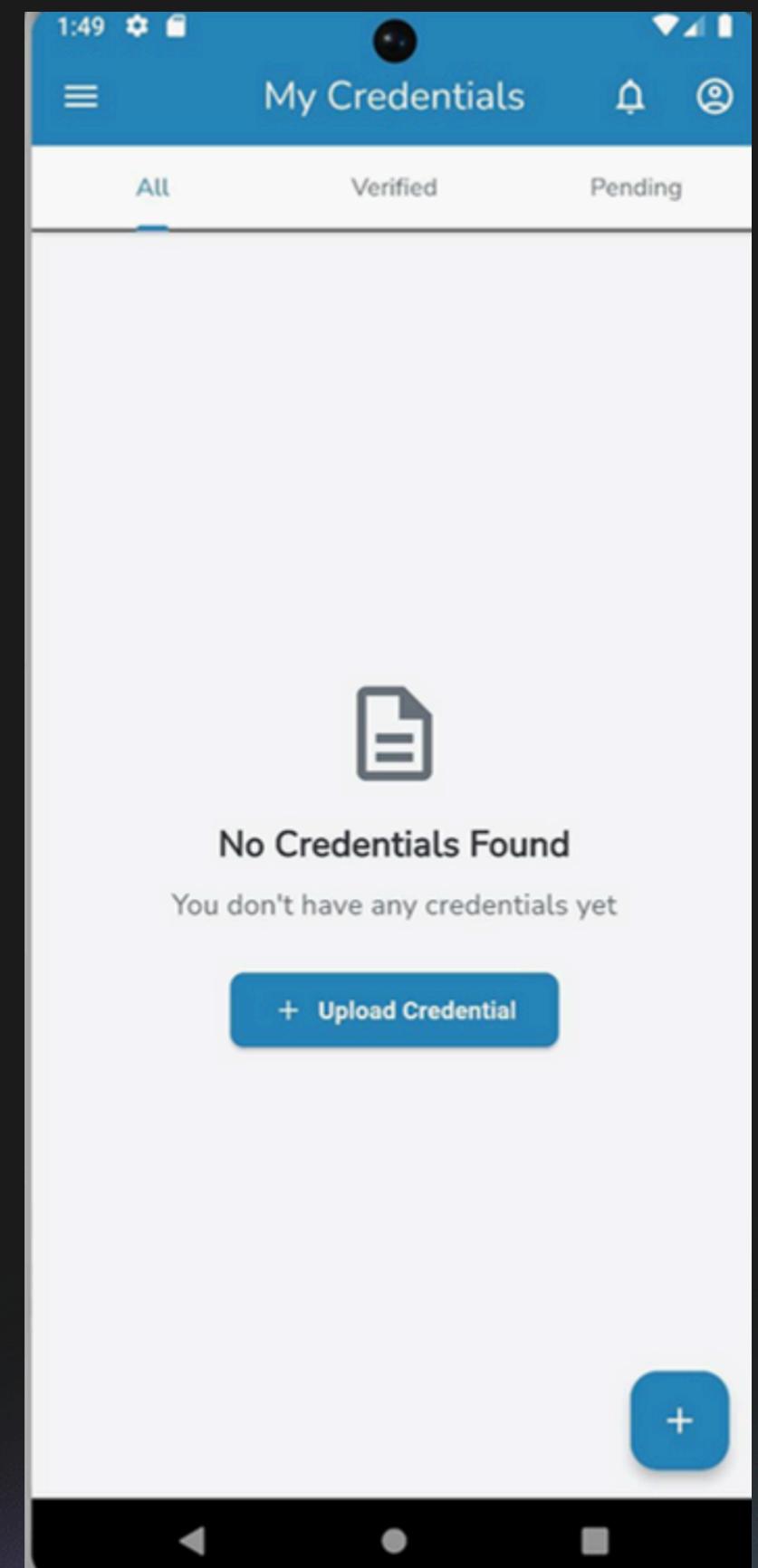


- UI Diagram Admin Verification

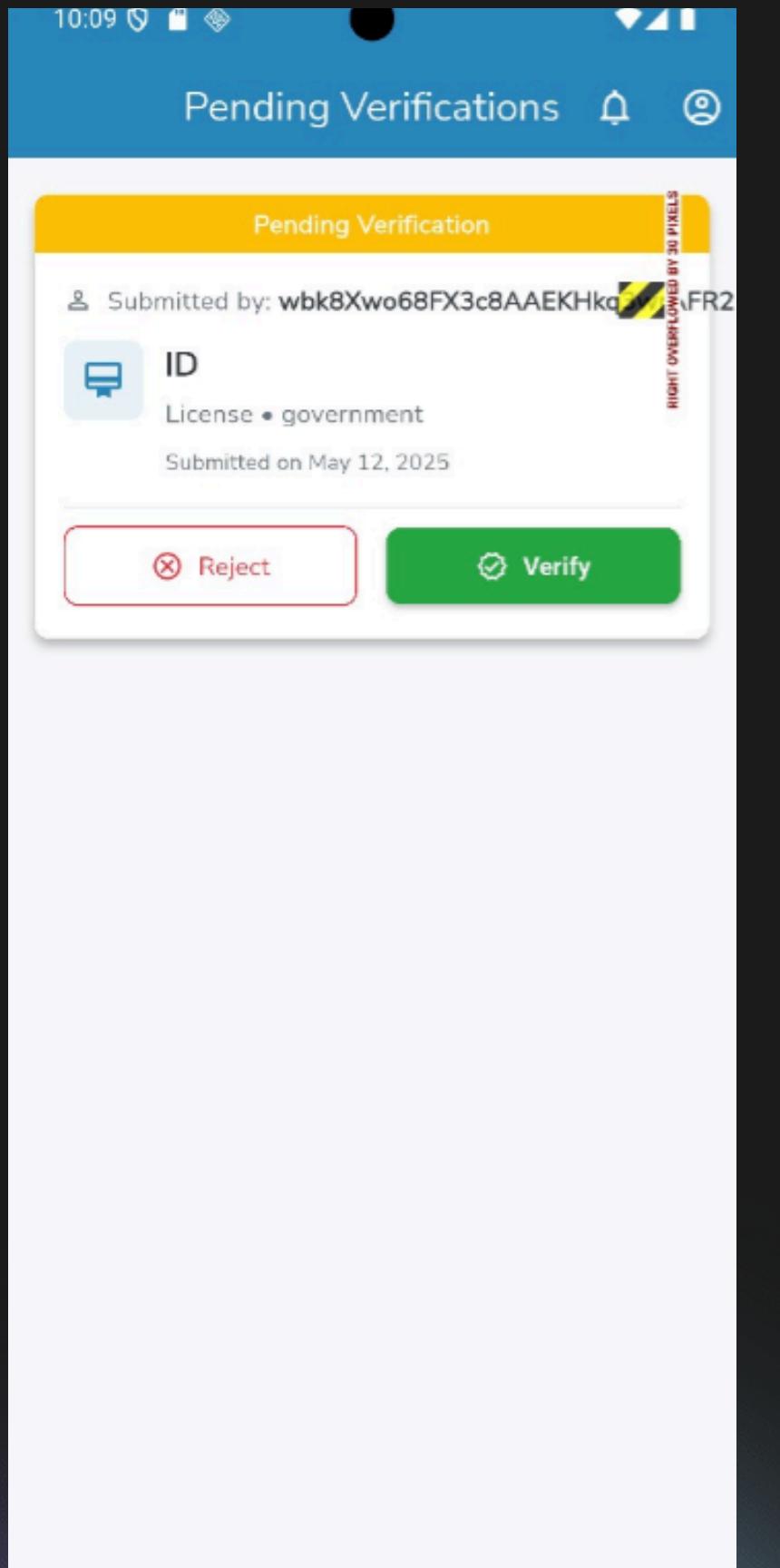


- UI Diagram Admin Page



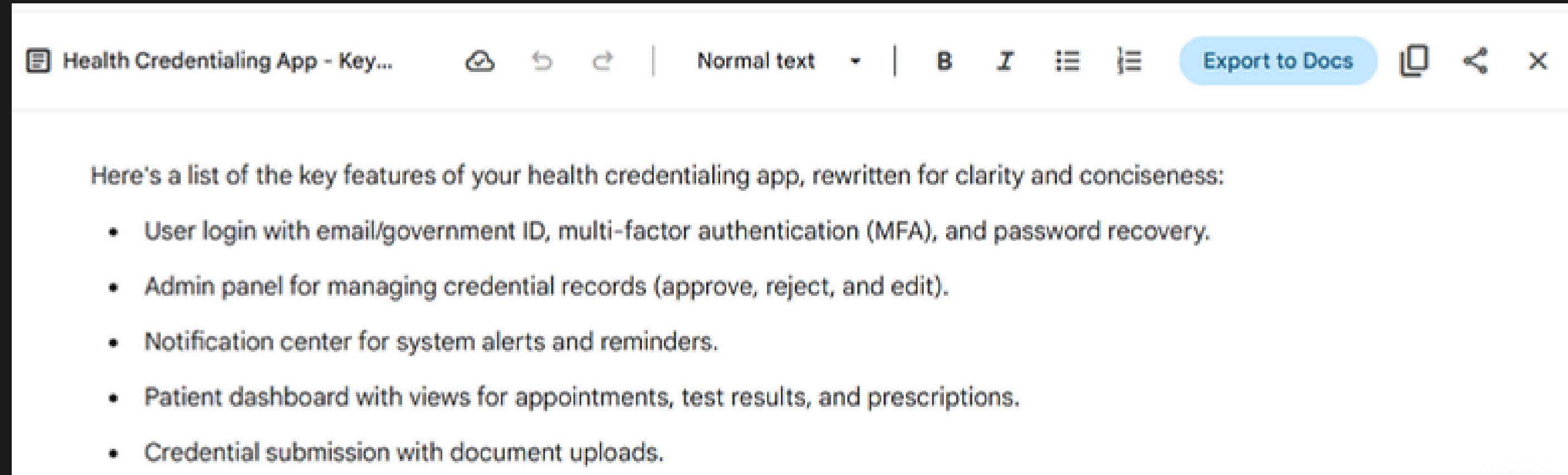


- UI Diagram Credential Upload



- UI Diagram Verification Page

Key Features



The screenshot shows a document editor interface with a dark theme. The title bar reads "Health Credentialing App - Key...". The toolbar includes icons for cloud storage, undo, redo, font size, bold, italic, and alignment. A blue button labeled "Export to Docs" is highlighted. The main content area contains the following text:

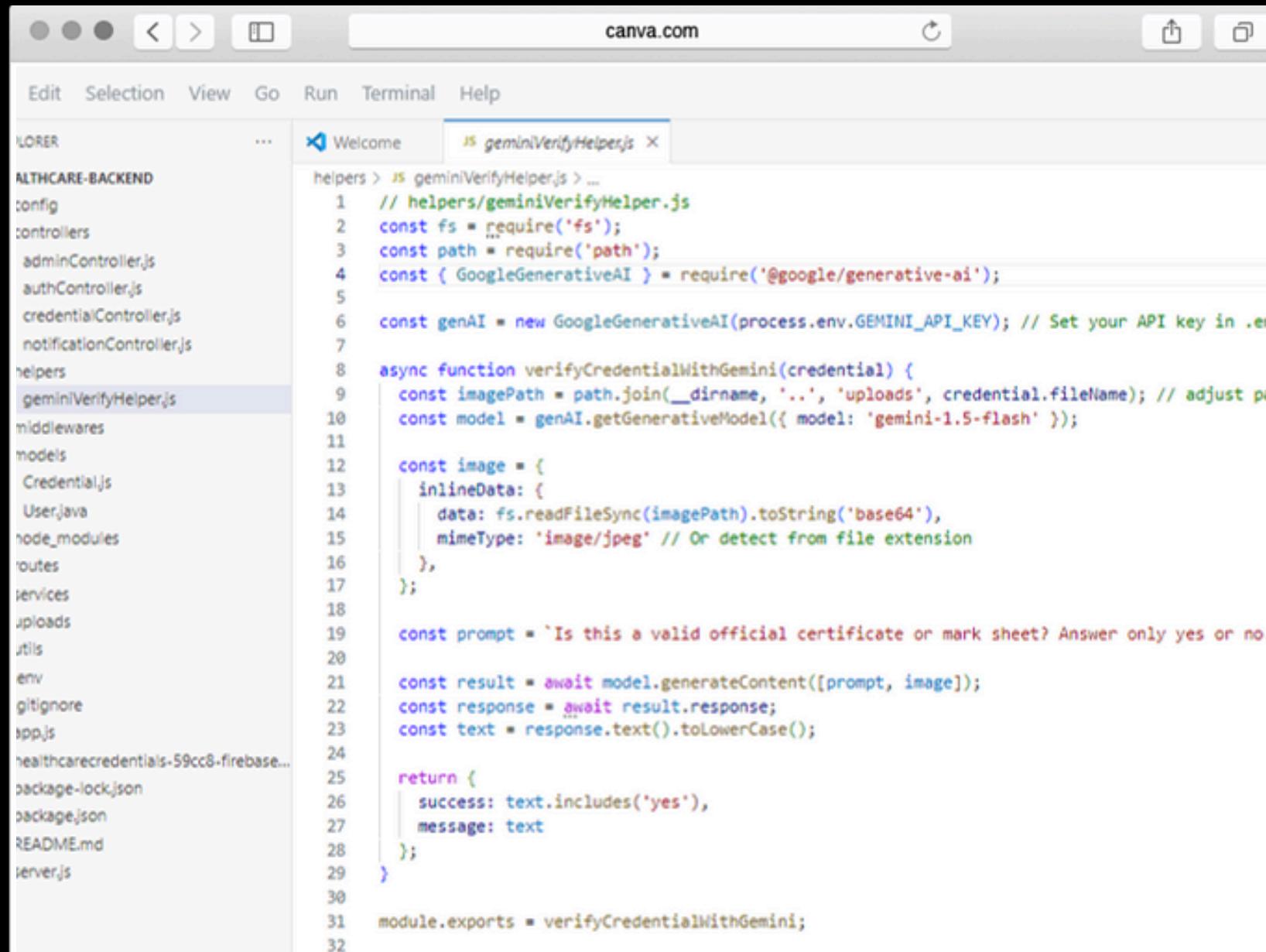
Here's a list of the key features of your health credentialing app, rewritten for clarity and conciseness:

- User login with email/government ID, multi-factor authentication (MFA), and password recovery.
- Admin panel for managing credential records (approve, reject, and edit).
- Notification center for system alerts and reminders.
- Patient dashboard with views for appointments, test results, and prescriptions.
- Credential submission with document uploads.

Google AI Studio

— GEMENIVERIFYHELPER.JS

Used to access Gemini, Google's family of advanced AI models. This specific page allows you to generate an API key to interact with Gemini via REST APIs.



The screenshot shows a code editor window with the title bar "canva.com". The menu bar includes "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". The left sidebar shows a file tree with various files like "config", "controllers", "adminController.js", "authController.js", "credentialController.js", "notificationController.js", "helpers", "geminiVerifyHelper.js", "middlewares", "models", "Credential.js", "User.java", "node_modules", "routes", "services", "uploads", "utils", "env", "gitignore", "app.js", "healthcarecredentials-59cc8.firebaseio...", "package-lock.json", "package.json", "README.md", and "server.js". The main editor area is titled "Welcome" and contains the following code:

```
// helpers/geminiVerifyHelper.js
const fs = require('fs');
const path = require('path');
const { GoogleGenerativeAI } = require('@google/generative-ai');

const genAI = new GoogleGenerativeAI(process.env.GEMINI_API_KEY); // Set your API key in .env

async function verifyCredentialWithGemini(credential) {
  const imagePath = path.join(__dirname, '../', 'uploads', credential.fileName); // adjust path
  const model = genAI.getGenerativeModel({ model: 'gemini-1.5-flash' });

  const image = {
    inlineData: {
      data: fs.readFileSync(imagePath).toString('base64'),
      mimeType: 'image/jpeg' // Or detect from file extension
    },
  };

  const prompt = 'Is this a valid official certificate or mark sheet? Answer only yes or no.';

  const result = await model.generateContent([prompt, image]);
  const response = await result.response;
  const text = response.text().toLowerCase();

  return {
    success: text.includes('yes'),
    message: text
  };
}

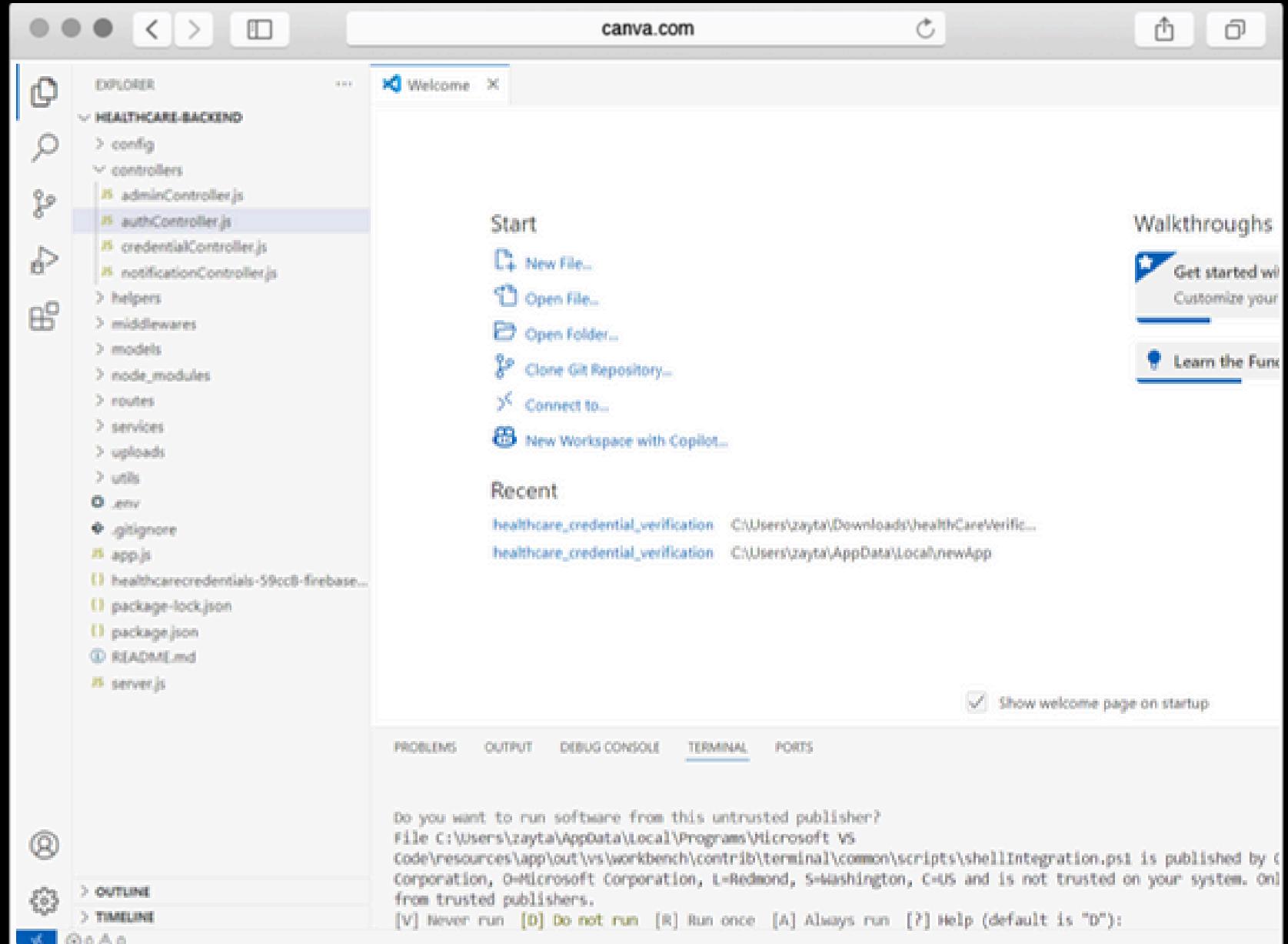
module.exports = verifyCredentialWithGemini;
```

[Get API key | Google AI Studio](#)

Challenges & Obstacles

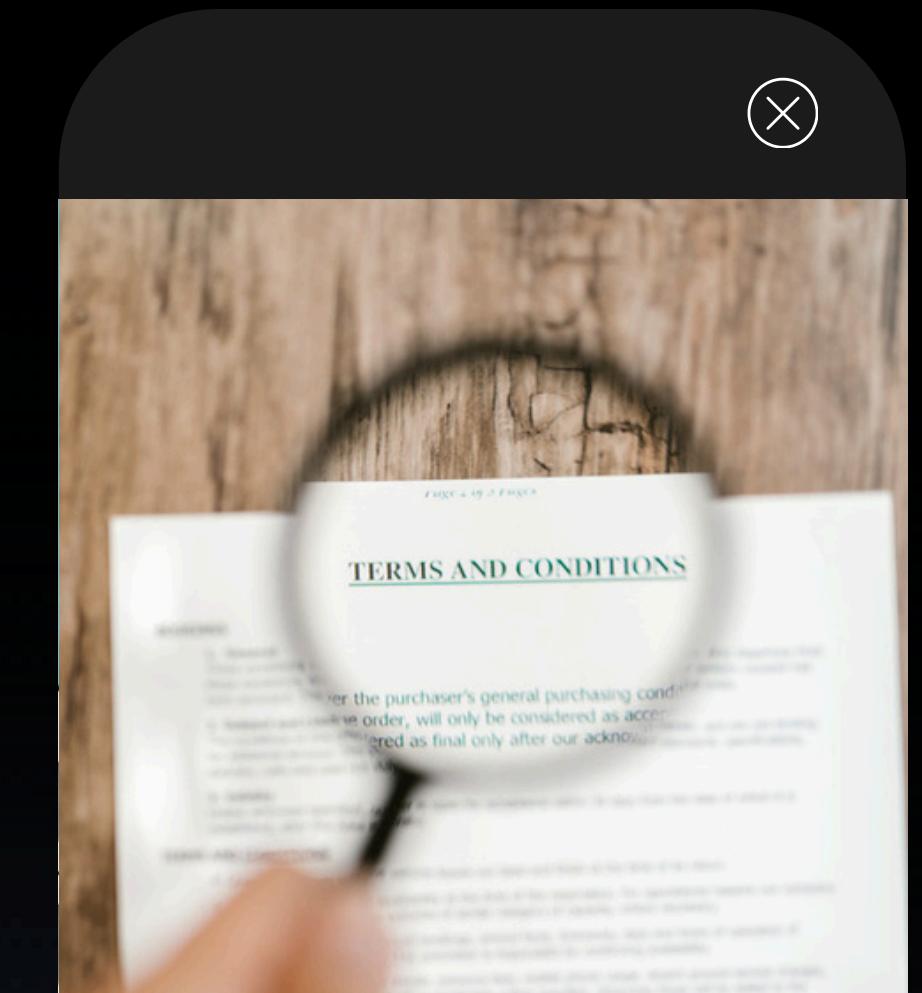
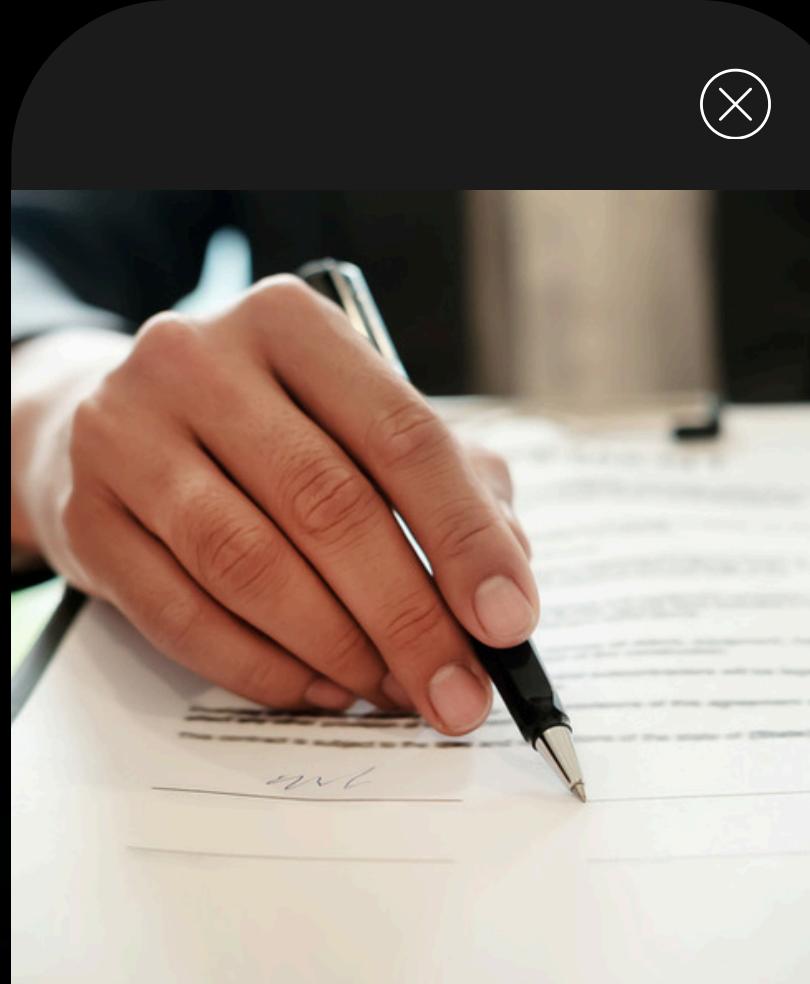
— INCLUDING THEIR SOLUTIONS

- Faced issues with state management; Google's library was unstable, and the alternative was messy. We used ChatGPT, Stack Overflow, and explored Flutter design patterns to resolve this.
- Needed to adapt the app for different user types, like admins and healthcare workers.
- Canceled blockchain data sharing due to challenges with Web3, mobile integration, and time constraints. Integrating AI had its own difficulties, but it was more suitable for our scope.



...

Impact & Benefits



REDUCES ADMIN WORKLOAD

Automating credential verification.

HIRING PROCESS

Streamlining documentation and approval

COMPLIANCE WITH POLICIES AND TRUST

Ensuring verified credentials and real-time updates



Conclusion

For the Credential Verification System to become popular in the healthcare field, it should align with the normal way professionals are interviewed and admitted. This would allow patients and admins to trust their caregivers, and HR recruiters would benefit from reduced workload as well as increased trust. In our proof of concept, we could explore enhancing AI, adding more features and iterations, and revisiting blockchain technology as it continues to modernize.

...



Thank you!

We appreciate your time and attention!

