💪 FitTrack – Full Project Roadmap (Step-by-Step)

✅ Phase 1 – Project Setup

- Create your project folder: fittrack
- Inside VS Code terminal:
    - Run npx create-react-app client to create the frontend.
    - Create a server folder with npm init -y to initialize backend.
- Install required backend packages:

nginx

CopyEdit

npm install express mongoose cors dotenv

- Create folders inside server:

CopyEdit

routes/

controllers/

models/

- Setup MongoDB connection with mongoose in server.js.
- Add "proxy": "http://localhost:5000" in client/package.json.

📌 Why?
You separate frontend (client) and backend (server) like real-world scalable apps.

✅ Phase 2 – User Authentication (Firebase)

- Go to firebase.google.com → Create a new project.
- Enable **Email/Password** authentication.
- Inside client, install Firebase:

nginx

CopyEdit

```
npm install firebase
```

- Create firebase.js in src/:
  - Initialize Firebase app using config from console.
  - Export Firebase Auth instance using getAuth().
- Build Signup and Login pages:
  - Use createUserWithEmailAndPassword() for Signup.
  - Use signInWithEmailAndPassword() for Login.
  - Add success/error messages.
- Add routing with react-router-dom:

nginx

CopyEdit

```
npm install react-router-dom
```

📌 Why?
Firebase handles secure authentication so you don’t need to store passwords manually.

✅ Phase 3 – Dashboard UI (Responsive)

- After login, navigate to /dashboard page.
- Create Dashboard.js with:
  - Welcome message
  - Logout button
  - Sidebar (for navigation)
  - Topbar (for user info)
- Ensure full responsiveness:
  - Mobile: Sidebar collapses
  - Desktop: Full layout

📌 Why?
Good UX matters. Responsive design works across all screen sizes.

✅ Phase 4 – Workout CRUD Features

- Create a Workout model (MongoDB schema):
  - Fields: name, category, sets, reps, media, date
- Add API routes:

    - POST /api/workouts – Create new workout
    - GET /api/workouts – Get all workouts
    - PUT /api/workouts/:id – Update workout
    - DELETE /api/workouts/:id – Delete workout
- Use axios or fetch() in frontend to connect.
- Show all workouts in Dashboard with edit/delete buttons.

📌 Why?
This is the core of your fitness tracker — managing daily workouts.

✅ **Phase 5 – Media Integration (Images/Videos)**

- Option 1: Use YouTube video links for exercises.
- Option 2: Use Multer in Node.js to upload images to /uploads.
- Option 3: Use Firebase Storage to upload and fetch images/videos.
- Display previews in the workout card UI.

📌 Why?
Visual media improves user engagement and understanding of exercises.

✅ **Phase 6 – Progress & Stats with Charts**

- Install chart.js or recharts:

nginx

CopyEdit

npm install chart.js react-chartjs-2

- Show:
  - Bar chart for weekly workout counts
  - Line chart for weight/steps progression
- Fetch workout logs from backend and process into chart data.

📌 Why?
Progress visualization motivates users and improves retention.

✅ **Phase 7 – Email Reminders (Optional)**

- Use nodemailer in backend:

nginx

CopyEdit

npm install nodemailer

- Use a cron job or schedule emails using node-cron:

nginx

CopyEdit

npm install node-cron

- Send motivational emails or reminders every morning at 7 AM.

📌 Why?
Keeps users engaged and reminds them to stay consistent.

✅ Phase 8 – Dark Mode + Theme Toggle

- Add a toggle button in the topbar.
- Use React state or localStorage to remember preference.
- Apply CSS class to root <div> like className={darkMode ? 'dark' : ''}.

📌 Why?
Gives users visual comfort and modern touch.

✅ Phase 9 – Deployment

- Deploy frontend to **Vercel** or **Netlify**:
  - Connect GitHub repo
  - Auto deploy on push
- Deploy backend to **Render**, **Railway**, or **Cyclic**
  - Add environment variables (.env)
  - MongoDB URI, Firebase keys, etc.
- Use Postman to test live APIs
- Test your website on phone + desktop

📌 Why?
Make your project public, share with recruiters, and use it live!

🎁 Bonus – Templates & Google Calendar Sync

- Create exercise templates like:
  - Push Day
  - Pull Day
  - Leg Day
- Add “Save as Template” feature in workout creation
- (Optional) Use Google Calendar API to sync workouts as calendar events:

nginx

CopyEdit

npm install googleapis

📌 Why?
Professional gyms and trackers use templates & calendar sync.

✅ Final Note:

This roadmap is designed to **level you up** from:

Beginner React Developer → Real World Full Stack Dev

Every feature teaches you new skills:

- Firebase → Auth
- Express + MongoDB → Backend APIs
- Chart.js → Data Visualization

- Deployment → Going public
- Responsiveness → Real UI/UX skill