

Develop a web banking application

Overview

In this task, you need to build a web application for a bank. The user will be able to sign-up & sign in to the bank using a username & password, and validating their phone number using a one-time passcode sent by SMS message.

Technical notes

1. ☐ Use React for frontend, NodeJS for server, Mongo for DB.
2. ☐ Implement the Figma designs for the application UI.
3. 卿 Use Twilio integration for SMS service.
4. 侯 The web application needs to be hosted on a platform of your choice.
5. ☐ Once submitted, send us a git link to the source code + web link to the live app.

Requirements

1. ☐ Sign up
 - a) The user can sign up to the bank by choosing an email address, a password, and inserting their phone number.
 - b) user can't register with the same email address more than once (otherwise, present an error message).
 - c) Validate that the entered email address format and the phone number format are valid.
2. ☐ Phone validation
 - a) During sign up, an SMS message with a passcode (6 random digits) is sent to the user in order to validate their phone number.
 - b) Once the user enters the correct passcode, the registration process is succeeded and the user is redirected to the dashboard.
 - c) If the passcode is not correct, show an error message.
3. 卿 Dashboard
 - a) The dashboard page is protected, access only after sign up / sign in
 - b) A random account balance will be set once the user was sign up.
 - c) The user can see their balance and recent transactions.
 - d) The user can sign out by pressing the Sign Out button, and redirect to the login page.
4. 侯 Login
 - a) The user can sign in by inserting their correct email & password.
 - b) If validated, the user is redirected to the dashboard. Otherwise, present an error message.
5. ☐ Bonus: Transaction
 - a) The user can send money to another registered user, by typing the user's email address & amount, and press Submit.
 - b) If the user doesn't have enough balance, or the email address doesn't exist, show an error message.
 - c) Once submitted successfully, present another transaction record
 1. for sender: the receiver email & the sent amount with '-' sign.
 2. for receiver: the sender email & the received amount with '+' sign.

Good luck!