Peer-to-Peer Payment App - Documentation

Overview

- App for users to register, log in, send/request money, view history, add funds, and manage requests.
- Uses HTML for UI, PHP for backend logic, MySQL for storage, and JS for interactivity on the HTML files.

Main Files and Their Purposes

Frontend (HTML + JS)

- index.html – Login form for users.
- register.html – User registration form.
- history.html – Shows user's balance, transaction history, and main navigation.
- transaction.html – Form to send/request money.
- pending.html – Displays pending requests to accept/decline.
- account.html – Add funds to your account (shows card number for logged in user).
- search.html – Search for users by username/email.
- style.css – Styles for all pages.
- loadRequests.js – Loads pending requests (AJAX) to pending.html.

Backend (PHP)

- db.php – Database connection details, included in all backend scripts.
- create_user.php – Handles registration, checks for duplicates, creates user and account (with random card number).
- login.php – Authenticates user, starts session.
- logout.php – Logs out and destroys session.
- transactions.php – Handles sending/requesting money and updates database.
- history.php – Returns user balance and transaction history.
- load_requests.php – Shows logged-in user's pending payment requests.

- handle_requests.php – Handles accepting/declining requests.
- add_to_account.php – Adds money (with 2-day delay logic).
- session_check.php – Checks if session is still active (AJAX endpoint).

Database Structure

- **Users**
    - UserID (PK)
    - Username (unique)
    - Email (unique)
    - PasswordHash
    - CreatedAt
- **Accounts**
    - AccountID (PK)
    - UserID (FK)
    - Balance (Decimal)
    - CardNumber (16-digit, unique, random)
    - CreatedAt
    - UpdatedAt
- **Transactions**
    - TransactionID (PK)
    - SenderID (FK)
    - ReceiverID (FK)
    - Amount (Decimal)
    - TransactionDate
- **Requests**
    - RequestID (PK)

- RequestorID (FK)
- RequesteeID (FK)
- Amount (Decimal)
- Status (PENDING/ACCEPTED/DECLINED)
- RequestDate

How To Use

- **Register:**
    - Go to index.html
    - Select the register button
    - Fill username, email, password, and starting balance
    - A random card number is generated to represent a bank card
- **Log In:**
    - Use index.html
    - Enter username and password
    - On success, sent to history.html
- **View History:**
    - history.html shows your username, current balance, transaction history
    - Navigation buttons to search users, add funds, or view pending requests
        - The button to direct to pending requests only appears if the logged in user actually has requests pending
- **Send or Request Money:**
    - Use transaction.html
    - Enter recipient user ID (autofilled from search or request)
    - Enter amount, choose to send/request
    - Success redirects to history.html

- **Search Users:**
  - Use search.html
  - Search by username or email
  - If user found, redirected to transaction.html with user ID filled
- **Manage Requests:**
  - Go to pending.html
  - See pending requests for you
  - Accept: Goes to transaction form (autofills details)
    - If funds insufficient, status remains pending, alert shown
  - Decline: Removes request
- **Add Money:**
  - Go to account.html
  - Enter amount (shows logged in user's card number)
  - Money added after 2-day delay
- **Log Out:**
  - Use "Back to Login" button (calls logout.php)
  - Session destroyed
  - Pressing back reloads and checks session; if not logged in, alert and redirect back to index

Security/Validation Notes

- Duplicate username/email checks on registration.
- Passwords are hashed.
- Prepared statements for SQL in php scripts.
- Session required for all user actions.
- If user not logged in, all backend scripts redirect to login.

- "Back" button after logout reloads and checks session status.
- User cannot access others' data by changing user IDs.

How to Install/Run

- Import SQL schema to your MySQL server.
- Edit db.php for your DB credentials.
- Upload all files to your PHP web server. (EC2)
- Open register.html in your browser to create the first user.
- App is now ready to use.

Troubleshooting

- **Duplicate registration:** Usernames/emails must be unique.
- **Session issues:** Clear browser cookies or restart server.
- **Page not updating after logout:** Make sure session checks are in place on all pages.
- **Database errors:** Ensure all tables/columns exist as shown above.