

要件定義書: 掲示板Webアプリケーション

要件定義書: 掲示板Webアプリケーション

1. 概要

このWebアプリケーションは、ユーザーがログインし、掲示板にスレッドを作成し、そのスレッドに対してコメントを投稿することができるものです。ユーザーはアカウントを作成し、ログイン後は自身のプロフィールを管理できます。

2. 機能要件

2.1 ユーザー管理機能

- ・ ユーザーはアカウントを作成し、ログイン・ログアウトができる。
- ・ ユーザーはプロフィール画面でユーザー名を変更できる。

2.2 掲示板機能

- ・ ホームページにはスレッド（投稿）が一覧表示される。
- ・ ユーザーは新しいスレッドを作成できる。
 - スレッドにはタイトルが必要である。
- ・ 各スレッドには、ユーザーが投稿したコメントが表示される。
- ・ コメントはスレッドに対して投稿され、投稿者のユーザー名と投稿日時と共に表示される。
- ・ スレッド一覧でのタブ切り替えにより、スレッドの並び順が変更できる。
- ・ コメントは古い順に上から表示される。

2.3 非同期処理機能

- ・ コメントの編集と削除が非同期で行われる。
 - ユーザーは自分のコメントのみ編集と削除ができる。

2.4 パスワードリセット機能

- ユーザーがパスワードを忘れた場合、ログインページからメール送信確認フォームに遷移できる。
- メール送信確認フォームでは、ユーザーが登録したメールアドレスを入力し、送信ボタンをクリックすることで、そのメールアドレス宛にパスワードリセットの手順を記載したメールが送信される。
- メールに含まれるリセット用のURLにアクセスすると、新しいパスワードを入力し、確認のためにもう一度入力し、更新ボタンをクリックすることでパスワードが更新される。

3. ユーザーインターフェース

- ログインページ: ユーザー名とパスワードの入力フィールドとログインボタンが表示される。
- ホームページ: スレッドが一覧表示される。スレッドを選択すると、そのスレッド内のコメントが表示される。
- ユーザープロフィール画面: ユーザー名の変更フォームが表示される。
- ヘッダー:
 - ログインしていない場合:
 - ログインページへのリンクが表示される。
 - 新規登録ページへのリンクが表示される。
 - ログイン済みの場合:
 - ログアウトボタンが表示される。

4. データ管理

- ユーザー情報: ユーザー名、パスワード、メールアドレス
- スレッド情報: スレッドID、スレッドタイトル、投稿者のユーザー名、投稿日時
- コメント情報: コメントID、スレッドID、コメント本文、投稿者のユーザー名、投稿日時

5. セキュリティ要件

- パスワードはハッシュ化して保存される。
- ユーザーは自分のコメントのみ編集と削除できる。

6. 技術要件

- フロントエンド: HTML, CSS, Bootstrap, JavaScript
- バックエンド: Laravel
- データベース: MySQL

7. テスト要件

- ユーザー登録、ログイン、ログアウトの機能テスト。
- スレッドの表示、およびコメント投稿のテスト。
- コメントの編集と削除の非同期処理テスト。
- パスワードリセット機能のテスト。

8. 可用性要件

- サーバダウン時のエラーハンドリングを実装する。

1. 概要

このWebアプリケーションは、ユーザーがログインし、掲示板を閲覧し、コメントを投稿できるものです。ユーザーはアカウントを作成し、ログイン後は自身のプロフィールを管理できます。

2. 機能要件

2.1 ログイン機能

- ユーザーはアカウントを作成し、ログインできる。
- ログインが成功した場合、ホームページにリダイレクトされる。
- ログインにはユーザー名とパスワードが必要である。

2.2 ログアウト機能

- ログアウトすると、最初のログインページにリダイレクトされる。

2.3 ユーザープロフィール管理機能

- ログイン時に表示されるユーザーアカウント名の箇所をクリックすると、ユーザープロフィール画面に遷移する。
- ユーザープロフィール画面では、ユーザー名の変更が可能である。
- ユーザーはプロフィール画面からログアウトできる。

2.4 掲示板表示機能

- ホームページには複数の掲示板が表示される。
- 各掲示板には投稿されたコメントが表示される。
- コメントは投稿日時と投稿者のユーザー名と共に表示される。

2.5 コメント投稿機能

- ユーザーは各掲示板にコメントを投稿できる。
- コメントは投稿者のユーザー名と投稿日時と共に表示される。

3. ユーザーインターフェース

- ログインページ: ユーザー名とパスワードの入力フィールドとログインボタンが表示される。
- ホームページ: 掲示板が一覧表示され、各掲示板には投稿されたコメントが表示される。
- ユーザープロフィール画面: ユーザー名の変更フォームとログアウトボタンが表示される。

4. データ管理

- ユーザー情報: ユーザー名、パスワード
- コメント情報: 投稿者のユーザー名、掲示板名、コメント内容、投稿日時

5. セキュリティ要件

- パスワードはハッシュ化して保存される。
- ユーザーは自分のコメントのみを削除できる。

6. 技術要件

- フロントエンド: HTML, CSS, JavaScript
- バックエンド: フレームワーク（例: Flask, Django）
- データベース: SQLデータベース（例: SQLite, MySQL）

7. テスト要件

- ユーザー登録、ログイン、ログアウトの機能テスト。
- 掲示板の表示とコメント投稿のテスト。

- ユーザープロフィールの編集とログアウトのテスト。

8. 可用性要件

- サーバーダウン時のエラーハンドリングを実装する。
- バックアップシステムを定期的に行う。