Patient Dashboard

GLP-1 based weight-loss solutions:

- **GLP-1** stands for *Glucagon-Like Peptide-1*
- **Acme Corp** is a fast-growing company.
- It focuses on **helping people lose weight** using **GLP-1 based medications** (like Ozempic, Wegovy, etc.), which are effective for controlling appetite and improving metabolism.
- The company already has **many users**, and that number is **increasing steadily**.
- Customers don't just get **prescription drugs** from Acme — they also receive **ongoing help and guidance**, such as:
  - Nutrition advice
  - Exercise plans
  - Progress tracking
  - Motivational or lifestyle support

The company wants to
improve the user experience for patients (people using their weight-loss service).

Patient Dashboard — a special online interface where users can:

- Track their weight-loss progress
- View medications, appointments, or health tips
- Get personalized updates or support

✅ Summary Table

Feature	Purpose	Example Component
Secure User Authentication	Protect patient data	Login/Register Page
Dashboard Overview	Show quick health & shipment summary	Summary cards
Weight Loss Progress	Help users visualize their journey	Graph, BMI indicator
Medication & Shipment Tracking	Keep users informed on what they're taking and when it's arriving	

✅ All Modules & Their MongoDB Data Models (Patient Dashboard Only)

🔷 1. User Collection (Authentication Info)

✅ 🔐 User Authentication APIs

API Route	Method	Protected	Purpose
`/api/auth/register`	POST	❌ No	Register a new user
`/api/auth/login`	POST	❌ No	Authenticate user and return token
`/api/auth/logout`	POST	✅ Yes	Clear user session (on frontend/local)
`/api/auth/me`	GET	✅ Yes	Get currently logged-in user's info
`/api/auth/delete`	DELETE	✅ Yes	Delete account (authenticated user)

Field	Type	Required	Description
`_id`	ObjectId	Yes	Auto-generated by MongoDB

`email`	String	Yes	Unique email for login
`password`	String	Yes	Hashed password for authentication
`createdAt`	Date	No	Timestamp when account was created

🔷 2. Profile Collection (Personal & Health Info)

API Route	Method	
`POST /api/profile`	Create a profile after user signs up	
`GET /api/profile`	Fetch the logged-in user’s profile	
`PUT /api/profile`	Update profile info (name, goal, etc.)	

Field	Type	Required	Description
`_id`	ObjectId	Yes	Auto-generated by MongoDB
`userId`	ObjectId (ref)	Yes	Reference to `User._id` (1-to-1 relation)
`name`	String	Yes	Patient's full name
`age`	Number	No	Patient's age
`gender`	String	No	Male, Female, Other
`goalWeight`	Number	Yes	Patient’s weight-loss goal (in kg/lbs)
`updatedAt`	Date	No	Timestamp when profile was last updated

✅ 3. Weight Progress Module

🔷 Purpose:

Tracks the user's weight over time and optionally their BMI, for visualizing progress (charts, goal tracking, etc.)

✅ 📉 Weight Progress APIs

API Route	Method	Protected	Purpose
`/api/weight`	POST	✅ Yes	Add a new weight entry
`/api/weight`	GET	✅ Yes	Get all weight entries for logged-in user (timeline/graph)
`/api/weight/latest`	GET	✅ Yes	Get the latest weight entry
`/api/weight/:entryId`	GET	✅ Yes	Get a single entry by ID
`/api/weight/:entryId`	PUT	✅ Yes	Update a specific weight entry
`/api/weight/:entryId`	DELETE	✅ Yes	Delete a specific weight entry

🧠 Bonus: Optional Analytics APIs

API Route	Method	Purpose
`/api/weight/summary`	GET	Return progress summary (e.g. total lost)
`/api/weight/monthly-avg`	GET	Return monthly averages for graphing

🔶 `WeightEntry` Model (MongoDB/Mongoose)

Field	Type	Required	Notes
`_id`	ObjectId	Yes	Auto-generated by MongoDB
`userId`	ObjectId (ref)	Yes	Reference to `User._id`
`weight`	Number	Yes	User's weight at this entry
`BMI`	Number	No	Optional: Calculated from weight, height (in profile)

`date`	Date	Yes	Date of entry (default: now)

✅ 4 Medication or Product Model

This model represents the medication a user is currently using or has used in the past

It supports tracking type, dosage, duration, and linkage to shipments.

🔷 Medication Model (MongoDB/Mongoose)

✅💊 Medication API Routes

API Route	Method	Protected	Purpose
`/api/medications`	POST	✅ Yes	Add a new medication entry
`/api/medications`	GET	✅ Yes	Get all medications for logged-in user
`/api/medications/active`	GET	✅ Yes	Get currently active medication
`/api/medications/:id`	GET	✅ Yes	Get a specific medication by ID
`/api/medications/:id`	PUT	✅ Yes	Update medication info (dosage, etc.)
`/api/medications/:id`	DELETE	✅ Yes	Delete a medication entry

Field	Type	Required	Description
`_id`	ObjectId	Yes	Auto-generated by MongoDB
`userId`	ObjectId (ref)	Yes	Refers to `User._id`
`name`	String	Yes	Name of the medication (e.g., Ozempic)
`dosage`	String	Yes	Dosage detail (e.g., "0.5mg weekly")
`startDate`	Date	Yes	Date when the user started the medication

endDate	Date	No	Date when medication was stopped (if any)
notes	String	No	Optional description or remarks
createdAt	Date	No	Auto-set creation timestamp

? Whether `/api/medications` is added by the **patient** or **doctor** depends entirely on the **role-based access control** (RBAC)

✅ Scenario 1:
Patient Adds Medication ✅
✅ Scenario 2:
Doctor Adds Medication

✅ 5. Shipment (Order) Module

✅ 🗂️ Order Model (formerly Shipment)

✅ 🔁 Updated Orders API Routes

API Route	Method	Protected	Purpose
/api/orders	POST	✅ Yes	Create new order (admin/doctor only)
/api/orders	GET	✅ Yes	Get all orders for logged-in patient
/api/orders/upcoming	GET	✅ Yes	Get next upcoming order
/api/orders/:id	GET	✅ Yes	Get a specific order by ID
/api/orders/:id	PUT	✅ Yes	Update status, tracking, ship date
/api/orders/:id	DELETE	✅ Yes	Delete an order (if necessary)

Field	Type	Required	Description
_id	ObjectId	Yes	Auto-generated

`userId`	ObjectId (ref)	Yes	Refers to `User._id`
`medicationId`	ObjectId (ref)	Yes	Refers to `Medication._id`
`status`	String	Yes	`"pending"`, `"shipped"`, `"delivered"`
`trackingNumber`	String	No	External courier tracking number
`expectedDate`	Date	Yes	Estimated delivery date
`shippedDate`	Date	No	Date medication was shipped
`createdAt`	Date	No	Auto-set timestamp

MongoDB Schema Relationships (Finalized)

✅ 1. users Collection

Stores authentication details.

```
json
CopyEdit
{
  _id: ObjectId,          // Primary Key
  email: String,          // Unique, required
  password: String,       // Hashed
  createdAt: Date
}
```

✅ 2. profiles Collection

Stores patient's personal info and health goals — **1:1 relationship with users** via `userId`.

```
json
CopyEdit
{
  _id: ObjectId,
  userId: ObjectId,       // Reference to users._id
  name: String,
  age: Number,
  gender: String,
  goalWeight: Number,
  updatedAt: Date
}
```

✅ 3. weightentries Collection

Each weight record belongs to a user — **1:N relationship with users**.

```
json
CopyEdit
{
  _id: ObjectId,
  userId: ObjectId,       // Reference to users._id
  weight: Number,         // e.g., 75
  BMI: Number,            // Optional
  date: Date              // Default: now
}
```

✅ 4. medications Collection

Stores user's medication info — **1:N with users**.

```
json
CopyEdit
{
  _id: ObjectId,
  userId: ObjectId,       // Reference to users._id
  name: String,           // e.g., "Ozempic"
  dosage: String,         // e.g., "0.5mg weekly"
  startDate: Date,
  endDate: Date,          // Optional
  notes: String,
  createdAt: Date
}
```

✅ 5. orders Collection

Stores shipment/order info — references both the `userId` and `medicationId`.

```
json
CopyEdit
{
  _id: ObjectId,
  userId: ObjectId,          // Reference to users._id
  medicationId: ObjectId,    // Reference to medications._id
  status: String,            // "pending" | "shipped" | "delivered"
  trackingNumber: String,
  expectedDate: Date,
  shippedDate: Date,
  createdAt: Date
}
```

✅ Dashboard Overview — What to Show

1. 🔢 Current Weight

- The most recent weight entry

- Optional: Show BMI + goal difference

2. 📦 Next Shipment Date

- From the upcoming order
- Show `expectedDate`, `status`, `trackingNumber`

3. 📈 Progress Snapshot

- Total weight lost
- Progress toward goal (start → current → goal)
- Optionally: progress in last 30 days

✅ Required APIs to Call (from existing ones)

Feature	API Route	Purpose
Current weight	`GET /api/weight/latest`	Get latest weight entry
Goal weight (and name)	`GET /api/profile`	Get goal weight + user name
Progress summary	`GET /api/weight/summary`	(Optional) Total lost, % to goal
Upcoming order	`GET /api/orders/upcoming`	Get next order/shipment info
Current medication	`GET /api/medications/active`	Show what medication they're on

📊 Example Dashboard Layout (Component-wise)

```
tsx
CopyEdit
📍 Dashboard Summary
----------------------------------------
👤 Hello, [User's Name]!

⚖️ Current Weight: 76 kg
    - BMI: 23.5
    - Goal: 68 kg
    - You've lost 4.5 kg so far! 🎯
```

```
📦 Next Shipment:
   - Medication: Ozempic
   - Status: Shipped ✅
   - ETA: July 10, 2025
   - Tracking: XYZ123456

📈 Progress Overview:
   - Start Weight: 80 kg
   - Current Weight: 76 kg
   - Goal Weight: 68 kg
   - 🔻 Lost: 4.0 kg (50% to goal)
```

💡 How to Handle It in Code

💻 Backend (Optional Composite API)

You can build one **combined API** to return all at once:

```
pgsql
CopyEdit
GET /api/dashboard/summary
```

✅ Tech Stack (Patient Dashboard Project)

🔷 Frontend: Next.js (React-based)

Tool/Lib	Purpose	
Next.js	React-based framework with routing, SSR	
Tailwind CSS	Utility-first styling (fast + clean UI)	

Axios / Fetch	HTTP requests to backend APIs	
Recharts	For weight progress charts	
shadcn		

🔷 Backend:

Node.js + Express.js + MongoDB

Tool/Lib	Purpose
Node.js	JavaScript runtime for server-side logic
Express.js	API framework for building REST routes
MongoDB	NoSQL document database
Mongoose	ODM for schema modeling and validation
JWT	Authentication token for secure access
bcryptjs	Password hashing for user accounts
dotenv	Manage environment variables
helmet	Secure HTTP headers