

1 管理者機能

(アフィリエイト管理)

- 管理画面

-アフィリエイト一覧表示

すべてのアフィリエイトを一覧で表示できる機能。

-アフィリエイト詳細表示

各アフィリエイトの詳細情報（プロフィール、登録日、報酬設定、振込先口座）を表示

-アフィリエイトの報酬設定

特定の売上高になったアフィリエイトにはランク付けをする。その基準をこちらで設定可能にする XMTrading 参照

-紹介ユーザー管理

紹介ユーザー管理 複数の管理者がいる場合、それぞれの権限レベル（例：閲覧のみ、編集可能など）を設定できる機能。

-紹介ユーザーの追跡

*各アフィリエイトが紹介したユーザーの情報を表示。

*ユーザーの支払い状況や売上状況をリアルタイムで確認可能。

-管理者設定

複数の管理者がいる場合、それぞれの権限レベル（例：閲覧のみ、編集可能など）を設定できる機能。

-自動通知設定

ユーザーが追加されたらメールで通知する、管理者がログインしたら通知する

-ユーザー通知機能

ユーザーに対してランクが上がったり下がったりしたときに自動でメッセージを送る

-ユーザー一斉通知機能

システムメンテナンスなどが発生したときに通知する用。こちら側から任意でメールを送れる

2 ユーザー機能

1)会員機能

-会員登録

参考 UI：XMTrading 参照

-ログイン

参考 UI : XMTrading 参照

2)マイ ページ

-ログイン

参考 UI : XMTrading 参照

-月別報酬確認

参考 UI : XMTrading 参照

-月別結果の確認

"クリック数、コンバージョン数(会員登録者数、紹介者の課金額の2通り)CVR 参考 UI :

XMTrading 参照

報酬ポイント

→ユーザー登録して入金した時点で〇〇円のアフィリ報酬支払い

→紹介したユーザーがガチャなどを回すのに入金した金額の〇%の支払い"

-振込先登録フォーム

-振込先確認ページ

-アフィリリンク発行ページ

"アフィリエイトが複数のアフィリエイトリンクを作成し、異なる流入経路に対応で

きる機能

各リンクごとのパフォーマンス (クリック数、CVR、売上) を追跡し、比較分析可能"

-出金申請

アフィリエイト報酬を出金できる申請ページ

-出金履歴

出金済みの金額などを確認できるページ

登録、ログイン機能

ダッシュボード

2.2.1アフィリエイトを一覧機能

2.2.2アフィリエイトが紹介したユーザー、紹介ユーザーの支払い状況、売上状況一覧機能

2.2.3アフィリエイト紹介リンクのクリック数確認機能

3.2.1アフィリエイト紹介リンクからのユーザー登録の CVR 確認機能

3.2.2アフィリエイトランクシステム

4.2.1アフィリエイトユーザープロフィールページ

5.2.1 アフィリエイトプロモーション機能

6.2.1 振込先登録フォーム&振込先確認ページ

完成とテスト

オリパ及び ASP の非機能要件です。

この部分も随時追加などあるかもしれませんが、気になる所があればご提案ください。

1. パフォーマンス要件

- ・ レスポンス時間:
 - ・ ユーザーのアクションに対して、主要な操作（例：ページロード、データの保存や取得）は 3 秒以内に完了することを目標とする。
- ・ スケーラビリティ:
 - ・ 同時に 1000 人以上のユーザーがアクセスしてもパフォーマンスが低下しないように設計する。必要に応じて、負荷分散や自動スケーリングを実装。
- ・ データ処理速度:
 - ・ 大量のデータを処理する場合でも、データベースクエリが効率的に実行され、結果が 5 秒以内に返されるようにする。

2. 信頼性・可用性要件

- ・ システム可用性:
 - ・ サービス稼働率は 99.9%以上を目標とし、月間のダウンタイムを 43 分以内に抑える。
- ・ フォールトトレランス:
 - ・ サーバーやネットワーク障害が発生した場合でも、システムが自動的に切り替えられ、ユーザーへの影響を最小限に抑える。
- ・ データバックアップとリカバリ:
 - ・ データは日次で自動バックアップを行い、障害発生時には最長でも 24 時間以内に復旧可能とする。

3. セキュリティ要件

- ・ データ保護:
 - ・ ユーザーの個人情報や取引データを暗号化して保存し、通信時には SSL/TLS を使用してデータの安全性を確保する。
- ・ アクセス制御:

- 管理者やユーザーのアクセス権限を厳格に管理し、重要な操作には二段階認証を導入する。

- 脆弱性対策:

- 定期的にセキュリティスキャンを実施し、SQL インジェクションやクロスサイトスクリプティング (XSS) などの一般的な脆弱性を防ぐ対策を講じる。

- ログ管理:

- すべての重要な操作 (ログイン、データ編集、出金処理など) を記録し、一定期間保存する。

4. メンテナンス性要件

- コードの可読性とドキュメント:

- コードは読みやすく、メンテナンスしやすいように記述し、各機能に関する詳細なドキュメントを整備する。

- バージョン管理:

- ソースコードのバージョン管理を行い、リリース時には明確なバージョン番号を付ける。また、リリースの変更履歴を容易に追跡できるようにする。

5. 使用サーバー

- Xserver などのレンタルサーバーで運営予定