

Mobile Financial Service (MFS) Task

Objective: Create a basic MFS application (like bKash or Nagad) using React.js, Node.js, Express.js, and MongoDB. Implement essential features: user authentication, send money, cash-out, and balance inquiries. Ensure a simple and secure web interface.

Roles: **There will be 3 roles here.** One for User and one for Agent, one for Admin

Features that has to be implemented:

For User:

1. Registration:

- An user needs to give these Inputs for the registration: **Name, 5-digit PIN (must be number), Mobile Number, Email.** Initially user status will be pending which will be approved by the admin later.
- **Use JWT for authentication and hash PINs for security** (you may use bcrypt.js for pin hashing). PINs must be hashed before saving it in the database. Plain PINs must not be saved in the database. And for verifying PINs you have to compare the hashed password saved in DB to the password from the input.
- After the account is activated by admin approval, **the user will get a Bonus of 40 Taka** which will be credited to the user's balance. This is a one time bonus for new users.

2. Secure Login:

- Users can log in using their **Mobile Number/Email and PIN.** That is either the user can use a mobile number and pin. Or users can use Email and pin.
- **Use JWT for login security.** Logging in will generate a token from the server and it will be saved to the browser.

3. Send Money:

- For every transaction (Send Money), users have to provide their pin. **Pin & JWT Verification** is required.
- Users can send money to other users.
- For every transaction over 100 taka, a user has to pay a fee of 5 Taka.
- An user needs to do a transaction with at least 50 taka. Less than 50 taka is not allowed for transactions.

4. Cash-Out:

- For every transaction (Cash out), users have to provide their pin. Pin & JWT Verification is required
- **Users can only cash out through an agent.** The amount of cash out will be deducted from the user's balance and added to the agent's balance.
- For every cash out, there will be a fee which is **1.5% of the transaction amount** and the fee will be deducted from the user's balance and added to the balance of the agent.

5. **Cash-in:**

- Users can cash-in through agents without a fee.
- For cash-in the user will send a request to an agent, if the agent approves the request, then the amount will be added to the balance of the user and deducted from the balance of the agent.

6. **Balance Inquiry:**

- Users can check their account balances at any time.

7. **Transactions History:**

- JWT Verification is required to see transaction history.
- Users can view their last 10 transactions.

For Agent:

1. **Registration:**

- a. An agent needs to give these Inputs for the registration: **Name, 5-digit PIN(must be number), Mobile Number, Email**. Initially agent status will be pending which will be approved by the admin later.
- b. After the account is activated by admin approval, the agent will get a Bonus of **10,000 Taka** which will be credited to the agent's balance. This is a one time bonus for new agents.

2. **Secure Login:**

- a. Agents can log in using their **Mobile Number/Email and PIN**. That is either the agent can use a mobile number and pin. Or agents can use Email and pin.
- b. Use JWT for authentication and hash PINs for security (you may use bcrypt.js for pin hashing). PINs must be hashed before saving it in the database. Plain PINs must not be saved in the database. And for verifying PINs you have to compare the hashed password saved in DB to the password from the input.

3. **Transaction Management:**

- a. Agents can manage transactions like e.g. cash-in request or cash-out request. If the agent approves the cash-in request from the user, the amount will be deducted from the agent's balance and added to the user's balance. And if the agent approves a cash-out request from the user, the amount will be added to the agent's balance and deducted from the user's balance.

4. **Balance Inquiry:**

- a. Agents can check their account balances at any time.

5. **Transactions History:**

- a. JWT Verification is required to see transaction history.
- b. Agents can view their last 20 transactions.

Admin:

1. **Secure Login:**

- Admin can log in using their Mobile Number/Email and PIN.
- Use JWT for authentication and hash PINs for security (you may use bcrypt.js for pin hashing)

2. **User Management:**

- JWT Verification is required
- View all users, search any specific user by name, and manage user accounts like activate account/block account..

3. **System Monitoring:**

- JWT Verification is required
- Admin can see all transactions within the system.

UI Instructions:

- **Dashboard Only:** No landing page is required. Please make it visually attractive.
- **Protected Application:** User will be redirected to login page if he/she is not authenticated.
- **Responsive Design:** Dashboard must be responsive for desktop and mobile.
- **Dashboard Pages:**
 - Overview: Shows relevant stats for the user (Name, Email address, Phone number, Account Balance) .
 - Transactions: Shows the transaction history for the logged-in user.
 - Other pages should be added according to the need of the user role.