



b7a55823-d47e-4e58-8a...

File

اختر الخيار الثاني:

Must be a non-null

`CalendarSchemes.AnchorStarId` **column**

وأرسل له هذا التوضيح إن أمكن:

`CalendarScheme.AnchorStarId` is a required, non-null foreign key to `Stars.Id`. The anchor star must be persisted and database-enforced, not domain-only. If the required one-to-many relationship creates circular migration/seed ordering, handle insertion in two steps or use a link row internally, but retain the non-null `AnchorStarId` column as the authoritative .anchor reference

هذا يتوافق مع مواصفات المشروع التي عرّفت `AnchorStarId` صراحةً.

احتثرت الخيار الثاني ولم استطع ان ارسل
التوضيح لانه شغال هو

Worked for 29s >

لا تقلق، اختيارك الثاني صحيح ومتوافق مع المواصفات:

يجب أن يكون عمودًا `CalendarSchemes.AnchorStarId`
إلزاميًا وغير قابل لـ `NULL`، ومرتبًا بـ `Stars.Id`.

اترك الـ `sub-agent` يكمل ولا توقفه. بعد انتهائه راجع فقط أن:

- `AnchorStarId` غير قابل لـ `NULL`.
- يوجد FK من `CalendarSchemes.AnchorStarId` إلى `Stars.Id`.
- لا توجد علاقة Domain-only فقط.

- تهيئة البيانات لا تنتج نجومًا مكررة.
- إذا ظهرت مشكلة ترتيب الإدخال الدائري، يعالجها بخطوتين أثناء الـ seed دون جعل nullable AnchorStarId.

إذا عاد وسأل عن هذه النقطة مرة أخرى، اختر **Other / custom** واكتب:

Keep `CalendarSchemes.AnchorStarId` as a required non-null foreign key. Resolve any circular seed ordering with a two-step initialization; do not make the anchor nullable .or domain-only

Today 1:23 PM

Implemented Phase 2A —
Persisted Calendar Vertical
Slice.

Delivered

- PostgreSQL persistence for CalendarScheme, Star, and Region.
- EF Core mappings, audit/concurrency fields,

Show more ▾

Worked for 22s >

رائع. Phase 2A مكتملة بشكل صحيح، ولا تحتاج إلى تعديل علاقة AnchorStarId طالما أن الاختبارات والـ migration نجحت.

قبل الانتقال، احفظ نسخة أو أنشئ Git commit باسم مثل:



feat: complete phase 2a persisted calendar

ثم أرسل إلى Codex هذا البرومبت:



Edit

Continue from the current StarCalendar repository. Phase 2A is complete and verified. Do not rewrite or recreate the existing implementation.

Read `AGENTS.md` and the project documents before making changes. Preserve the existing PostgreSQL persistence model, including the required non-null `CalendarSchemes.AnchorStarId` and the existing deferred seed strategy.

Task: Implement Phase 2B — Public Stars, Content, Crops, and Region Read APIs

Implement only Phase 2B. Do not begin Phase 3 administration authentication or the admin CRUD interface yet.

1. Domain and persistence

Add the project-specific domain and persistence models required for public cultural/agricultural content:

- `ContentItem`
- `Crop`
- `StarCropRecommendation`

Use the existing `Star` and `Region` relationships.

Content types must support at least:

- Overview
- Proverb
- ProverbExplanation
- TraditionalWeather
- Planting
- Harvest
- Irrigation
- GeneralAgriculture

- Warning

Content items must include:

- Star relationship.
- Optional Region relationship.
- Arabic title and body.
- Content type.
- Publication status: Draft, Published, Archived.
- `IsSample` flag.
- Optional source fields.
- Sort order.
- Audit fields.

Crop recommendations must include:

- Star.
- Crop.
- Optional Region.
- Activity type.
- Arabic notes.
- Publication status.
- `IsSample` flag.

Add EF Core configurations, constraints, migration, and idempotent seed behavior.

Use a migration name such as:

```
AddStarContentAndCropData
```

Do not modify the already verified calendar calculation or persisted scheme behavior unless a failing test proves it is necessary.

2. Sample data

Seed clearly labeled placeholder data for all 28 stars.

For every star, seed at least:

- One sample overview.

- One sample proverb.
- One sample proverb explanation.
- One sample traditional-weather entry.
- One sample planting entry.
- One sample harvest entry.
- One sample general farming entry.

Every invented item must:

- Set `IsSample = true`.
- Begin with `[بيانات تجريبية]` or display an equivalent sample marker.
- Avoid claiming to be an authentic local proverb or verified agricultural fact.
- Avoid invented source names or citations.
- Remain editable and removable later through the admin panel.

Seed a small set of clearly labeled sample crops and star-crop recommendations.

The public API may return sample content, but the response must expose the `isSample` value so the clients can display a visible notice.

3. Public API endpoints

Implement versioned public read endpoints:

```
GET /api/v1/stars
GET /api/v1/stars/{slug}
GET /api/v1/stars/{slug}/content?region
GET /api/v1/regions
GET /api/v1/crops
GET /api/v1/stars/{slug}/recommendation.
```

Requirements:

- Public reads do not require authentication.
- Return only `Published` content.

- Keep Draft and Archived content private.
- Preserve the confirmed star order.
- Support optional region filtering.
- Use explicit response contracts/DTOs.
- Do not expose EF Core entities.
- Use MediatR queries, validators where needed, Result Pattern, and existing Problem Details mapping.
- Use the existing caching pipeline for stable public lists where appropriate.
- Return consistent Arabic fields and stable technical slugs.

4. API response expectations

`GET /api/v1/stars` must return all 28 stars in their confirmed order.

`GET /api/v1/stars/suhail` must include:

- Arabic name.
- Slug.
- Order.
- Aliases if available.
- Published overview.
- Published content grouped by type.
- Sample markers.
- Related published crop recommendations.

Do not calculate Gregorian date ranges by storing annual tables. The existing calendar endpoint remains the only source for date calculations.

5. Automated tests

Add PostgreSQL-backed integration tests and application tests for:

- Migration applies to an empty database.
- Seed is idempotent.

- All 28 stars are returned in the correct order.
- Star lookup by slug works.
- Unknown slug returns 404 Problem Details.
- Published content is returned.
- Draft and Archived content are not returned publicly.
- Region filtering works.
- Sample markers are preserved.
- Crop recommendations are returned for the correct star.
- Empty content produces a valid empty collection rather than an exception.
- Public endpoints do not require authentication.
- Existing Phase 2A tests continue to pass unchanged.

6. Documentation and verification

Update:

- `README.md`
- API documentation/OpenAPI examples.
- Database and seed-data documentation.
- A short ADR if a new content/publication decision is introduced.
- The Phase 2 checklist.

Run:

- .NET formatting.
- .NET Release build.
- All .NET tests.
- PostgreSQL migration and integration tests.
- React formatting, lint, tests, and production build.

- Flutter formatting, analysis, tests, and existing APK build.

Do not implement:

- Admin login.
- Admin CRUD.
- User accounts.
- Weather provider integration.
- Push notifications.
- Media uploads.
- Blazor.
- SQL Server.
- MechanicShop domain artifacts.
- Unlabeled invented cultural or agricultural claims.

At the end, report the entities, migration, endpoints, seed behavior, tests, and verification results. Stop after Phase 2B and wait before starting Phase 3.