



Banking Application Capstone Presentation

Dipo Omotoso

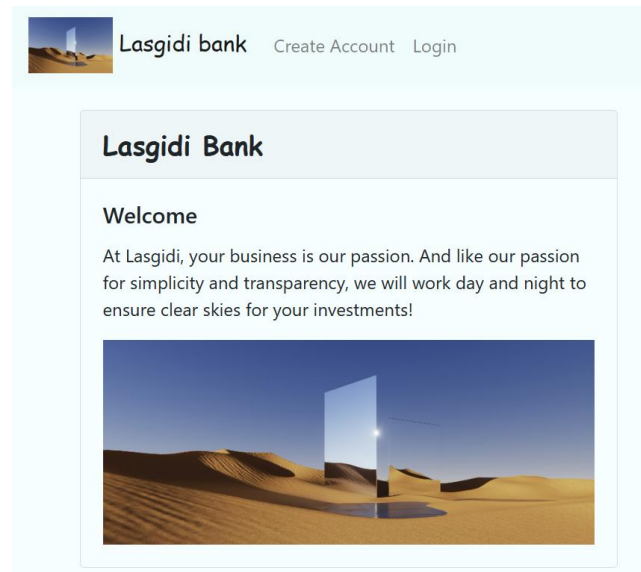


xPRO

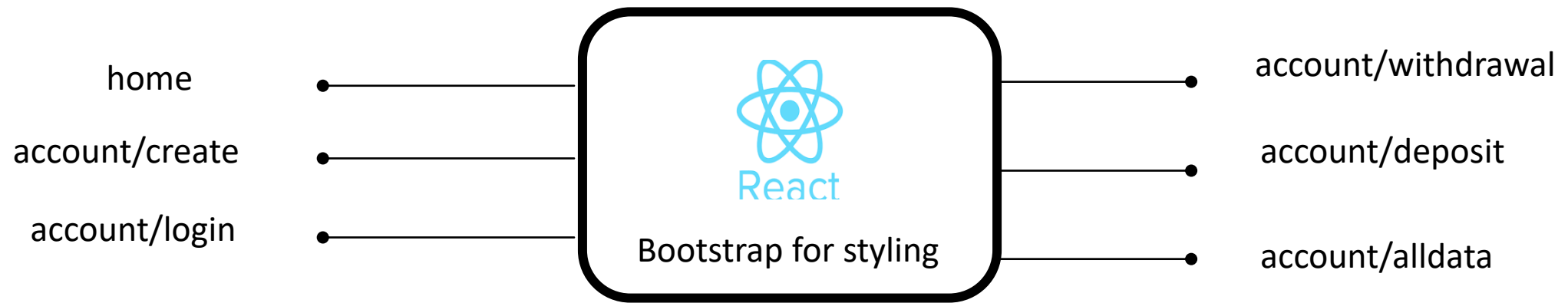


Part 1: Front-End Architecture, Authentication, And App Diagram

Application Overview Diagram



Front-End Architecture



Authentication

Authentication

[Users](#) | [Sign-in method](#) | [Templates](#) | [Usage](#) | [Settings](#) | [Extensions](#)

[Add user](#) [Refresh](#) [More](#)

Identifier	Providers	Created ↓	Signed In	User UID
remi@doem.org		Mar 23, 2024	Mar 23, 2024	
kori@doem.org		Mar 23, 2024	Mar 23, 2024	
ayoka@doem.org		Mar 23, 2024	Mar 23, 2024	
dipo@doem.org		Mar 23, 2024	Mar 24, 2024	

Rows per page: 50 [1 - 4 of 4](#) [Previous](#) [Next](#)



xPRO



Part 2: Database And API

RESTful API

- Create user account: `get('/account/create/:name/:email/:password')`
- Login: `get('/account/login/:name/:email/:password')`
- Find user account: `get('/account/findOne/:email')`
- Account update: `get('/account/update/:email/:amount')`

Database

[+ ADD DATA](#)

EXPORT DATA

 UPDATE

 DELETE

```
_id: ObjectId('65ff5aa518bb85292405a0b7')  
name: "dipo"  
email: "dipo@doem.org"  
password: "verysecret"  
balance: 835
```

```
_id: ObjectId('65ff5b8918bb85292405a0b8')  
name: "kori"  
email: "kori@doem.org"  
password: "verysecret"  
balance: 0
```

```
_id: ObjectId('65ff7be57fb6a28060ca5a74')  
name: "ayoka"  
email: "ayoka@doem.org"  
password: "verysecret"  
balance: 0
```

```
_id: ObjectId('65ffaf81b0121643f8be8bae')  
name: "remi"  
email: "remi@doem.org"  
password: "verysecret"  
balance: 0
```

Authentication

Users

Sign-in method

Templates

Usage

Settings

 Extensions

🔍 Search by email address, phone number, or user UID

Add user

⌂ ⋮

Identifier	Providers	Created ↓	Signed In	User UID
remi@doem.org	✉	Mar 23, 2024	Mar 23, 2024	
kori@doem.org	✉	Mar 23, 2024	Mar 23, 2024	
ayoka@doem.org	✉	Mar 23, 2024	Mar 23, 2024	
dipo@doem.org	✉	Mar 23, 2024	Mar 24, 2024	

Rows per page: 50 1 - 4 of 4 < >

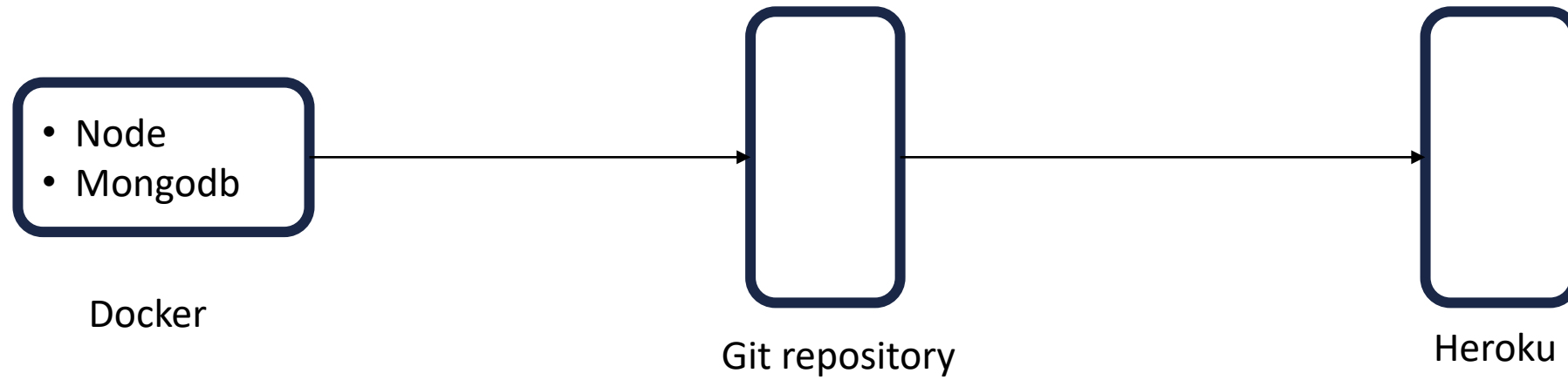


xPRO



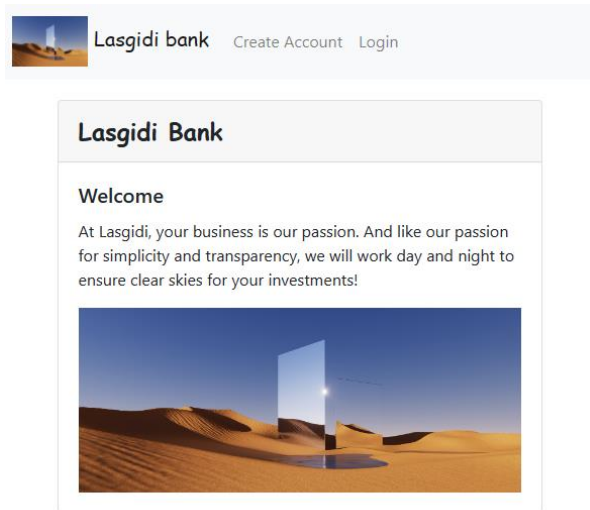
Part 3: Deployment, Additional Features, App Demonstration, Reflection

Deployment

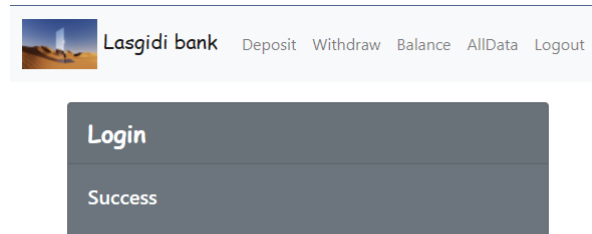


Additional Features

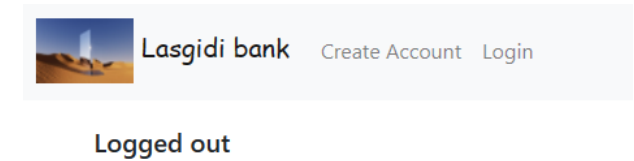
Welcome page (no user logged in)



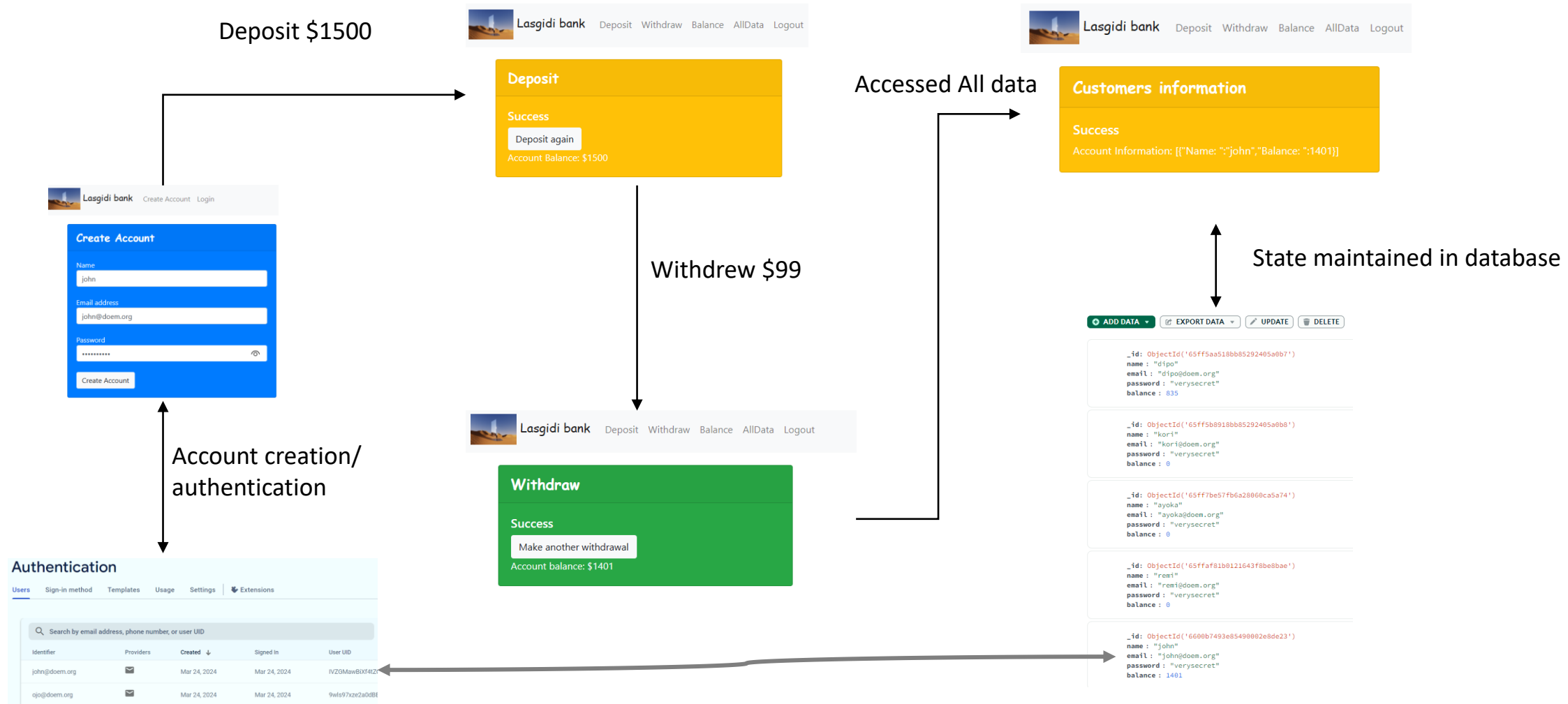
User logged in (all services available)



User logged out



Application Demonstration – Create Account, Log In, Deposit, Withdraw



- If I am starting the Banking App today, and given the knowledge I've gained in the course, below are a few decisions I would make prior to starting to code.
 1. Use ES6 modules from ground up to minimize version compatibility, for example, it took almost a day to figure out that the firebase scripts for initialization must be *firebase-app-**compat**.js* and *firebase-auth-**compat**.js* if one is not using ES6
 2. Will use serverless service for database and storage to minimize overhead
 3. Better UI to mimic a real banking App
 4. Implement transfers between authorized clients

