

# TRƯỜNG ĐẠI HỌC PHENIKAA KHOA CÔNG NGHỆ THÔNG TIN

## --------------------o0o--------------------

BÀI TẬP LỚN MÔN KỸ THUẬT PHẦN MỀM

ĐỀ TÀI: QUẢN LÝ HŨ TÀI CHÍNH

Lớp : Kỹ thuật phần mềm (COUR01)

Giảng viên hướng dẫn:

Nhóm sinh viên thực hiện:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Lớp | Điểm (bằng  số) | Điểm (bằng  chữ) |
| 1 | Phạm Lê Đình An |  | K17 – CNTT7 |  |  |
| 2 | Hà Lê Quang Minh | 23010708 | K17 – CNTT7 |  |  |
| 3 | Nguyễn Duy Anh Tuấn | 23010709 | K17 – CNTT7 |  |  |  |
| 4 | Nguyễn Đức Tuấn |  | K17 – CNTT7 |  |  |

***Hà Nội, Tháng 6 năm 2024***

**BẢNG BÁO CÁO PHÂN CÔNG CÔNG VIỆC**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Họ và tên | MSSV | Công việc | Tiến độ |
| 1 | Phạm Lê Đình An |  |  | Hoàn thành |
| 2 | Hà Lê Quang Minh | 23010708 |  | Hoàn thành |
| 3 | Nguyễn Duy Anh Tuấn |  |  | Hoàn thành |
| 4 | Nguyễn Đức Tuấn |  |  |  |

### ****Mục lục****

* **Chương 1: Giới thiệu đề tài**
  + 1.1. Đặt vấn đề
  + 1.2. Mục tiêu và phạm vi đề tài
  + 1.3. Định hướng giải pháp
* **Chương 2: Giới thiệu nền tảng/Công nghệ sử dụng**
  + 2.1. Khái quát về PHP và MySQL
  + 2.2. Khái quát về JavaScript, HTML, CSS
  + 2.3. Khái quát về Chart.js
* **Chương 3: Kiến trúc Hệ thống**
  + 3.1. Kiến trúc Client-Server
  + 3.2. Cấu trúc thư mục dự án
  + 3.3. Thiết kế Cơ sở dữ liệu
* **Chương 4: Triển khai các chức năng chính**
  + 4.1. Xác thực người dùng (Đăng ký, Đăng nhập)
  + 4.2. Quản lý Thu nhập và Tự động phân bổ hũ
  + 4.3. Quản lý Chi tiêu và Kiểm tra số dư
  + 4.4. Trực quan hóa dữ liệu với Chart.js
* **Chương 5: Xây dựng Test Plan và Kiểm thử**
  + 5.1. Test Plan
  + 5.2. Test Cases
* **Chương 6: Kết luận và hướng phát triển**
  + 6.1. Kết luận
  + 6.2. Hướng phát triển
* **TÀI LIỆU THAM KHẢO**

### ****BÁO CÁO TỔNG KẾT DỰ ÁN****

**Tên dự án: JFINs - Ứng dụng Quản lý Chi tiêu 6 Hũ**

### ****1. Mục Tiêu Dự Án****

* **Xây dựng ứng dụng web hoàn chỉnh:** Phát triển một sản phẩm có thể truy cập trên trình duyệt, hỗ trợ người dùng quản lý tài chính cá nhân.
* **Áp dụng phương pháp 6 hũ:** Tự động hóa việc phân bổ thu nhập vào 6 hũ theo tỷ lệ do người dùng thiết lập, bao gồm:
  + Nhu cầu thiết yếu (NEC): 55%
  + Tự do tài chính (FFA): 10%
  + Giáo dục (EDU): 10%
  + Tiết kiệm dài hạn (LTSS): 10%
  + Hưởng thụ (PLAY): 10%
  + Cho đi (GIVE): 5%
* **Cung cấp các chức năng cốt lõi:** Bao gồm quản lý người dùng (đăng ký, đăng nhập), quản lý thu nhập, quản lý chi tiêu và tùy chỉnh cài đặt.
* **Trực quan hóa dữ liệu:** Sử dụng biểu đồ để người dùng dễ dàng nắm bắt được tình hình phân bổ tài chính và so sánh giữa thu nhập và chi tiêu.

### ****2. Công Nghệ Sử Dụng****

Dự án được xây dựng theo kiến trúc Client-Server, tận dụng các công nghệ hiện đại để đảm bảo tính ổn định, bảo mật và trải nghiệm người dùng tốt.

* **Backend:**
  + **Ngôn ngữ:** PHP.
  + **Cơ sở dữ liệu:** MySQL.
  + **Mô tả:** Phía server xử lý toàn bộ logic nghiệp vụ, bao gồm xác thực người dùng, các thao tác CRUD (Thêm, Đọc, Sửa, Xóa) với dữ liệu thu nhập/chi tiêu, và các tính toán phân bổ tài chính. Kết nối đến cơ sở dữ liệu được thực hiện thông qua PDO để tăng cường bảo mật.
* **Frontend:**
  + **Ngôn ngữ:** HTML, CSS, JavaScript.
  + **Thư viện:** Chart.js được sử dụng để vẽ các biểu đồ tròn (doughnut) và biểu đồ cột (bar) một cách sinh động.
  + **Mô tả:** Phía client xây dựng giao diện người dùng (UI) và trải nghiệm người dùng (UX). Toàn bộ tương tác của người dùng như thêm, sửa, xóa dữ liệu đều được thực hiện thông qua các lệnh gọi API bất đồng bộ (Fetch API) tới backend, giúp trang không cần tải lại và mang lại cảm giác mượt mà như một ứng dụng đơn trang (Single Page Application).

### ****3. Phân Tích Chức Năng****

Ứng dụng cung cấp đầy đủ các chức năng để quản lý tài chính cá nhân theo phương pháp 6 hũ:

* **1. Quản lý Người dùng:**
  + **Đăng ký:** Người dùng có thể tạo tài khoản mới. Khi đăng ký thành công, hệ thống tự động tạo ra các cài đặt phân bổ hũ mặc định cho người dùng đó.
  + **Đăng nhập/Đăng xuất:** Hệ thống xác thực thông tin người dùng với dữ liệu được lưu trong CSDL. Mật khẩu được mã hóa (hashing) để đảm bảo an toàn.
  + **Đổi mật khẩu:** Người dùng đã đăng nhập có thể thay đổi mật khẩu của mình.
* **2. Bảng điều khiển (Dashboard):**
  + Hiển thị tổng quan số dư, tổng thu nhập và tổng chi tiêu.
  + Biểu đồ tròn (Doughnut Chart) thể hiện tỷ lệ phân bổ và số tiền thực tế trong từng hũ.
  + Biểu đồ cột (Bar Chart) so sánh trực quan giữa tổng thu nhập và tổng chi tiêu.
  + Hiển thị chi tiết số dư của 6 hũ tài chính.
* **3. Quản lý Thu nhập:**
  + **Thêm thu nhập:** Người dùng có thể ghi lại các khoản thu nhập mới.
  + **Tự động phân bổ:** Khi một khoản thu nhập mới được thêm vào, hệ thống sẽ tự động tính toán và cộng số tiền tương ứng vào số dư của 6 hũ dựa trên tỷ lệ đã được người dùng cài đặt.
  + **Xem, sửa, xóa thu nhập:** Hiển thị danh sách các khoản thu nhập đã ghi lại và cho phép người dùng chỉnh sửa hoặc xóa chúng.
* **4. Quản lý Chi tiêu:**
  + **Thêm chi tiêu:** Người dùng có thể thêm một khoản chi mới bằng cách chọn một trong 6 hũ để trừ tiền.
  + **Kiểm tra số dư:** Hệ thống sẽ kiểm tra số dư của hũ được chọn. Nếu không đủ, giao dịch sẽ không được thực hiện.
  + **Cập nhật số dư:** Sau khi chi tiêu thành công, số dư của hũ tương ứng sẽ được cập nhật.
  + **Xem, sửa, xóa chi tiêu:** Hiển thị danh sách và cho phép quản lý các khoản chi tiêu đã có.
* **5. Cài đặt:**
  + **Tùy chỉnh tỷ lệ hũ:** Cho phép người dùng thay đổi tỷ lệ phần trăm (%) phân bổ cho 6 hũ tùy theo nhu cầu cá nhân.
  + **Xóa dữ liệu:** Cung cấp một "vùng nguy hiểm" cho phép người dùng xóa toàn bộ dữ liệu cá nhân (thu nhập, chi tiêu, cài đặt) sau khi xác nhận.

### ****4. Cấu Trúc Cơ Sở Dữ Liệu****

Dự án sử dụng MySQL làm hệ quản trị cơ sở dữ liệu. Dữ liệu được tổ chức qua các bảng chính sau:

* users: Lưu thông tin tài khoản của người dùng (email, mật khẩu đã mã hóa, username).
* jars: Lưu thông tin định nghĩa về 6 hũ (ví dụ: 'Thiết Yếu', 'Giáo Dục').
* jar\_allocations: Lưu trữ tỷ lệ phân bổ (%) của mỗi hũ cho từng người dùng.
* incomes: Ghi lại lịch sử các khoản thu nhập của người dùng.
* expenses: Ghi lại lịch sử các khoản chi tiêu, mỗi khoản chi được liên kết với một hũ cụ thể.
* jar\_balances: Lưu số dư tiền hiện tại của mỗi hũ cho từng người dùng. Bảng này được cập nhật mỗi khi có giao dịch thu nhập hoặc chi tiêu, giúp truy vấn số dư nhanh chóng.

(Chi tiết các trường và mối quan hệ được định nghĩa trong file *jfins/sql/create\_database.sql*)

### ****5. Đánh Giá và Hướng Phát Triển****

* **Ưu điểm:**
  + **Hoàn thiện và đúng mục tiêu:** Dự án đã xây dựng thành công một ứng dụng hoạt động tốt, đáp ứng đầy đủ các mục tiêu đề ra ban đầu.
  + **Kiến trúc vững chắc:** Việc sử dụng mô hình Client-Server với PHP/MySQL thay vì chỉ dùng localStorage (như trong một số tài liệu kế hoạch ban đầu) giúp ứng dụng có khả năng mở rộng, bảo mật và lưu trữ dữ liệu bền vững.
  + **Trải nghiệm người dùng tốt:** Giao diện sạch sẽ, dễ sử dụng, cùng với việc áp dụng AJAX giúp các thao tác diễn ra mượt mà, không cần tải lại trang.
  + **Tính thực tiễn cao:** Cung cấp một công cụ hữu ích và thiết thực cho việc quản lý tài chính cá nhân.
* **Hướng phát triển trong tương lai:**
  + **Phát triển phiên bản di động:** Xây dựng ứng dụng cho nền tảng di động (Android, iOS) để tăng tính tiện lợi.
  + **Nâng cao tính năng báo cáo:** Thêm các bộ lọc nâng cao (lọc theo khoảng thời gian tùy chỉnh, theo danh mục), và xuất báo cáo ra các định dạng như PDF, Excel.
  + **Đặt mục tiêu tài chính:** Cho phép người dùng đặt ra các mục tiêu tiết kiệm (ví dụ: mua nhà, mua xe) và theo dõi tiến độ thực hiện.
  + **Giao dịch định kỳ:** Thêm chức năng tự động ghi lại các khoản thu/chi định kỳ hàng tháng (ví dụ: lương, tiền thuê nhà).

### ****7. Kết Luận****

Dự án "JFINs - App Quản lý Chi tiêu 6 Hũ" đã được hoàn thành xuất sắc, không chỉ đáp ứng mà còn vượt qua các yêu cầu ban đầu bằng việc xây dựng một hệ thống backend mạnh mẽ. Sản phẩm cuối cùng là một công cụ quản lý tài chính cá nhân toàn diện, trực quan và có tính ứng dụng cao, thể hiện được năng lực và sự đầu tư nghiêm túc của nhóm phát triển.

tôi cần viết chi tiết hơn và bạn phải đưa code vào các nới chức năng chính kèm biều đồ như biểu đồ không vẽ được thì phải note đó cho tôi tự vẽ .còn các thứ khác như giới thiệu tính năng v.v.v. một báo cáo thật chi tiết

Tất nhiên, tôi sẽ soạn thảo một báo cáo chi tiết và chuyên sâu hơn cho dự án "JFINs - App Quản lý Chi tiêu 6 Hũ", bao gồm cả các đoạn mã nguồn minh họa cho các chức năng chính như bạn yêu cầu.

### ****BÁO CÁO CHI TIẾT DỰ ÁN****

**Đề tài: Xây Dựng Ứng Dụng Quản Lý Tài Chính Cá Nhân JFINs Dựa Trên Phương Pháp 6 Hũ**

### ****CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN****

#### **1.1. Bối cảnh và Lý do chọn đề tài**

Trong bối cảnh kinh tế hiện đại, việc quản lý tài chính cá nhân trở thành một kỹ năng thiết yếu đối với mọi người. Tuy nhiên, nhiều người gặp khó khăn trong việc theo dõi các dòng tiền, phân bổ thu nhập một cách hợp lý và đạt được các mục tiêu tài chính dài hạn.

Phương pháp "6 Hũ Tài Chính" (6 Jars Money Management System) của T. Harv Eker là một hệ thống quản lý tiền bạc đơn giản nhưng cực kỳ hiệu quả. Bằng cách chia thu nhập thành 6 quỹ riêng biệt, người dùng có thể đảm bảo mọi khía cạnh của cuộc sống tài chính đều được quan tâm, từ nhu cầu thiết yếu đến đầu tư, giáo dục, hưởng thụ và từ thiện.

Nhận thấy tiềm năng và tính ứng dụng cao của phương pháp này, dự án "JFINs" được ra đời nhằm mục đích số hóa và tự động hóa quy trình quản lý 6 hũ, mang đến một công cụ trực quan, dễ tiếp cận giúp người dùng kiểm soát tài chính cá nhân một cách hiệu quả.

#### **1.2. Mục tiêu của dự án**

* **Mục tiêu tổng quát:** Xây dựng một ứng dụng web hoàn chỉnh, ổn định và thân thiện với người dùng để quản lý tài chính cá nhân.
* **Mục tiêu cụ thể:**
  + Triển khai đầy đủ các chức năng quản lý người dùng: Đăng ký, Đăng nhập, Đăng xuất và Thay đổi mật khẩu.
  + Xây dựng module Quản lý Thu nhập cho phép người dùng thêm, sửa, xóa các khoản thu và tự động phân bổ tiền vào 6 hũ.
  + Xây dựng module Quản lý Chi tiêu cho phép ghi lại các khoản chi từ một hũ cụ thể, với cơ chế kiểm tra số dư nghiêm ngặt.
  + Trực quan hóa dữ liệu tài chính thông qua Bảng điều khiển (Dashboard) với các biểu đồ và số liệu thống kê.
  + Cho phép người dùng tùy chỉnh tỷ lệ phân bổ của 6 hũ và cung cấp tùy chọn xóa toàn bộ dữ liệu.

#### **1.3. Đối tượng và Phạm vi áp dụng**

* **Đối tượng:** Mọi cá nhân có nhu cầu quản lý tài chính, đặc biệt là những người mới bắt đầu tìm hiểu về lập kế hoạch chi tiêu và tiết kiệm.
* **Phạm vi:** Ứng dụng hoạt động trên nền tảng web, có thể truy cập bằng các trình duyệt phổ biến. Phiên bản hiện tại tập trung vào quản lý tài chính cho một người dùng duy nhất.

### ****CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG****

#### **2.1. Công nghệ sử dụng**

Dự án được phát triển dựa trên kiến trúc Client-Server với các công nghệ được lựa chọn cẩn thận:

* **Backend (Phía máy chủ):**
  + **Ngôn ngữ PHP:** Được chọn vì sự phổ biến, cộng đồng hỗ trợ lớn và dễ dàng tích hợp với các hệ quản trị CSDL như MySQL. PHP chịu trách nhiệm xử lý toàn bộ logic nghiệp vụ của ứng dụng.
  + **Cơ sở dữ liệu MySQL:** Một hệ quản trị CSDL quan hệ mạnh mẽ, ổn định, phù hợp để lưu trữ dữ liệu có cấu trúc như thông tin người dùng, các giao dịch thu chi.
  + **PDO (PHP Data Objects):** Được sử dụng để tạo kết nối và tương tác với CSDL MySQL, giúp ngăn chặn các lỗi bảo mật phổ biến như SQL Injection.
* **Frontend (Phía người dùng):**
  + **HTML5, CSS3:** Xây dựng cấu trúc và định hình giao diện của ứng dụng.
  + **JavaScript (ES6+):** Xử lý các tương tác của người dùng, tạo ra trải nghiệm động và mượt mà.
  + **Fetch API (AJAX):** Dùng để gửi và nhận dữ liệu từ backend một cách bất đồng bộ, giúp các thao tác không cần tải lại toàn bộ trang.
  + **Chart.js:** Thư viện JavaScript mạnh mẽ được dùng để vẽ các biểu đồ tài chính, giúp trực quan hóa dữ liệu.

#### **2.2. Thiết kế Cơ sở dữ liệu**

Cơ sở dữ liệu là xương sống của ứng dụng, được thiết kế để đảm bảo tính toàn vẹn và hiệu quả. Dưới đây là mô tả các bảng chính:

* users: Lưu trữ thông tin đăng nhập của người dùng.
  + user\_id: Khóa chính, tự tăng.
  + email: Email đăng nhập, duy nhất.
  + password: Mật khẩu đã được mã hóa.
  + username: Tên hiển thị của người dùng.
* jars: Lưu thông tin về 6 loại hũ.
  + jar\_id: Khóa chính.
  + name: Tên của hũ (ví dụ: 'Thiết Yếu').
* jar\_allocations: Lưu tỷ lệ phân bổ của người dùng cho mỗi hũ.
  + allocation\_id: Khóa chính.
  + user\_id: Khóa ngoại, liên kết đến bảng users.
  + jar\_id: Khóa ngoại, liên kết đến bảng jars.
  + percentage: Tỷ lệ % người dùng cài đặt.
* incomes: Ghi lại lịch sử các khoản thu nhập.
  + income\_id: Khóa chính.
  + user\_id: Khóa ngoại.
  + amount: Số tiền.
  + description: Mô tả.
  + income\_date: Ngày nhận.
* expenses: Ghi lại lịch sử các khoản chi tiêu.
  + expense\_id: Khóa chính.
  + user\_id: Khóa ngoại.
  + jar\_id: Khóa ngoại, chỉ định chi tiêu từ hũ nào.
  + amount, description, expense\_date.
* jar\_balances: Bảng quan trọng lưu số dư hiện tại của mỗi hũ cho mỗi người dùng.
  + balance\_id: Khóa chính.
  + user\_id, jar\_id: Khóa ngoại.
  + balance: Số tiền còn lại trong hũ.

### ****CHƯƠG 3: TRIỂN KHAI CÁC CHỨC NĂNG CHÍNH****

#### **3.1. Chức năng Xác thực người dùng (Authentication)**

**a. Đăng ký (Sign-up)**

* **Luồng hoạt động:** Người dùng điền thông tin vào form đăng ký. JavaScript phía client sẽ gửi dữ liệu này lên server. Server PHP kiểm tra tính hợp lệ (email đã tồn tại chưa, mật khẩu có khớp không), sau đó mã hóa mật khẩu và lưu người dùng mới vào CSDL. Đồng thời, hệ thống sẽ tạo các bản ghi phân bổ hũ mặc định cho người dùng này.
* **Mã nguồn phía Client (JavaScript - jfins/assets/js/login.js):**

JavaScript

A screen shot of a computer program

Description automatically generatedA screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

* **Mã nguồn phía Server (PHP - jfins/api/auth/signin.php):**

PHP

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

**b. Đăng nhập (Login)**

* **Luồng hoạt động:** Người dùng nhập email và mật khẩu. Server sẽ tìm người dùng bằng email, sau đó so sánh mật khẩu đã mã hóa trong CSDL với mật khẩu người dùng nhập vào. Nếu khớp, server tạo một phiên làm việc (session) và trả về kết quả thành công.
* **Mã nguồn phía Server (PHP - jfins/api/auth/login.php):**

PHP

A computer screen with many colorful text

Description automatically generated

#### **3.2. Chức năng Quản lý Thu nhập (Income Management)**

* **Luồng hoạt động:** Khi người dùng thêm một khoản thu nhập, backend không chỉ lưu giao dịch này vào bảng incomes mà còn phải thực hiện logic quan trọng: Dựa trên tỷ lệ phân bổ đã cài đặt trong jar\_allocations, hệ thống tính toán số tiền tương ứng cho mỗi hũ và cập nhật (hoặc tạo mới) số dư trong bảng jar\_balances.
* **Mã nguồn phía Server (PHP - jfins/api/income/add\_income.php):**

PHP

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

#### **3.3. Chức năng Quản lý Chi tiêu (Expense Management)**

* **Luồng hoạt động:** Đây là chức năng đòi hỏi sự cẩn trọng. Khi người dùng ghi một khoản chi, backend phải:
  1. Kiểm tra số dư của hũ được chọn trong bảng jar\_balances.
  2. Nếu số dư không đủ, trả về lỗi.
  3. Nếu đủ, tiến hành ghi giao dịch vào bảng expenses VÀ trừ số tiền tương ứng khỏi bảng jar\_balances. Các thao tác này được đặt trong một transaction để đảm bảo tính nhất quán của dữ liệu.
* **Mã nguồn phía Server (PHP - jfins/api/expense/add\_expense.php):**

PHP

<?php

// ...

try {

// Bắt đầu một transaction

$conn->beginTransaction();

// 1. Kiểm tra số dư của hũ

$check\_balance\_sql = "SELECT balance FROM jar\_balances WHERE user\_id = :user\_id AND jar\_id = :jar\_id FOR UPDATE";

$check\_balance\_stmt = $conn->prepare($check\_balance\_sql);

$check\_balance\_stmt->bindParam(':user\_id', $user\_id);

$check\_balance\_stmt->bindParam(':jar\_id', $jar\_id);

$check\_balance\_stmt->execute();

$current\_balance = $check\_balance\_stmt->fetchColumn();

// 2. Nếu số dư không đủ, ném ra ngoại lệ

if ($current\_balance < $amount) {

throw new Exception('Số dư trong hũ không đủ!');

}

// 3. Thêm chi tiêu vào bảng expenses

$sql = "INSERT INTO expenses (user\_id, expense\_date, amount, description, jar\_id) VALUES (:user\_id, :expense\_date, :amount, :description, :jar\_id)";

$stmt = $conn->prepare($sql);

// ... (bind params và execute)

if($stmt->execute()) {

// 4. Cập nhật (trừ tiền) số dư của hũ

$update\_balance\_sql = "UPDATE jar\_balances SET balance = balance - :amount WHERE user\_id = :user\_id AND jar\_id = :jar\_id";

$update\_balance\_stmt = $conn->prepare($update\_balance\_sql);

// ... (bind params và execute)

// Nếu mọi thứ thành công, commit transaction

$conn->commit();

$response['success'] = true;

$response['message'] = 'Thêm chi tiêu thành công';

} else {

throw new Exception('Có lỗi khi thêm chi tiêu');

}

} catch (Exception $e) {

// Nếu có bất kỳ lỗi nào, rollback transaction

$conn->rollBack();

$response['message'] = 'Có lỗi xảy ra: '. $e->getMessage();

}

// ...

?>

#### **3.4. Bảng điều khiển và Trực quan hóa dữ liệu (Dashboard & Visualization)**

* **Mô tả:** Dashboard là trung tâm của ứng dụng, nơi dữ liệu được tổng hợp và hiển thị một cách trực quan. Chức năng này sử dụng thư viện Chart.js để vẽ biểu đồ.
* **Biểu đồ Phân bổ Hũ (Doughnut Chart):** Hiển thị tỷ lệ % và số tiền thực tế trong mỗi hũ.
  + ***[LƯU Ý: Đây là mã nguồn cấu hình biểu đồ bằng thư viện Chart.js. Cần được thực thi trong môi trường JavaScript để hiển thị biểu đồ thực tế.]***
  + **Mã nguồn (jfins/assets/js/chart.js):**

JavaScript

const pieChart = new Chart(ctx, {

type: 'doughnut',

data: {

labels: ['Thiết Yếu', 'Tự Do Tài Chính', 'Giáo Dục', 'Hưởng Thụ', 'Thiện Tâm', 'Tiết Kiệm'],

datasets: [{

data: [55, 10, 10, 10, 5, 10], // Dữ liệu % sẽ được cập nhật động

backgroundColor: [

'#FF6384', '#36A2EB', '#FFCE56',

'#4BC0C0', '#9966FF', '#FF9F40'

],

// ...

}]

},

options: {

responsive: true,

plugins: {

title: {

display: true,

text: 'Phân bố hũ',

},

tooltip: {

callbacks: {

label: function(context) {

// Hiển thị cả % và số tiền

const label = context.label || '';

const value = context.raw || 0;

const amount = jarData.amounts[context.dataIndex]; // Lấy số tiền thực tế

return [

`${label}: ${value}%`,

`Số tiền: ${amount.toLocaleString('vi-VN')} đ`

];

}

}

}

},

// ...

}

});

* **Biểu đồ Thu nhập và Chi tiêu (Bar Chart):** So sánh tổng thu nhập và tổng chi tiêu trong một khoảng thời gian.
  + ***[LƯU Ý: Đây là mã nguồn cấu hình biểu đồ bằng thư viện Chart.js. Cần được thực thi trong môi trường JavaScript để hiển thị biểu đồ thực tế.]***
  + **Mã nguồn (jfins/assets/js/chart.js):**

JavaScript

const barChart = new Chart(barCtx, {

type: 'bar',

data: {

labels: ['Thu nhập', 'Chi tiêu'],

datasets: [{

data: [0, 0], // Dữ liệu sẽ được cập nhật động

backgroundColor: [

'#36A2EB', // Màu cho thu nhập

'#FF6384' // Màu cho chi tiêu

],

// ...

}]

},

options: {

plugins: {

title: {

display: true,

text: 'Thu nhập và Chi tiêu',

},

legend: {

display: false // Không cần legend cho biểu đồ này

},

// ...

},

scales: {

y: {

beginAtZero: true,

ticks: {

callback: function(value) {

return value.toLocaleString('vi-VN') + ' đ'; // Định dạng tiền tệ

}

}

}

}

}

});

### ****CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN****

#### **4.1. Kết quả đạt được**

Dự án đã thành công trong việc xây dựng một ứng dụng web quản lý tài chính cá nhân hoàn chỉnh và hoạt động ổn định.

* Hệ thống đã triển khai thành công logic cốt lõi của phương pháp 6 hũ, từ việc phân bổ thu nhập tự động đến việc kiểm soát chi tiêu theo từng quỹ.
* Kiến trúc Client-Server với backend PHP và CSDL MySQL đảm bảo dữ liệu được lưu trữ an toàn, bền vững và có khả năng mở rộng.
* Giao diện người dùng trực quan, dễ sử dụng, kết hợp với các biểu đồ phân tích giúp người dùng có cái nhìn rõ ràng về tình hình tài chính của mình.

#### **4.2. Hạn chế**

* **Báo cáo và thống kê:** Chức năng báo cáo còn khá cơ bản, chưa cho phép người dùng tùy chỉnh sâu hoặc xuất dữ liệu.
* **Thiếu tính năng di động:** Ứng dụng hiện chỉ có phiên bản web, chưa có ứng dụng riêng cho di động.
* **Tài liệu chưa đồng bộ:** Một số tài liệu kế hoạch ban đầu (ví dụ: Tong\_hop\_App\_Quan\_Ly\_Chi\_Tieu\_6\_Hu.docx) đề cập đến việc sử dụng localStorage, trong khi phiên bản cuối cùng đã sử dụng CSDL, cho thấy sự thay đổi tích cực trong quá trình phát triển nhưng cần cập nhật lại tài liệu.

#### **4.3. Hướng phát triển**

* **Phát triển ứng dụng di động (Mobile App):** Xây dựng phiên bản cho iOS và Android để người dùng có thể quản lý tài chính mọi lúc, mọi nơi.
* **Nâng cấp hệ thống Báo cáo & Thống kê:** Thêm các bộ lọc nâng cao, so sánh các kỳ tài chính, và cho phép xuất báo cáo ra file PDF/Excel.
* **Thêm tính năng Đặt mục tiêu (Financial Goals):** Cho phép người dùng tạo các mục tiêu tiết kiệm (mua nhà, du lịch,...) và theo dõi tiến độ hoàn thành.
* **Tích hợp Giao dịch định kỳ:** Tự động hóa việc ghi nhận các khoản thu/chi lặp lại hàng tháng như lương, tiền thuê nhà, hóa đơn.

### ****Chương 1: Giới thiệu đề tài****

#### **1.1. Đặt vấn đề**

Trong cuộc sống hiện đại, quản lý tài chính cá nhân là một kỹ năng quan trọng nhưng thường bị bỏ qua. Việc thiếu một kế hoạch chi tiêu rõ ràng dẫn đến tình trạng bội chi, không có khoản tiết kiệm dự phòng và khó đạt được các mục tiêu tài chính lớn.

Phương pháp "6 Hũ Tài Chính" cung cấp một khung làm việc đơn giản và hiệu quả để phân bổ thu nhập, đảm bảo mọi khía cạnh tài chính đều được cân bằng. Tuy nhiên, việc tính toán và theo dõi thủ công có thể tốn thời gian và dễ xảy ra sai sót. Do đó, việc phát triển một ứng dụng web để tự động hóa quy trình này là một vấn đề cấp thiết, giúp người dùng dễ dàng áp dụng phương pháp này vào thực tế.

#### **1.2. Mục tiêu và phạm vi đề tài**

* **Mục tiêu của đề tài:**
  + Xây dựng một ứng dụng web hoàn chỉnh có tên "JFINs" để quản lý tài chính cá nhân theo phương pháp 6 hũ.
  + Lập trình các chức năng cơ bản bao gồm đăng ký, đăng nhập, quản lý thu nhập và chi tiêu.
  + Tự động hóa việc phân bổ thu nhập vào 6 hũ theo tỷ lệ người dùng cài đặt.
  + Tạo giao diện người dùng thân thiện, trực quan, có biểu đồ thống kê để người dùng dễ dàng theo dõi.
  + Đảm bảo hệ thống hoạt động ổn định và dữ liệu được lưu trữ an toàn.
* **Phạm vi của đề tài:**
  + Tập trung vào việc triển khai các chức năng cốt lõi của việc quản lý thu chi theo phương pháp 6 hũ.
  + Ứng dụng hoạt động trên nền tảng web.
  + Không đi sâu vào các phân tích tài chính phức tạp hay đưa ra lời khuyên đầu tư.

#### **1.3. Định hướng giải pháp**

Để giải quyết vấn đề đặt ra, nhóm đã lựa chọn các giải pháp công nghệ sau:

* **Sử dụng kiến trúc Client-Server:** Tách biệt phần giao diện người dùng (Frontend) và phần xử lý logic (Backend) để dễ dàng phát triển, bảo trì và mở rộng.
* **Lưu trữ dữ liệu trên CSDL MySQL:** Đảm bảo dữ liệu người dùng được lưu trữ bền vững, an toàn và có thể truy xuất hiệu quả.
* **Tương tác động bằng JavaScript (AJAX):** Mọi thao tác thêm/sửa/xóa dữ liệu đều được xử lý ngầm, mang lại trải nghiệm mượt mà không cần tải lại trang.
* **Kiểm thử liên tục:** Áp dụng quy trình kiểm thử các chức năng trong suốt quá trình phát triển để đảm bảo chất lượng sản phẩm.

### ****Chương 2: Giới thiệu nền tảng/Công nghệ sử dụng****

* **PHP (Hypertext Preprocessor):** Là một ngôn ngữ kịch bản phía máy chủ mạnh mẽ, được sử dụng để xây dựng Backend cho ứng dụng. PHP chịu trách nhiệm xử lý các yêu cầu từ client, tương tác với CSDL và trả về dữ liệu.
* **MySQL:** Là hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến nhất thế giới, được sử dụng để lưu trữ toàn bộ dữ liệu của ứng dụng.
* **JavaScript, HTML, CSS:** Bộ ba công nghệ nền tảng để xây dựng Frontend. HTML tạo cấu trúc, CSS định dạng giao diện, và JavaScript xử lý các logic và tương tác phía người dùng.
* **Chart.js:** Là một thư viện JavaScript đơn giản và linh hoạt để vẽ các biểu đồ. Trong dự án này, Chart.js được dùng để tạo biểu đồ tròn và biểu đồ cột, giúp trực quan hóa dữ liệu tài chính.

### ****Chương 3: Kiến trúc Hệ thống****

#### **3.1. Cấu trúc thư mục dự án**

Dự án được tổ chức theo cấu trúc module rõ ràng, tách biệt các thành phần khác nhau:

* /api: Chứa các file PHP xử lý logic nghiệp vụ và đóng vai trò là API cho Frontend.
* /assets: Chứa các tài nguyên tĩnh như CSS, JavaScript, hình ảnh, icon.
* /config: Chứa file cấu hình kết nối CSDL.
* /page: Chứa các file PHP chính để hiển thị giao diện người dùng (trang đăng nhập, trang chính).
* /sql: Chứa file mã lệnh SQL để tạo cơ sở dữ liệu và các bảng.

#### **3.2. Thiết kế Cơ sở dữ liệu**

CSDL của JFINs được thiết kế để lưu trữ hiệu quả thông tin người dùng và các giao dịch tài chính.

* **Sơ đồ kiến trúc CSDL:** (LƯU Ý: Đây là mô tả cho sơ đồ. Bạn cần tự vẽ sơ đồ dựa trên mô tả này) Sơ đồ bao gồm các thực thể chính: users, jars, incomes, expenses. Bảng users có quan hệ một-nhiều với các bảng incomes, expenses. Bảng jar\_allocations và jar\_balances đóng vai trò là bảng liên kết, thể hiện mối quan hệ nhiều-nhiều giữa users và jars, lưu trữ tỷ lệ phân bổ và số dư cụ thể cho từng người dùng.
* **Mã nguồn tạo bảng (jfins/sql/create\_database.sql):**

SQL

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

### ****Chương 4: Triển khai các chức năng chính****

#### **4.1. Xác thực người dùng (Đăng ký, Đăng nhập)**

Chức năng này đảm bảo chỉ người dùng hợp lệ mới có thể truy cập vào dữ liệu tài chính của mình.

* **Đăng ký:**
  + **Mã nguồn Frontend (JavaScript - jfins/assets/js/login.js):** Đoạn mã này thu thập dữ liệu từ form và gửi yêu cầu POST đến API đăng ký.

JavaScript

document.getElementById('signupForm').addEventListener('submit', async function(event) {

event.preventDefault();

const formData = new FormData(this);

const response = await fetch('../../api/auth/signin.php', {

method: 'POST',

body: formData

});

const result = await response.json();

if (result.success) {

// Chuyển hướng hoặc hiển thị thông báo thành công

window.location.href = '../index.php';

} else {

// Hiển thị lỗi

alert(result.message);

}

});

* + **Mã nguồn Backend (PHP - jfins/api/auth/signin.php):** Xử lý yêu cầu, kiểm tra, mã hóa mật khẩu và lưu vào CSDL.

PHP

<?php

require\_once '../../config/db\_connect.php';

$username = $\_POST['username'];

$email = $\_POST['email'];

$password = $\_POST['password'];

// Kiểm tra email tồn tại

$check\_sql = "SELECT user\_id FROM users WHERE email = :email";

$check\_stmt = $conn->prepare($check\_sql);

$check\_stmt->execute(['email' => $email]);

if ($check\_stmt->rowCount() > 0) {

echo json\_encode(['success' => false, 'message' => 'Email đã được sử dụng']);

exit();

}

// Mã hóa mật khẩu

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

// Chèn người dùng mới

$sql = "INSERT INTO users (username, email, password) VALUES (:username, :email, :password)";

$stmt = $conn->prepare($sql);

$stmt->execute(['username' => $username, 'email' => $email, 'password' => $hashed\_password]);

// (Thêm logic tạo allocation mặc định...)

echo json\_encode(['success' => true, 'message' => 'Đăng ký thành công']);

?>

#### **4.2. Quản lý Thu nhập và Tự động phân bổ hũ**

Đây là chức năng cốt lõi của phương pháp 6 hũ. Khi một khoản thu nhập được thêm, nó sẽ được tự động chia vào các hũ.

* **Mã nguồn Backend (PHP - jfins/api/income/add\_income.php):**

PHP

<?php

session\_start();

require\_once '../../config/db\_connect.php';

// Lấy dữ liệu từ request

$data = json\_decode(file\_get\_contents('php://input'), true);

$user\_id = $\_SESSION['user\_id'];

$amount = $data['amount'];

$conn->beginTransaction();

try {

// 1. Thêm vào bảng incomes

$sql = "INSERT INTO incomes (user\_id, amount, description, income\_date) VALUES (?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->execute([$user\_id, $amount, $data['description'], $data['date']]);

// 2. Lấy tỷ lệ phân bổ của người dùng

$alloc\_sql = "SELECT jar\_id, percentage FROM jar\_allocations WHERE user\_id = ?";

$alloc\_stmt = $conn->prepare($alloc\_sql);

$alloc\_stmt->execute([$user\_id]);

$allocations = $alloc\_stmt->fetchAll();

// 3. Phân bổ vào các hũ

foreach ($allocations as $allocation) {

$jar\_amount = ($amount \* $allocation['percentage']) / 100;

// Cập nhật bảng jar\_balances

$update\_sql = "INSERT INTO jar\_balances (user\_id, jar\_id, balance) VALUES (?, ?, ?)

ON DUPLICATE KEY UPDATE balance = balance + VALUES(balance)";

$update\_stmt = $conn->prepare($update\_sql);

$update\_stmt->execute([$user\_id, $allocation['jar\_id'], $jar\_amount]);

}

$conn->commit();

echo json\_encode(['success' => true, 'message' => 'Thêm thu nhập thành công']);

} catch (Exception $e) {

$conn->rollBack();

echo json\_encode(['success' => false, 'message' => 'Lỗi: ' . $e->getMessage()]);

}

?>

#### **4.3. Quản lý Chi tiêu và Kiểm tra số dư**

Chức năng này cho phép người dùng chi tiêu từ một hũ cụ thể, đảm bảo không chi tiêu quá số tiền hiện có.

* **Mã nguồn Backend (PHP - jfins/api/expense/add\_expense.php):**

PHP

<?php

session\_start();

require\_once '../../config/db\_connect.php';

$data = json\_decode(file\_get\_contents('php://input'), true);

$user\_id = $\_SESSION['user\_id'];

$jar\_id = $data['jar\_id'];

$amount = $data['amount'];

$conn->beginTransaction();

try {

// 1. Kiểm tra số dư trong hũ

$balance\_sql = "SELECT balance FROM jar\_balances WHERE user\_id = ? AND jar\_id = ? FOR UPDATE";

$balance\_stmt = $conn->prepare($balance\_sql);

$balance\_stmt->execute([$user\_id, $jar\_id]);

$current\_balance = $balance\_stmt->fetchColumn();

if ($current\_balance === false || $current\_balance < $amount) {

throw new Exception('Số dư trong hũ không đủ để thực hiện giao dịch!');

}

// 2. Thêm vào bảng expenses

$sql = "INSERT INTO expenses (user\_id, jar\_id, amount, description, expense\_date) VALUES (?, ?, ?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->execute([$user\_id, $jar\_id, $amount, $data['description'], $data['date']]);

// 3. Trừ tiền khỏi hũ

$update\_sql = "UPDATE jar\_balances SET balance = balance - ? WHERE user\_id = ? AND jar\_id = ?";

$update\_stmt = $conn->prepare($update\_sql);

$update\_stmt->execute([$amount, $user\_id, $jar\_id]);

$conn->commit();

echo json\_encode(['success' => true, 'message' => 'Thêm chi tiêu thành công']);

} catch (Exception $e) {

$conn->rollBack();

echo json\_encode(['success' => false, 'message' => 'Lỗi: ' . $e->getMessage()]);

}

?>

#### **4.4. Trực quan hóa dữ liệu với Chart.js**

(LƯU Ý: Đây là mã nguồn cấu hình biểu đồ. Để hiển thị, cần có dữ liệu được truyền từ Backend và thực thi trong môi trường JavaScript.)

* **Biểu đồ tròn phân bổ các hũ (jfins/assets/js/chart.js):**

JavaScript

// Giả sử `jarData` là một object chứa dữ liệu lấy từ API

const jarData = {

labels: ['Thiết Yếu', 'Tự Do TC', 'Giáo Dục', 'Hưởng Thụ', 'Thiện Tâm', 'Tiết Kiệm'],

amounts: [5000000, 1000000, 1000000, 1000000, 500000, 1500000] // Dữ liệu động

};

const ctx = document.getElementById('pieChart').getContext('2d');

const pieChart = new Chart(ctx, {

type: 'doughnut',

data: {

labels: jarData.labels,

datasets: [{

label: 'Số dư các hũ',

data: jarData.amounts,

backgroundColor: [

'#FF6384', '#36A2EB', '#FFCE56',

'#4BC0C0', '#9966FF', '#FF9F40'

],

borderWidth: 1

}]

},

options: {

responsive: true,

plugins: {

legend: { position: 'top' },

title: { display: true, text: 'Tỷ lệ phân bổ các hũ' }

}

}

});

### ****Chương 5: Xây dựng Test Plan và Kiểm thử****

#### **5.1. Test Plan**

* **Mục đích:** Đảm bảo các chức năng của ứng dụng JFINs hoạt động chính xác, ổn định và đáp ứng đúng yêu cầu người dùng.
* **Phạm vi kiểm thử:**
  + **Kiểm thử chức năng (Functional Testing):** Kiểm tra các chức năng chính như đăng ký, đăng nhập, thêm/sửa/xóa thu nhập/chi tiêu.
  + **Kiểm thử giao diện người dùng (UI Testing):** Đảm bảo các thành phần hiển thị đúng, dễ sử dụng.
  + **Kiểm thử hiệu năng (Performance Testing):** Đánh giá tốc độ phản hồi của hệ thống khi có nhiều dữ liệu.
* **Công cụ:** Kiểm thử thủ công (manual testing) cho các kịch bản sử dụng và kiểm tra giao diện.

#### **5.2. Test Cases**

Dưới đây là một số trường hợp kiểm thử (test cases) tiêu biểu:

|  |  |  |  |
| --- | --- | --- | --- |
| **Danh mục chức năng** | **Test Case ID** | **Mô tả Test Case** | **Kết quả mong đợi** |
| **Đăng ký** | TC-REG-01 | Đăng ký với email đã tồn tại. | Hệ thống báo lỗi "Email đã được sử dụng". |
| **Thu nhập** | TC-INC-01 | Thêm một khoản thu nhập mới. | Giao dịch được lưu, số dư các hũ tăng lên theo đúng tỷ lệ. |
| **Chi tiêu** | TC-EXP-01 | Chi tiêu một khoản lớn hơn số dư trong hũ. | Hệ thống báo lỗi "Số dư không đủ" và không tạo giao dịch. |
| **Chi tiêu** | TC-EXP-02 | Chi tiêu một khoản hợp lệ. | Giao dịch được lưu, số dư hũ tương ứng giảm đi. |

Xuất sang Trang tính

### ****Chương 6: Kết luận và hướng phát triển****

#### **6.1. Kết luận**

Dự án đã thành công xây dựng ứng dụng web JFINs, một công cụ mạnh mẽ và hữu ích cho việc quản lý tài chính cá nhân theo phương pháp 6 hũ. Hệ thống đã đáp ứng đầy đủ các mục tiêu đề ra, với kiến trúc vững chắc, chức năng hoàn thiện và giao diện thân thiện. Quá trình thực hiện dự án đã giúp các thành viên trong nhóm nâng cao kỹ năng lập trình, làm việc nhóm và quản lý dự án.

#### **6.2. Hướng phát triển**

Để nâng cao giá trị của sản phẩm, nhóm đề xuất các hướng phát triển trong tương lai:

* **Thêm nhiều tính năng và báo cáo nâng cao:** Phát triển các báo cáo chi tiết theo tuần/tháng/năm, so sánh các kỳ và cho phép người dùng tùy chỉnh báo cáo.
* **Mở rộng đa nền tảng:** Phát triển phiên bản ứng dụng di động cho Android và iOS để tăng tính tiện dụng.
* **Chế độ "Gia đình":** Phát triển chế độ cho phép nhiều người dùng (ví dụ: các thành viên trong gia đình) cùng quản lý tài chính trên một tài khoản chung.
* **Cải thiện giao diện người dùng (UI/UX):** Cung cấp thêm các tùy chọn tùy chỉnh giao diện như theme (sáng/tối), và tối ưu hóa trải nghiệm trên các thiết bị khác nhau.
* **Tích hợp hệ thống thông báo:** Gửi thông báo nhắc nhở người dùng ghi lại chi tiêu hàng ngày hoặc cảnh báo khi sắp hết ngân sách trong một hũ nào đó.

### ****TÀI LIỆU THAM KHẢO****

* Tài liệu dự án JFINs (Mã nguồn và README.md).
* Godot Docs-4.2 branch. (n.d.). Retrieved from <https://docs.godotengine.org/en/stable/>
* Tài liệu tham khảo từ dự án "Trò chơi đặt bom vượt ải trên nền tảng Godot".

### 

# KẾT LUẬN

Như vậy với đề tài thiết kế ứng dụng quản lý bán hàng siêu thị, nhóm chúng em đã hiểu được cách hình thành, vận dụng lập trình hướng đối tượng để đưa ra được cách xây dựng một ứng dụng mang tính khả thi và thực tiễn cao. Báo cáo của chúng em đề cập đến nghiên cứu và triển khai một hệ thống quản lý bán hàng siêu thị, nhằm giúp quản lý hoạt động kinh doanh của siêu thị một cách hiệu quản hơn. Mục tiêu hàng đầu của chúng em là xây dựng một hệ thống an toàn, thông minh, giúp người dùng dễ dàng tiếp cận và sử dụng mà không gặp phải những rắc rối không đáng có...

Chúng em hi vọng rằng hệ thống sẽ đáp ứng được nhu cầu của cả người dùng và các siêu thị. Chân thành cảm ơn sự quan tâm và hỗ trợ từ cô Hà Thị Kim Dung đối với dự án của chúng em, và chúng em sẵn sàng tiếp tục phát triển và cải thiện hệ thống để mang lại lợi ích tốt nhất cho cộng đồng. Do hạn chế về mặt thời gian và hiểu biết, đề tài của chúng em có thể còn nhiều thiết sót, rất mong sẽ nhận được sự góp ý, bổ sung từ cô và các bạn để nhóm chúng em có thể hoàn thiện vốn kiến thức của mình, tạo hành trang vững chắc cho việc phát triển trong tương lai.

# TÀI LIỆU THAM KHẢO

[1]. PGS.TS. Đỗ Trung Tuấn,(2011), Đại học quốc gia Hà Nội, Chuyển đổi giữa Cơ sở dữ liệu quan hệ và XML.

[2]. XML là gì? Tìm hiểu về XML <https://topdev.vn/blog/xml-la-gi/> , truy cập ngày 15/03/2024.

[3]. VietNix(26/07/2022), Tìm hiểu mô hình MVC là gì? Ví dụ về cách sử dụng mô hình MVC <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/> , truy cập ngày 15/03/2024.