



Expense Tracker Web Application



Submitted By: Farheen Imam

Registration ID: 2312363

Instructors: Sir Abdul Ahad and Dr. Khalid Rasheed

Index

Index	2
1. Problem Statement.....	3
2. Objectives	3
3. Features & Functionalities.....	4
4. System Architecture	5
5. Database Design.....	6
6. File Structure & Connections.....	7
7. Page Descriptions	12
8. Validation Rules	16
9. Server-Side Validation (PHP)	17
10. Usage Guide	18

1. Problem Statement

Managing personal finances is a critical aspect of daily life, yet many individuals struggle to keep track of their income and expenses effectively. Traditional methods like paper-based tracking or basic spreadsheets are often:

- **Time-consuming** and prone to human error
- **Difficult to access** from multiple devices
- **Lack proper categorization** and organization
- **Missing validation** leading to inconsistent data entry
- **No user authentication** making data insecure

There was a need for a simple, web-based solution that allows users to:

- Securely manage their personal financial data
- Add, view, update, and delete income and expense records
- Maintain data integrity through proper validation

2. Objectives

Primary Objectives:

1. Develop a secure user authentication system with proper validation.
2. Create an intuitive interface for managing financial records.
3. Implement CRUD operations for both income and expenses.
4. Ensure data integrity through comprehensive validation rules.

Secondary Objectives:

1. Maintain clean code structure.
2. Implement user-friendly alerts.
3. Provide consistent navigation.
4. Ensure database security.
5. Deliver detailed project documentation.

3. Features & Functionalities

The Expense Tracker Web Application includes authentication, income and expense management, user profile handling, and a visually appealing interface. Users can register, log in, and manage their finances.

Authentication System

- **User Registration:** Secure signup with email validation
- **User Login:** Authentication with credential verification
- **Password Security:** 8-20 character validation with no spaces
- **Duplicate Prevention:** Email uniqueness validation
- **Session Management:** User ID tracking across sessions

Income Management

- **Add Income:** Record income with source, amount, and date
- **View Income:** Display all income records in tabular format
- **Search Income:** Search income records by source name
- **Update Income:** Edit existing income entries
- **Delete Income:** Remove income records with confirmation
- **Amount Validation:** Decimal support with min/max limits
- **Total Calculation:** Automatic total income calculation

Expense Management

- **Add Expenses:** Record expenses with description, category, amount, and date
- **View Expenses:** Display all expense records in organized tables
- **Search Expenses:** Search expense records by description
- **Update Expenses:** Modify existing expense entries
- **Delete Expenses:** Remove expense records safely
- **Description Validation:** 50-character limit with alphanumeric validation
- **Total Calculation:** Automatic total expenses calculation

User Profile

- **Profile View:** Display username and email information
- **Logout Functionality:** Logs the user out on click
- **User-specific Data:** All records are user-isolated

User Interface

- **Design:** Works on desktop
- **Gradient Backgrounds:** Modern visual appeal
- **Consistent Navigation:** Easy access to all features
- **Form Validation:** Real-time input validation
- **Alert System:** User-friendly success/error messages

4. System Architecture

Technology Stack

- **Frontend:** HTML5, CSS3, JavaScript
- **Backend:** PHP
- **Database:** MySQL
- **Server:** Apache
- **Styling:** Custom CSS with gradient effects

Architecture Pattern

The application follows a **simple MVC-like pattern**:

- **Models:** Database interactions through db.php
- **Views:** HTML templates with embedded PHP
- **Controllers:** PHP logic within each page file

Security Features

- **Input Validation:** Client-side and server-side validation
- **XSS Protection:** Direct output without sanitization
- **Session Security:** User ID-based access control

5. Database Design

The database consists of three main tables as shown below:

Tables Structure

1. Users Table

Purpose: Store user authentication and profile information

Column	Type	Key
Id	INT	Primary Key
name	VARCHAR	-
email	VARCHAR	-
password	VARCHAR	-

2. Income Table

Purpose: Store user income records

Column	Type	Key
i_id	INT	Primary Key
amount	DECIMAL	-
source	VARCHAR	-
date	DATE	-
user_id	INT	Foreign Key

3. Expenses Table

Purpose: Store user expense records

Column	Type	Key
e_id	INT	Primary Key
amount	DECIMAL	-
description	VARCHAR	-
category	VARCHAR	-
date	DATE	-
user_id	INT	Foreign Key

Database Relationships

- Users → Income (One user can have multiple income records)
- Users → Expenses (One user can have multiple expense records)
- **Data Isolation:** Each user can only access their own records

6. File Structure & Connections

Core Files

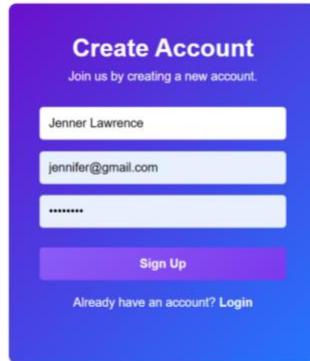
1. Database Connection

- **db.php:** Central database connection file
 - Establishes MySQL connection
 - Used by all other PHP files
 - Contains database credentials

2. Authentication Files

- **index.php:** User registration/signup page
 - Handles new user creation
 - Validates input data

- Redirects to login on success
- **login.php:** User authentication page
 - Validates user credentials
 - Starts user session
 - Redirects to expense form on success



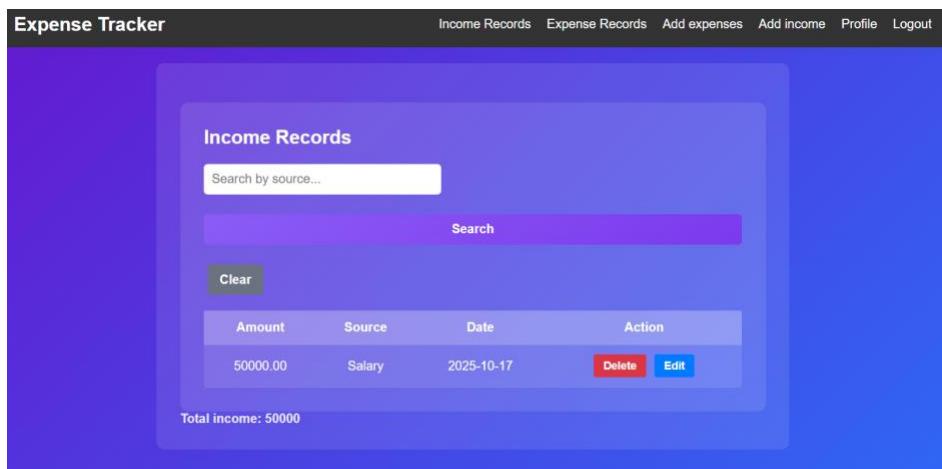
3. Navigation

- **navbar.php:** Shared navigation component
 - Displays consistent navigation across pages
 - Passes user ID to all links
 - Includes logout functionality

4. Income Management

- **income_form.php:** Add new income records
 - Form for income entry
 - Validates amount, source, and date
 - Redirects to list of income.php.
- **income.php:** Display income records
 - Lists all user income in table format
 - Provides edit and delete buttons
 - Shows user-specific data only

- **income_update.php:** Edit existing income
 - Pre-fills form with current data
 - Updates records in database
 - Validates modified data
- **income_delete.php:** Remove income records
 - Deletes specified income record
 - Redirects back to income list
 - Maintains user session



5. Expense Management

- **expenses_form.php:** Add new expense records

- Form for expense entry
- Validates amount, description, category, date
- Includes word count validation

- **expenses.php:** Display expense records

- Lists all user expenses in table format
- Provides edit and delete buttons
- Shows user-specific data only

- **expenses_update.php:** Edit existing expenses

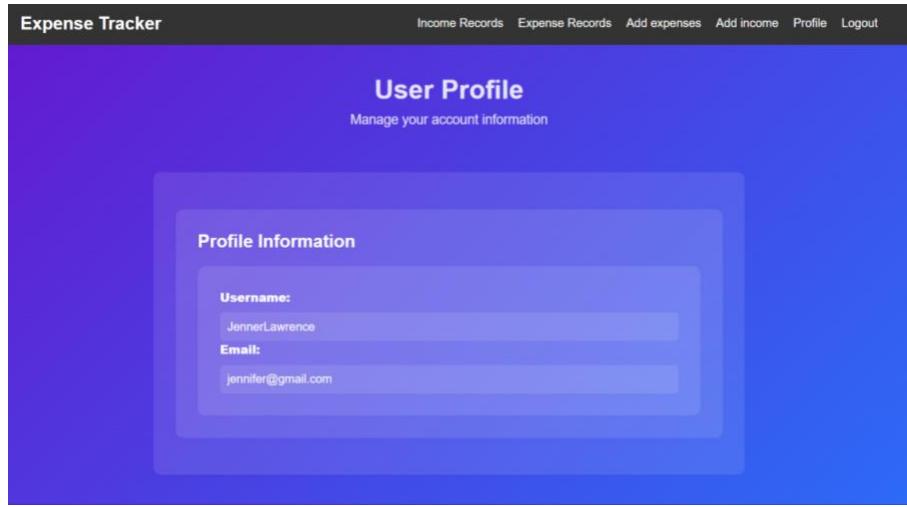
- Pre-fills form with current data

- Updates records in database
- Validates modified data
- **expenses_delete.php:** Remove expense records
 - Deletes specified expense record
 - Redirects back to expense list
 - Maintains user session

The screenshot shows a web-based expense tracker application. At the top, there's a navigation bar with links for 'Income Records', 'Expense Records', 'Add expenses', 'Add income', 'Profile', and 'Logout'. Below the navigation bar, a purple header section says 'Welcome to Your Expense Tracker' and provides instructions: 'Use the navigation bar to view or add records.' The main content area has a light blue background and contains a form titled 'Add Expenses'. The form fields are: 'Amount:' with an input field containing 'Enter amount', 'Description (max 50 characters):' with an input field containing 'Enter description', 'Category (max 15 characters):' with an input field containing 'Enter category', and 'Date:' with an input field containing 'mm/dd/yyyy'. At the bottom of the form is a purple button labeled 'Add Expense'. In the bottom right corner of the page, there's a watermark that says 'Activate Windows' and 'Go to Settings to activate Windows 10'.

6. User Profile

- **profile.php:** User profile management
 - Displays user information
 - Provides logout functionality
 - Shows username and email



7. Styling

- **style.css:** Application styling
 - Modern design
 - Gradient backgrounds
 - Form and table styling
 - Button and navigation styles

File Interconnections

index.php (Signup) → login.php → expenses_form.php (Dashboard)



navbar.php ↔ All authenticated pages ↔ profile.php



income_form.php ↔ income.php ↔ income_update.php



expenses_form.php ↔ expenses.php ↔ expenses_update.php



db.php (Used by all PHP files for database operations)

7. Page Descriptions

Authentication Pages

Index.php (Signup Page)

Purpose: New user registration **Features:**

- Username validation (3-20 characters, letters and spaces only)
- Email validation with uniqueness check
- Password validation (8-20 characters, no spaces)
- Duplicate email prevention with alerts
- Automatic redirect to login on success

User Flow:

1. User enters registration details
2. System validates input format
3. System checks for existing email
4. Creates new user account
5. Redirects to login page

Login.php

Purpose: User authentication **Features:**

- Email and password validation
- Credential verification against database
- Error alerts for invalid credentials
- Session initiation on successful login
- Redirect to main dashboard

User Flow:

1. User enters login credentials
2. System validates input format
3. System verifies credentials

4. Starts user session
5. Redirects to expense form (dashboard)

Main Application Pages

Expenses_form.php (Dashboard)

Purpose: Primary landing page and expense entry **Features:**

- Add new expense records
- Amount validation (0-999,999,999)
- Description validation (3-50 characters, alphanumeric)
- Category input with validation
- Date selection
- Success alerts on submission

User Flow:

1. User lands on dashboard after login
2. Fills expense form
3. System validates all inputs
4. Saves expense to database
5. Shows success alert

Income_form.php

Purpose: Income record entry **Features:**

- Add new income records
- Amount validation with decimal support
- Source description input
- Date selection
- Success alerts on submission

User Flow:

1. User navigates from navbar

2. Fills income form
3. System validates inputs
4. Saves income to database
5. Shows success alert

Expenses.php

Purpose: View and manage expense records **Features:**

- Display all user expenses in table
- Search expenses by description
- Edit button for each record
- Delete button with confirmation
- User-specific data filtering
- Total expenses calculation
- Clear search functionality

User Flow:

1. User clicks "Expense Records" in navbar
2. System loads user's expenses
3. User can search by description (optional)
4. Displays filtered results in organized table
5. User can edit or delete records

Income.php

Purpose: View and manage income records **Features:**

- Display all user income in table
- Search income by source name
- Edit and delete functionality
- Date and source information
- Amount display with proper formatting

- Total income calculation
- Clear search functionality

User Flow:

1. User clicks "Income Records" in navbar
2. System loads user's income
3. User can search by source (optional)
4. Displays filtered results in organized table
5. User can edit or delete records

Profile.php

Purpose: User profile management **Features:**

- Display username and email
- Logout functionality with confirmation
- Clean profile information layout
- Secure logout process

User Flow:

1. User clicks "Profile" in navbar
2. Views personal information
3. Can logout with confirmation
4. Redirects to login page

Update Pages

Expenses_update.php

Purpose: Edit existing expense records **Features:**

- Pre-filled form with current data
- Same validation as expense creation
- Update confirmation alerts
- Maintains data integrity

Income_update.php

Purpose: Edit existing income records **Features:**

- Pre-filled form with current data
- Same validation as income creation
- Update confirmation alerts
- Maintains data integrity

8. Validation Rules

Client-Side Validation (HTML5 + JavaScript)

Username Validation

- **Pattern:** ^[a-zA-Z]{3,20}\$
- **Rules:** 3-20 characters, letters and spaces only
- **Purpose:** Ensure clean, readable usernames

Email Validation

- **Pattern:** ^\S{5,50}\$
- **Rules:** 5-50 characters, no spaces
- **Purpose:** Ensure valid email format

Password Validation

- **Pattern:** ^\S{8,20}\$
- **Rules:** 8-20 characters, no spaces
- **Purpose:** Ensure strong, secure passwords

Amount Validation

- **Type:** Number input with step="any"
- **Range:** Min: 1, Max: 999,999,999
- **Purpose:** Prevent negative amounts and unrealistic values

Description Validation (Expenses)

- **Pattern:** ^[A-Za-z0-9]{3,50}\$

- **Rules:** 3-50 characters, alphanumeric and spaces only
- **Purpose:** Ensure clean, readable descriptions

Date Validation

- **Type:** HTML date input
- **Rules:** Valid date format required
- **Purpose:** Ensure proper date entry

9. Server-Side Validation

Duplicate Email Check

```
$sql = "SELECT * FROM users WHERE email='$_email'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    $error_message = "Email already exists!";
```

Empty Field Validation

```
if ($name == "" || $email == "" || $password == "") {
    $error_message = "All fields are required!";
}
```

User Authentication

```
$sql = "SELECT Id FROM users WHERE email='$_email' AND
password='$_password'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // Login successful
} else {
    $error_message = "Invalid email or password!";
```

```
}
```

10. Usage Guide

Users start by registering on the signup page. Once logged in, they can add income or expenses, view and update records, and manage their profile. Navigation is provided via a consistent navbar.