ʿAwn
Stage 1 & 2 & 3 Report

عُون

Table of Contents

Table of Figures

1. **Team Formation Overview**

1.1. Team Members:

Name	Primary Roles	Secondary Roles	Strengths
Rawan Albaraiki	UX/UI Lead	Developer	Multidisciplinary Skillset, Analytical Thinking
Donna Almadani	Project Manager	Developer, Product management	Problem-Solving Mindset, Commitment & Reliability
Shahad Aljahdali	Documentation Lead	Developer	Problem-solving ability, excellent communication, experience with database management.
Munirh Alsubaie	Technical Lead	Developer	Technical Proficiency, Adaptability

1.2. Collaboration Strategy:

- Communication through WhatsApp.
- Weekly check-ins to align progress.
- Shared documentation in Google Docs.

2. **Ideas Explored**

Idea Name	Description	WeaknessStrengths	Weakness	Decision
Desert Whispers	A website that collects the natural sounds of the Saudi desert (night calm, wind, birds).	Unique and simple to implement; highlights Saudi identity.	Very niche audience; limited long-term sustainability as a product	Rejected
TimeSwap	A platform where people exchange services using time (e.g., 1 hour Photoshop lesson = 1 hour language tutoring).	Innovative, promotes community collaboration.	Technically complex (trust system, time banking); hard to launch as MVP	Rejected
Emotion Garden	Users write strange dreams → app generates illustrations or short animations.	Fun, viral potential, highly innovative.	Requires advanced AI + animation resources; high technical barrier	Rejected
ʿAwn	Web app connecting patients with physical therapists + awareness resources.	Solves unemployment, improves access, raises awareness.	Requires careful scope definition	Selected
Dream Journal AR	Users write strange dreams → app generates illustrations or short animations.	Fun, viral potential, highly innovative.	Requires advanced AI + animation resources; high technical barrier.	Rejected
Star Gazing	Website that highlights top locations in Saudi Arabia for star gazing, with maps and photos.	Attractive, localized, clear target audience (travelers, astronomy fans); feasible with existing mapping tools.	Requires accurate data collection for spots, time consuming.	Rejected

3. Selected MVP Concept – ʿAwn

3.1. Introduction:

ʿAwn is a web-based platform designed to make physical therapy services more accessible while supporting therapists in finding employment opportunities. Inspired by existing healthcare apps like *Labayh*, *ʿAwn* focuses exclusively on physical therapy services, filling a gap in both awareness and availability.

3.2. Pain Points Addressed:

Physical therapy in Saudi Arabia is often exclusive to hospitals and medical centers, leaving many patients unaware of its importance or unable to access services. At the same time, certified therapists face unemployment challenges due to limited opportunities. *ʿAwn* bridges this gap by providing a platform that connects therapists and patients while also offering educational resources.

3.3. Target Audience:

- Patients in need of accessible physical therapy services.
- Certified physical therapists seeking employment opportunities.
- Educators and health organizations raising awareness of physical therapy.
- General public with limited knowledge of physical therapy benefits.

3.4. MVP Features:

Patient Side:

1. Dual Registration System – On the homepage, users can register either as a Specialist (Therapist) or as a Patient, with sign-in linked to their National ID and verified through Absher for authenticity.
2. Appointment Calendar – A visual calendar showing weekly or monthly appointments, making it easy for patients to track upcoming sessions.
3. Therapist Directory – A searchable and filterable list of specialists, with filters for location proximity and gender preference (male or female therapist).
4. Therapist Profile Page – Displays:
    - Short bio and professional summary.
    - Certification(s) and qualifications.
    - Years of experience.
    - Patient ratings and reviews.
5. Booking System – Patients can select available slots directly from a therapist's calendar.

Therapist Side:

1. Therapist Dashboard – Different view from patients, tailored to specialists.
2. Medical Records Access – Therapists can view relevant patient records and health history to prepare for sessions.
3. Appointment Management – Specialists can manage their schedule, view upcoming bookings, and mark availability.
4. Patient Notes – Option to log session notes, track patient progress, and update case information.

Shared Features (Both Roles):

- Authentication via Absher – Ensures trusted identity verification for both patients and specialists.
- Ratings & Reviews System – Patients can leave feedback after consultations, building trust and accountability.
- Awareness/Education Section – Centralized space with articles, FAQs, and media about physical therapy.

4. Documentation of the Process (SCAMPER Framework)

Substitute**:** Instead of general healthcare apps, focus only on physical therapy to fill a market gap.
Combine**:** Merge job opportunities for therapists with patient access and cultural awareness.
Adapt**:** Adapt successful healthcare models like *Labayh* but narrow the scope to physical therapy.
Modify**:** Replace complex features like user accounts with simpler engagement (emoji reactions).
Put to Another Use: Repurpose physical therapy expertise into educational digital content.
Eliminate**:** Exclude advanced features (payments, insurance integration) to keep MVP feasible.
Reverse**:** Instead of patients searching only in hospitals, flip access by making therapists available digitally.

Outcome**:** After exploring multiple ideas, *ʿAwn* was chosen as the MVP because it is feasible, socially impactful, and aligned with the team's skills and goals.

5. Project Charter

5.1. Project Objectives

Purpose:
The purpose of *ʿAwn* is to improve access to physical therapy services while addressing therapist unemployment and increasing public awareness of physical therapy's importance.

SMART Objectives:

1. Develop a functional web MVP by 9 November with therapist profiles, booking, and awareness features.
2. Onboard at least 10 certified therapists and publish 10 awareness resources by the MVP launch.
3. Reach at least 20 patients during pilot testing, with 5 completed consultations.

5.2. Stakeholders and Team Roles

Internal Stakeholders

- Team Members – developers, designers, documentation lead, project manager.

Role	Member	Responsibilities
Project Manager	Donna Almadani	Oversees planning, tracks progress.
Technical Lead	Munirh Alsubaie	Guides tech stack decisions.
Developer(s)	All	Build booking, profiles, awareness section.
Designer/UX Lead	Rawan Albaraiki	UI/UX design for patients and therapists.
Documentation Lead	Shahad Aljahdali	Maintains project documentation.

External Stakeholders

- Mentors – guidance and technical/strategic support.
- Therapists – professionals offering services via the platform.
- Patients – end-users in need of accessible physical therapy.

6. Scope

6.1 In-Scope

- Therapist profiles and availability.
- Basic patient booking system (no payments).
- Awareness/education content (articles, FAQs, videos).
- Suggestion box, emoji reactions, and share button.

6.2 Out-of-Scope

- Advanced payment systems.
- Telemedicine (video calls).
- Mobile app version (web MVP only).
- Insurance or hospital system integrations.

7. Risks and Mitigation

Risk	Potential Impact	Mitigation
Few therapists join the platform	Limited services	Start with small pilot group, partner with mentors/clinics.
Lack of awareness	Low user adoption	Awareness campaign via social media and community groups.
Technical challenges with scheduling	Booking feature fails	Start with simple time-slot booking.
Team inexperience in healthcare compliance	Data/security risks	Avoid storing sensitive data in MVP.

8. High-Level Project Plan

Timeline (18 August – 9 November)

- **Stage 1 (18 Aug – 31 Aug):** Team Formation & Idea Development Completed
- **Stage 2 (1 Sept – 14 Sept):** Project Charter Development *(current stage)*
- **Stage 3 (15 Sept – 28 Sept):** Technical Documentation – define architecture, datasets, tools.
- **Stage 4 (29 Sept – 12 Oct):** MVP Development – implement therapist profiles, booking, awareness.
- **Stage 5 (13 Oct – 26 Oct):** MVP Refinement – bug fixes, improvements, pilot testing.
- **Stage 6 (27 Oct – 9 Nov):** Project Closure – final testing, presentation, and submission.

9. User Stories and Mockups

9.1 MoSCoW User Stories

Must Have

- As a patient, I want to sign up and log in so that I can book a session.
- As a therapist, I want to sign up and log in so that I can publish my profile and availability.
- As a patient, I want to browse a directory of therapists with filters (city, gender, specialty) so that I can find a suitable therapist.
- As a patient, I want to book an available time slot so that I can schedule a therapy consultation.
- As a therapist, I want to see my upcoming appointments in a calendar so that I can manage my schedule.
- As an admin, I want to verify therapist profiles so that only legitimate professionals are listed.

Should Have

- As a patient, I want to receive email reminders before my appointment so that I don't miss it.
- As a patient, I want to see ratings/reviews on therapist profiles so that I can choose confidently.
- As a user, I want to switch language (AR/EN) so that I can use the site comfortably.

Could Have

- As a patient, I want to save favorite doctors, so that I can book with them easily later.
- As a therapist, I want to leave short session notes (non-sensitive) so that I can track follow-ups.
- As a patient, I want to get SMS reminders so that I'm notified on my phone.
- As a therapist, I want to see analytics of my consultations, so that I can improve my services.

Won't Have

- Insurance integrations, online payments, video telemedicine, storing detailed medical records, Absher/Nafath identity integration.

9.2 Mockups

Figure 1 - Mockups

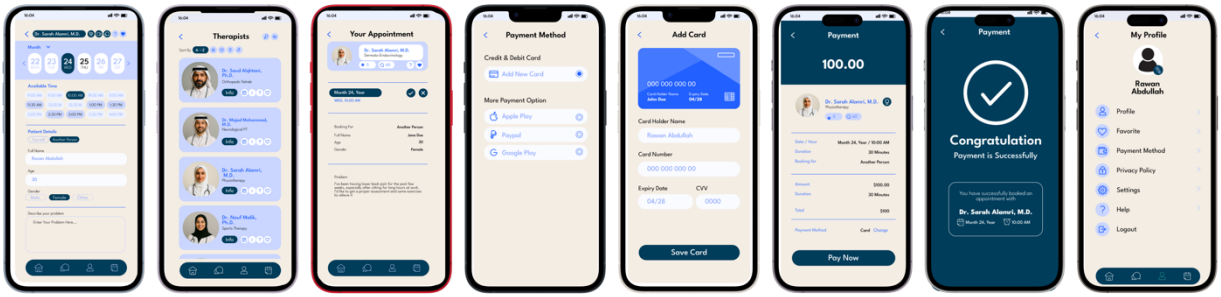

Figure 2 - Mockups

10. System Architecture

Tech choice: Laravel 11 (PHP 8.3), Blade + Livewire, MySQL/Postgres, Filament Admin, FullCalendar (JS), Scout (+ Meilisearch if time), Mail (email reminders).

Hosting: Laravel Forge or simple VPS; staging + production.

10.1 High-level flow

- Frontend (Blade/Livewire) → Laravel Controllers → Services/Policies → Eloquent (DB)
- Notifications (Mail; later SMS via Unifonic) via Laravel Notifications/Queues
- Search via DB queries; optional Scout/Meilisearch if time permits

10.2 Data flow examples

- Patient opens “Therapist Directory” → API/Controller fetches profiles (with filters) → returns list → click profile → fetch availability → book → creates booking with transaction guard → sends email notifications.
- Therapist edits availability (Livewire component) → creates availability_slots → shows in calendar to patients.

11. Components, Classes, and Database Design

11.1 Key Backend Components

- Auth & Roles: Breeze + spatie/permission (patient, therapist, admin)
- Modules:
  - Therapists (profile, verification)
  - Directory (search/filter)
  - Availability (create/manage slots)
  - Bookings (create/cancel/confirm)
  - Reviews (post-booking)
  - Content (awareness articles)
  - Notifications (email reminders)
  - Admin (Filament resources)

11.2 ER Diagram

- users (id, name, email, phone, password_hash, role, gender, locale, created_at)
- therapist_profiles (id, user_id FK, bio, specialties (JSON/array), years_experience, city, lat, lng, certifications_url, verified boolean, created_at)
- availability_slots (id, therapist_id FK, starts_at, ends_at, status: open|blocked, created_at)
- bookings (id, patient_id FK (users), therapist_id FK (users), slot_id FK, status: pending|confirmed|completed|cancelled, notes_short, created_at)
- reviews (id, booking_id FK, rating int (1–5), comment, created_at)
- articles (id, title, slug, body_md, tags (JSON), published_at)

Rules

- A therapist_profile belongs to one user with role=therapist.
- A booking must reference an existing availability_slot; enforce no double-book with unique constraint on slot_id where status active.
- A review can be created only if booking.status = completed and by that booking's patient.

12. High-Level Sequence Diagrams

A) Patient books an appointment

1. Patient → UI: open therapist profile
2. UI → API: GET /therapists/{id}/availability
3. Patient → UI: pick slot → POST /bookings { therapist_id, slot_id }
4. API: validates role=patient, slot open, creates booking (transaction), marks slot reserved
5. API → Mailer: send emails to patient & therapist
6. API → UI: return booking confirmation

B) Therapist publishes availability

1. Therapist → UI: open “My Availability”
2. UI → API: POST /availability { slots[] }
3. API: validates role=therapist, inserts slots
4. API → UI: returns updated calendar

C) Patient posts a review

1. Patient → UI: open past appointment
2. UI → API: POST /reviews { booking_id, rating, comment }
3. API: checks booking belongs to patient & status=completed; saves review
4. API → UI: success; therapist profile aggregates rating

13. API Specifications

All responses are JSON; all endpoints require auth unless stated. Headers: Authorization: Bearer <token> (Breeze session or Sanctum token, whichever you choose).

Public / Auth

- POST /auth/register — body: {name, email, password, role} → 201 {user, token}
- POST /auth/login — body: {email, password} → 200 {user, token}
- POST /auth/logout — 204

Directory & Profiles

- GET /therapists?city=&gender=&specialty= → 200 [{id, name, city, gender, rating_avg, specialties}]
- GET /therapists/{id} → 200 {profile, reviews_summary}
- GET /therapists/{id}/availability?from=&to= → 200 [{slot_id, starts_at, ends_at, status}]

Availability (therapist role)

- POST /availability — body: [{starts_at, ends_at}, …] → 201 [{slot_id,…}]
- DELETE /availability/{slot_id} → 204

Bookings

- POST /bookings — body: {therapist_id, slot_id} → 201 {booking_id, status}
- GET /bookings/mine (patient) → 200 […]
- GET /bookings/assigned (therapist) → 200 […]
- PATCH /bookings/{id} — body: {status} → 200 (therapist can confirm/cancel; patient can cancel)

Reviews (post-booking)

- POST /reviews — body: {booking_id, rating (1–5), comment} → 201
- GET /therapists/{id}/reviews → 200 […]

Content (awareness)

- GET /articles?tag= → 200 [{id, title, slug, excerpt}]
- GET /articles/{slug} → 200 {title, body_md, tags}

14. SCM and QA Strategies

SCM

- Repo: GitHub
- Branches: main (stable), develop (integration), feature/*
- PR rule: at least 1 review; no direct pushes to main
- Conventional commits: feat:, fix:, chore:, docs:, test:

CI/CD

- GitHub Actions: run tests + lint on PR
- Deploy develop → staging (Forge) after tests pass
- Manual approval to deploy main → production

QA

- Unit/Feature tests: PHPUnit (models, policies, booking rules, availability)
- HTTP tests: Laravel HTTP tests ($this->post('/bookings'…))
- Postman/Newman collections**:** smoke test endpoints
- Manual E2E (staging): key user flows (register, list therapists, book, cancel, review)
- Non-functional checks: Arabic/RTL layout, accessibility basics, performance (N+1 queries)

Test priorities

- Prevent double-booking
- Role policies (patient vs therapist vs admin)
- Reviews only after completed bookings

15. Technical Justifications

- Laravel: fastest full-stack path (auth, RBAC, notifications, localization) for a two-month MVP; huge local market relevance.
- Blade + Livewire: avoids SPA overhead; lets junior/mid devs ship interactive UI quickly.
- Relational DB (Postgres/MySQL)**:** strong integrity for bookings/availability; easy joins for directory filters.
- Filament Admin**:** admin CRUD and therapist verification without writing a whole admin app.
- Email first, SMS later: cut risk; plug Unifonic later via Notifications channel.
- Privacy**:** no sensitive medical records in MVP; only short session notes; enforce RBAC with policies; HTTPS only.