

# Make My Marriage

API Design Document — Version 1.0

Backend: FastAPI + Python Database: MongoDB Atlas Frontend: Next.js

FastAPI is the main business-logic, validation and authorization layer. Next.js calls FastAPI; it never connects directly to MongoDB.

## 1. API Principles

- REST + JSON, base path /api/v1.
- Private APIs require login and wedding membership/role checks.
- Public APIs use wedding slugs or secure public tokens.
- Pydantic models validate request/response data.
- Consistent errors, pagination and filtering are used across modules.

## 2. API Modules

/auth • /weddings • /events • /guests • /tasks • /shopping • /expenses • /vendors • /invitations • /photos • /documents • /public

## 3. Authentication

| Method | Endpoint              | Purpose                      |
|--------|-----------------------|------------------------------|
| GET    | /auth/google          | Start Google login           |
| GET    | /auth/google/callback | Handle Google OAuth callback |
| POST   | /auth/logout          | Logout                       |
| GET    | /auth/me              | Current user                 |

## 4. Weddings & Members

| Method | Endpoint                                   | Purpose                       |
|--------|--------------------------------------------|-------------------------------|
| POST   | /weddings                                  | Create wedding                |
| GET    | /weddings/{wedding_id}                     | Get wedding                   |
| PATCH  | /weddings/{wedding_id}                     | Update wedding                |
| GET    | /weddings/{wedding_id}/members             | List members                  |
| POST   | /weddings/{wedding_id}/members/invite      | Invite member by email + role |
| PATCH  | /weddings/{wedding_id}/members/{member_id} | Change role/status            |
| DELETE | /weddings/{wedding_id}/members/{member_id} | Remove member                 |

## 5. Events

| Method | Endpoint                      | Purpose             |
|--------|-------------------------------|---------------------|
| GET    | /weddings/{wedding_id}/events | List events         |
| POST   | /weddings/{wedding_id}/events | Create event        |
| GET    | /events/{event_id}            | Get event           |
| PATCH  | /events/{event_id}            | Update event        |
| DELETE | /events/{event_id}            | Delete/cancel event |
| GET    | /events/{event_id}/guests     | Event guests        |
| GET    | /events/{event_id}/photos     | Event photos        |

## 6. Guests & RSVP

| Method | Endpoint                             | Purpose                |
|--------|--------------------------------------|------------------------|
| GET    | /weddings/{wedding_id}/guests        | List/search guests     |
| POST   | /weddings/{wedding_id}/guests        | Add guest              |
| PATCH  | /guests/{guest_id}                   | Update guest           |
| DELETE | /guests/{guest_id}                   | Remove guest           |
| POST   | /events/{event_id}/guests            | Assign guest to event  |
| PATCH  | /events/{event_id}/guests/{guest_id} | Update RSVP/attendance |
| GET    | /weddings/{wedding_id}/rsvp/summary  | RSVP summary           |

## 7. Tasks & Shopping

| Method | Endpoint                        | Purpose                |
|--------|---------------------------------|------------------------|
| GET    | /weddings/{wedding_id}/tasks    | List tasks             |
| POST   | /weddings/{wedding_id}/tasks    | Create task            |
| PATCH  | /tasks/{task_id}                | Update task/status     |
| DELETE | /tasks/{task_id}                | Delete task            |
| GET    | /weddings/{wedding_id}/shopping | List shopping items    |
| POST   | /weddings/{wedding_id}/shopping | Add item               |
| PATCH  | /shopping/{item_id}             | Update purchase status |
| DELETE | /shopping/{item_id}             | Delete item            |

## 8. Expenses & Budget

| Method | Endpoint                              | Purpose                |
|--------|---------------------------------------|------------------------|
| GET    | /weddings/{wedding_id}/expenses       | List expenses          |
| POST   | /weddings/{wedding_id}/expenses       | Create expense         |
| GET    | /expenses/{expense_id}                | Get expense            |
| PATCH  | /expenses/{expense_id}                | Update expense/payment |
| DELETE | /expenses/{expense_id}                | Delete expense         |
| GET    | /weddings/{wedding_id}/budget/summary | Budget/spend/remaining |

## 9. Vendors

| Method | Endpoint                               | Purpose                   |
|--------|----------------------------------------|---------------------------|
| GET    | /vendors/search                        | Search Google Places/Maps |
| GET    | /weddings/{wedding_id}/vendors         | List wedding vendors      |
| POST   | /weddings/{wedding_id}/vendors         | Add vendor                |
| GET    | /vendors/{vendor_id}                   | Get vendor                |
| PATCH  | /vendors/{vendor_id}                   | Update vendor             |
| DELETE | /vendors/{vendor_id}                   | Remove vendor             |
| POST   | /vendors/{vendor_id}/events/{event_id} | Attach vendor to event    |
| DELETE | /vendors/{vendor_id}/events/{event_id} | Detach vendor             |

## 10. Invitations

| Method | Endpoint                              | Purpose                      |
|--------|---------------------------------------|------------------------------|
| GET    | /weddings/{wedding_id}/invitations    | List invitations             |
| POST   | /weddings/{wedding_id}/invitations    | Create invitation            |
| PATCH  | /invitations/{invitation_id}          | Update invitation            |
| POST   | /invitations/{invitation_id}/publish  | Publish invitation           |
| POST   | /weddings/{wedding_id}/members/invite | Send member email via Resend |

## 11. Photos & Documents

| Method | Endpoint                                 | Purpose                   |
|--------|------------------------------------------|---------------------------|
| POST   | /weddings/{wedding_id}/photos/upload-url | Get signed GCS upload URL |
| POST   | /weddings/{wedding_id}/photos            | Save photo metadata       |
| GET    | /weddings/{wedding_id}/photos            | List photos               |
| GET    | /photos/{photo_id}/download-url          | Get signed view URL       |
| PATCH  | /photos/{photo_id}                       | Update caption/visibility |
| DELETE | /photos/{photo_id}                       | Delete photo              |

| Method | Endpoint                                    | Purpose                 |
|--------|---------------------------------------------|-------------------------|
| POST   | /weddings/{wedding_id}/documents/upload-url | Get document upload URL |
| POST   | /weddings/{wedding_id}/documents            | Save document metadata  |
| GET    | /weddings/{wedding_id}/documents            | List documents          |
| DELETE | /documents/{document_id}                    | Delete document         |

## 12. Public APIs

Public endpoints expose only published/allowed wedding data or use a secure public token. Guests do not need accounts.

| Method | Endpoint                                | Purpose                     |
|--------|-----------------------------------------|-----------------------------|
| GET    | /public/weddings/{slug}                 | Public wedding website data |
| GET    | /public/weddings/{slug}/events          | Public events               |
| GET    | /public/weddings/{slug}/gallery         | Shared gallery              |
| GET    | /public/invitation/{token}              | Digital invitation          |
| POST   | /public/rsvp/{token}                    | Submit RSVP                 |
| POST   | /public/photo-upload/{token}/upload-url | Guest photo upload URL      |
| POST   | /public/photo-upload/{token}/photos     | Save guest photo metadata   |

## 13. Dashboard

GET /weddings/{wedding\_id}/dashboard — returns countdown, event summary, guest/RSVP summary, budget, tasks, shopping and vendor summary. This reduces multiple frontend calls for the main dashboard.

## 14. Standard API Response

Success: return the resource or result object. Error: use a consistent shape such as `{"error":{"code":"...","message":"..."}}`. List APIs support pagination and filters such as status, event\_id, category and search.

## 15. Authorization

For every private request FastAPI should authenticate the user, find their wedding membership, check their role/permission, and query using wedding\_id. Admin has full wedding management access; Manager and Member permissions can be restricted by operation. Public endpoints use controlled tokens/slugs.

## 16. Member Invitation Flow

Admin → POST /weddings/{id}/members/invite → FastAPI creates invitation → Resend sends link → Rahul opens link → Google login → FastAPI checks token + invited email → creates wedding\_members record with Manager/Member role.

## 17. Photo Upload Flow

Browser → request upload URL → FastAPI checks access → signed GCS URL → browser uploads directly to GCS → metadata saved in MongoDB. FastAPI never needs to receive the actual large image file.

## 18. Security

Use HTTPS, validate all input, enforce wedding-level authorization in FastAPI, rate-limit public RSVP/photo endpoints, validate upload type/size, use short-lived signed URLs, keep Google/Resend/MongoDB secrets outside source code, and log important backend errors.

## 19. V1 Scope & Future Changes

This is the current API baseline. Endpoint names, payload fields, session handling, pagination and individual APIs may change during implementation as the UI and database are built. The core rule remains: Next.js → FastAPI → MongoDB, with FastAPI as the business-logic and authorization boundary.