



## 2. Notes App

### Models

#### 1. users

#### ✓ User API

Method	Endpoint	Auth Required	Description
POST	/api/users/register	✗ No	Register a new user
POST	/api/users/login	✗ No	Log in a user and return JWT token
DELETE	/api/users/delete	✓ Yes	Delete the authenticated user account
GET	/api/users/logout	✗ (Client-side)	Logout handled on frontend (clear token)
PUT	/api/users/change-password	✓ Yes	Change the password of the authenticated user
PUT	/api/users/change-username	✓ Yes	Change the username of the authenticated user

#### User Data model

Field Name	Type	Required	Description
<code>_id</code>	ObjectId	Yes	Auto-generated unique identifier
<code>name</code>	String	Yes	Full name of the user
<code>email</code>	String	Yes	Unique email address of the user
<code>password</code>	String	Yes	Hashed password
<code>createdAt</code>	Date	Yes	Timestamp of user creation (auto)
<code>updatedAt</code>	Date	Yes	Timestamp of last update (auto)

## ✓ 2. Profile Model



### Profile APIs

Method	Endpoint	Auth Required	Description
<code>POST</code>	<code>/api/profiles</code>	✓ Yes	Create a profile for the authenticated user (if not already exists)
<code>GET</code>	<code>/api/profiles/me</code>	✓ Yes	Get the profile of the authenticated user
<code>GET</code>	<code>/api/profiles/:id</code>	✗ No	Get a profile by Profile ID (public view)
<code>PUT</code>	<code>/api/profiles/update</code>	✓ Yes	Update authenticated user's profile details (bio, profilePic, dob, location)
<code>PATCH</code>	<code>/api/profiles/profile-pic</code>	✓ Yes	Update only the profile picture
<code>DELETE</code>	<code>/api/profiles/delete</code>	✓ Yes	Delete the authenticated user's profile

### Profile Data model

Field	Type	Required	Description
<code>_id</code>	ObjectId	Yes	Profile ID
<code>user</code>	ObjectId	Yes	Ref to <code>User</code>

profilePic	String	No	URL/path to profile picture
bio	String	No	Short biography
dob	Date	No	Date of birth
location	String	No	(optional) Location info
createdAt	Date	Yes	Auto-generated
updatedAt	Date	Yes	Auto-generated

### 3. Notes

## ✓ Minimal Note Model – Fields

Notes → user ( relations )



### Note APIs

Method	Endpoint	Auth	Description
POST	/api/notes	✓	Create a new note
GET	/api/notes	✓	Get all notes for the authenticated user
GET	/api/notes/:id	✓	Get a specific note by its ID (if owned by user)
PUT	/api/notes/:id	✓	Update a note (title, content, tags, pinned, archived, starred, etc.)
DELETE	/api/notes/:id	✓	Delete a note by its ID
PATCH	/api/notes/:id/pin	✓	Toggle pinned state for a note
PATCH	/api/notes/:id/archive	✓	Toggle archived state for a note
PATCH	/api/notes/:id/star	✓	Toggle starred state for a note
POST	/api/notes/:id/upload	✓	📎 Upload a file to a note and store file URL in attachments[]
PATCH	/api/notes/:id/remove-attachment	✓	✍ Remove a specific attachment from

Method	Endpoint	Auth	Description
			attachments[]

## Notes Data models

Field Name	Type	Required	Description
Field Name	Type	Required	Description
_id	ObjectId	Yes	Auto-generated unique ID for each note
title	String	Yes	Title of the note
content	String	Yes	Main text content of the note
user	ObjectId	Yes	Reference to the User model (owner of the note)
tags	String[]	No	Array of tags/labels for organization
isPinned	Boolean	No	Whether the note is pinned
isArchived	Boolean	No	Whether the note is archived
isStarred	Boolean	No	Whether the note is marked as starred (default: false)
attachments	String[]	No	<input checked="" type="checkbox"/> Optional — Array of file URLs (Cloudinary or other storage)
createdAt	Date	Yes	Auto-generated timestamp on creation
updatedAt	Date	Yes	Auto-updated timestamp on modification