



# AnySpi.net 開発ケーススタディ

信頼と実績を可視化する、Laravelベースの占いマッチングプラットフォーム

個人開発 / フルスタック実装 | 使用技術: LARAVEL 11.31 / MYSQL / BOOTSTRAP 5.3

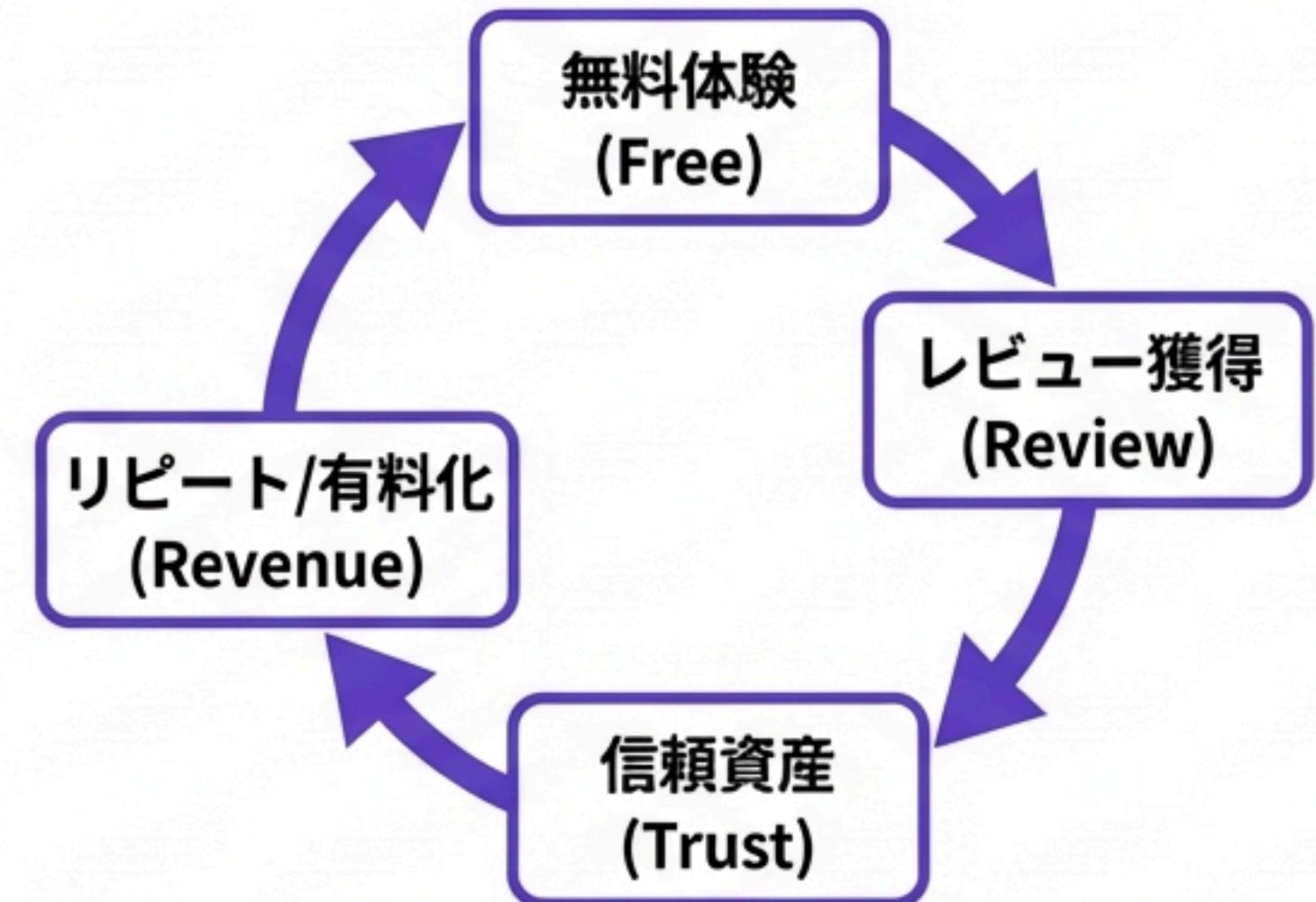
# 市場課題：1兆円市場に潜む「信頼の非対称性」

THE FRICTION



- 心理的ハードル：知らない相手への課金に対する不安。
- コールドスタート問題：新規占い師は実績（レビュー）がないため、集客が困難。

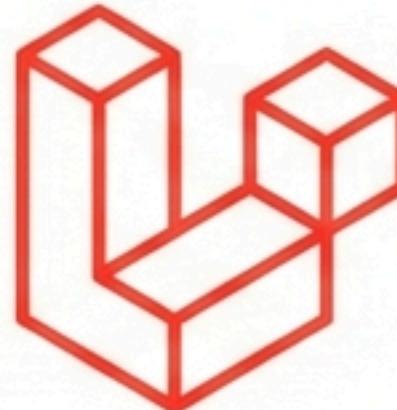
THE SOLUTION



AnySpiの解決策：「無料体験」を起点とした信頼構築モデル。

# 技術スタック：開発スピードと保守性の両立

## Backend



Laravel 11.31  
(Latest Standard)

## Frontend



Bootstrap 5.3 (Admin/User UI)  
+ Tailwind 2.2 (LP)

## Database



Architecture Policy: MVPとしての速さを重視し、Viteを除外したシンプルなLaravel UI構成を採用。

複雑性を排除し、ビジネスロジックの実装に集中できる環境を構築。

# データベース設計：拡張性を考慮したスキーマ

1

## Users Table:

管理者(Admin)・占い師(Reader)・ユーザー(User)を単一テーブルで管理し、JOINコストを削減。

users			
bigint	id	PK	
string	name		
string	email	UK	
boolean	is_reader		占い師フラグ
boolean	is_admin		管理者フラグ
integer	points		現在ポイント
integer	login_streak		連続ログイン数
string	invite	UK	招待コード

2

## Performance:

Eager Loading実装。N+1問題を回避。

partners		
bigint	id	PK
foreign_id	user_id	FK
string	name	
date	birth_date	
text	relationship	

login_benus_logs			
bigint	id	PK	
foreign_id	user_id	FK	
date	login_date	UK	同一日の重複付与防止

inviters			
bigint	id	PK	
foreign_id	inviter_id	PK	紹介した人
foreign_id	invitee_id	FK	紹介された人
integer	points_awarded		

reviews			
bigint	id	PK	
foreign_id	user_id	FK	相談者
foreign_id	reader_id	FK	占い師(Usersテーブル)
tinyinteger	rating		
text	comment		
boolean	is_public		

review_reports			
bigint	id	PK	
foreign_id	review_id	FK	
foreign_id	reporter_id	FK	送信者
string	status		pending/approved/rejected

3

Safety: `user\_id` と `login\_date` の複合ユニーク制約で重複付与を防止。

# MVP戦略：アジャイルな意思決定とトレードオフ

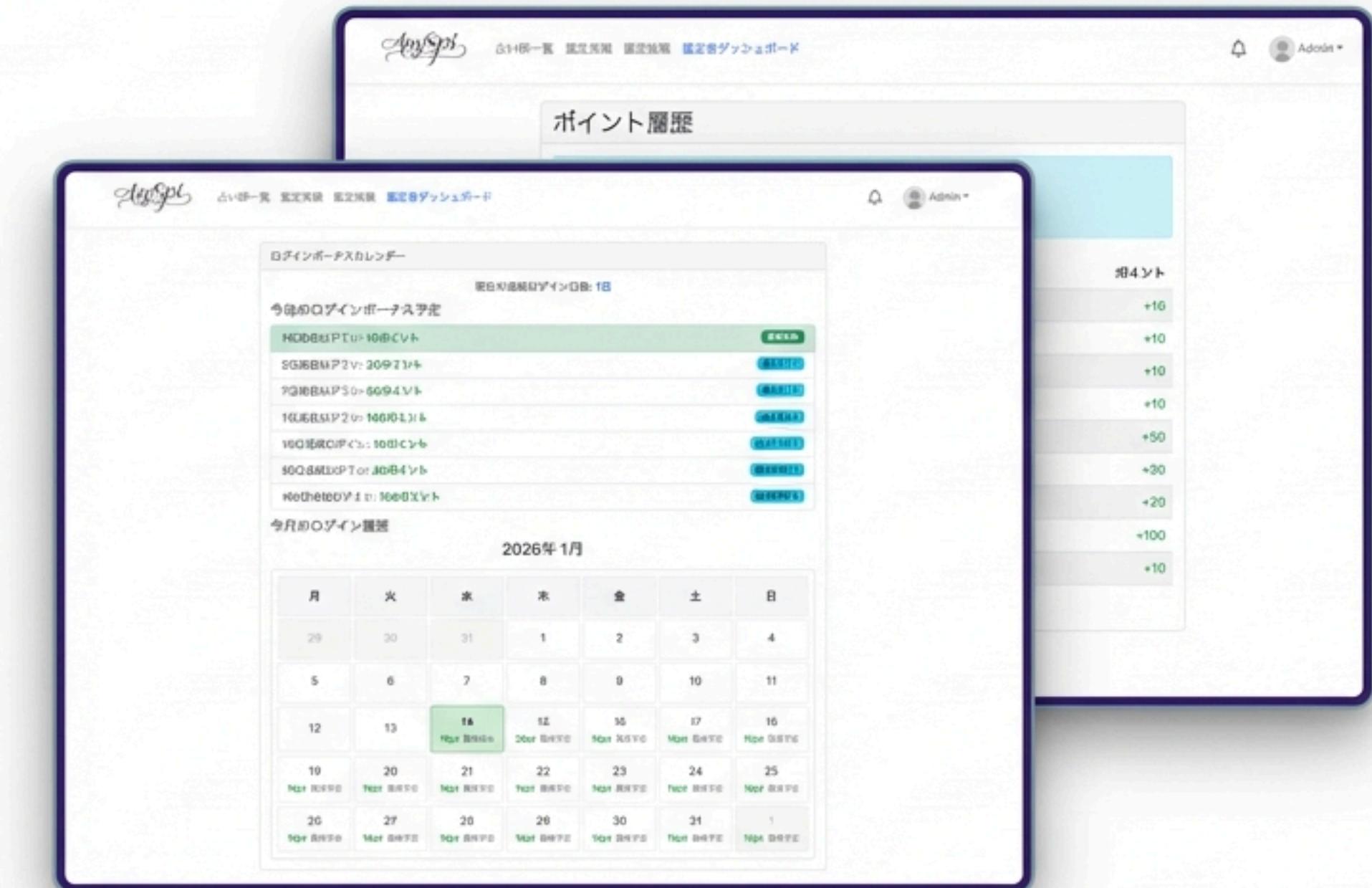
課題	3種類のユーザー（一般/占い師/管理者）の権限管理。
選択	`roles` テーブルによるRBACではなく、`is_admin` / `is_reader` フラグを採用。
理由	スクラム開発の考えに基づき、初期実装コストを最小化。要件が複雑化した時点でRole管理へ移行可能なモジュラーな設計。

```
$table->boolean('is_admin')->default(false);  
$table->boolean('is_reader')->default(false);
```

# ゲーミフィケーション：継続利用を促す仕組み

## Engagement Loop

- **ログインストリーク:** 連続ログイン日数に応じたインセンティブ設計。
- **透明性:** カレンダービューで将来の獲得ポイントを可視化し、翌日のログインモチベーションを向上。
- **実装:** `login_streak` カラムと `last_login_date` による効率的な判定ロジック。



# 信頼エンジン：レビューとポイントの循環システム



ユーザーが鑑定を依頼  
サービス提供  
レビュー投稿  
双方にポイント付与

The screenshot shows a user profile page for 'Sana'. The profile includes a circular logo with a tree, the name 'Sana', and the title 'タロット占い師'. Below the name is a green 'active' status indicator. A 5-star rating is shown with '0.0 (0件の評価)'. There are sections for 'RC紹介' (Introduction), '得意な2周' (Specialties), and '特徴を記入する場合は複数行へ6行まで' (Notes). A large blue button at the bottom says 'レビューを登録しました!' (Review registered!).

## Data Integrity

- `reviews` テーブルは `users` (reader) と `users` (client) を紐付け。
- `review\_reports` テーブルによる通報機能の実装。

# バイラル施策：招待システムとUXへの配慮



- X連携:** シェアボタンから自動生成されたテキストで即時投稿可能。
- Cookieトラッキング:** `invite=` パラメータをCookieに保存し、URLをクリーンに保ちつつリダイレクト(`InviteController` の実装)。
- Code Focus:** `InviteController` 内の二重付与防止ロジックとセッション管理。

```
public function handleInviteRequest(Request $request, $stag)
{
    // UniqueTagをもつ招待ユーザーを取得
    $inviter = User::where('invite', $stag)->first();

    if (!$invited) {
        // 紹介者の登録がまだない場合はトップページへリダイレクト
        return redirect('/');
    }

    // セッションに紹介者IDを保存
    session(['inviter_id' => $inviter->id]);

    // ログイン済みであれば、獲得ポイントをロック解除
    if (Auth::check()) {
        $invited = Auth::user();
        // 自身の登録を複数することはできない
        if ($inviter->id == $invited->id) {
            session()->forget('inviter_id');
            return redirect('/');
        }
    }

    // 図に紹介済みかチェック
    $alreadyInvited = Inviter::where('inviter_id', $inviter->id)
        ->where('invitee_id', $invited->id)
        ->exists();

    if ($alreadyInvited) {
        session()->forget('inviter_id');
        return redirect('/');
    }

    // ログインしていない場合は登録/ログインページへリダイレクト
    return redirect()->route('register');
}
```

# 鑑定師ダッシュボード：供給側の運用ツール

# 管理画面：プラットフォームの健全性を可視化

The screenshot shows the AnySpi platform's management dashboard. At the top left is the logo "AnySpi". To its right are links for "占い師一覧", "鑑定実績", "鑑定結果", and "管理者ダッシュボード". On the far right are icons for notifications and user status, with "Admin" displayed.

The main area is titled "管理者ダッシュボード" and contains four colored boxes with KPIs:

- Blue box: 13 総ユーザー数
- Green box: 3 鑑定者数
- Cyan box: 10 信頼者数
- Yellow box: 940 総ポイント数

Below these boxes is a sidebar titled "管理メニュー" containing the following items:

- POINT付与設定
- ユーザー一覧
- POINT履歴
- レビュー遮蔽管理
- お知らせメール送信
- メール設定

- **KPIモニタリング**: 総ユーザー数、アクティブ鑑定者数、発行済み総ポイント数をリアルタイム表示。
- **UI設計**: Bootstrap 5.3 ベースの直感的な管理インターフェース。

# ユーザーサポート：監査ログを伴うポイント管理

The screenshot shows a user management interface with a modal window for point editing.

**User List:** Shows a list of users with columns: ID, Name, Email Address, Type, Points, Registered Date, and Actions. A user with ID 13 is selected.

**Point Management Modal:** The modal title is "■■■■■ のポイント管理". It has two main sections: "ユーザー情報" (User Information) and "ポイント編集" (Point Edit).

**ユーザー情報 (User Information):**

ID	13
名前	■■■■■
メールアドレス	■■■■■@■■■■■.■■■
種別	招待者
現在のポイント	110pt

**ポイント編集 (Point Edit):**

Current points: 0  
正(+): 増加, 負(-): 減算  
Change Reason: (Example: 不正利用による減算、イベント特典による加算)  
Update Points

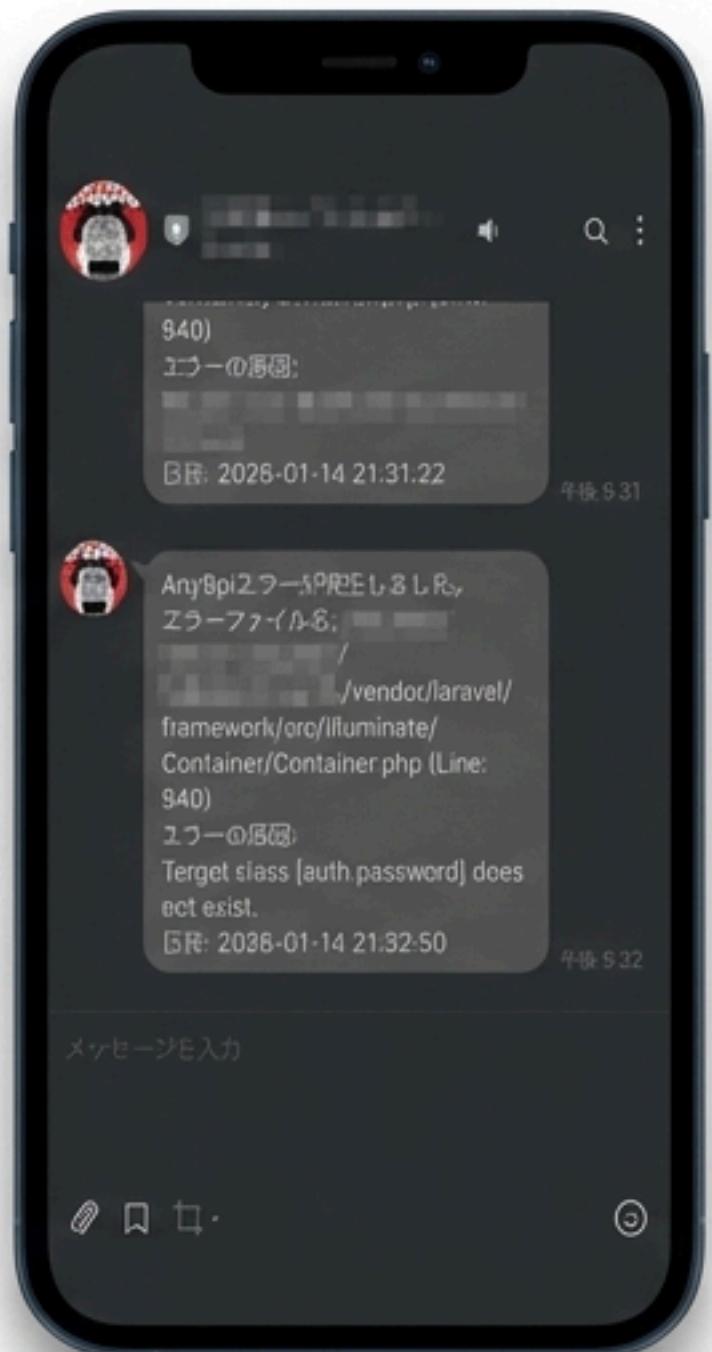
- 手動操作：**トラブルや返金対応時に管理者がポイントを増減可能。
- 監査証跡：**変更時には「変更理由」の入力を必須化し、不正やミスを防止。履歴はユーザー側のログにも反映。
- 検索機能：**IDやメールアドレスによる高速なユーザー特定。

# マーケティング：セグメント配信と変数埋め込み



- ・**セグメント送信**：「一般ユーザー」「鑑定者」「全員」から対象を選択可能。
- ・**変数差し込み**：{name} や {points} などのタグ機能により、パーソナライズされたメールを送信。
- ・**共通設定**：管理画面からメールフッターを一元管理し、ブランディングを統一。

# 障害対応：LINE通知による即時検知システム



## Error Handling Flow

- Exception Handler:** Laravelの例外処理を拡張し、500エラーをフック。
- LINE Messaging API:** エラー詳細（ファイル名、行数、内容）を開発者のLINEへプッシュ通知。
- 効果:** ユーザーからの問い合わせ前に障害を検知し、修正に着手する「プロアクティブ」な運用体制。

# 今後のロードマップと技術的展望



## Refinement

JetBrains Mono

整合性強化: 招待ロジックにおけるユニーク制約の厳格化（DBレベル）。



## Security

JetBrains Mono

セキュリティ: メール認証完了後のイベント連動によるポイント付与フローの確立。



## Scalability

JetBrains Mono

スケーラビリティ: ユーザー種別の増加に伴う、Flag管理からRole/Permission管理への移行。

“

AnySpiは、MVPとしての「スピード」と、将来の拡張に耐えうる「堅牢性」のバランスを重視したプロダクトエンジニアリングの実践です。

# Thank You

ご覧いただきありがとうございます。



A

杉浦 栄治