Mini Buyer Lead Intake App - Assignment Solution

1) Project layout

```
buyer-leads/
■■ app/
■  ■■ leads/
■  ■  ■■ page.tsx
■  ■  ■■ create/page.tsx
■  ■  ■■ [id]/page.tsx
■  ■  ■■ [id]/edit/page.tsx
■  ■■ import/
■  ■  ■■ page.tsx
■  ■■ api/
■     ■■ leads/
■     ■  ■■ route.ts
■     ■  ■■ import/route.ts
■     ■■ auth/route.ts
■■ lib/
■  ■■ prisma.ts
■  ■■ zod-schemas.ts
■  ■■ auth.ts
■  ■■ rateLimit.ts
■  ■■ csv.ts
■■ prisma/
■  ■■ schema.prisma
■  ■■ migrations/...
■■ tests/
■  ■■ csv-validator.test.ts
■■ package.json
■■ tsconfig.json
■■ README.md
```

2) Prisma Schema

```
generator client {
  provider = "prisma-client-js"
}
...
```

3) Zod Schema

```
import { z } from "zod";

export const BuyerCreateSchema = z.object({
  name: z.string().min(2).max(80),
  phone: z.string().min(10).max(15).regex(/^\d+$/, "Phone must be numeric"),
  ...
});
```

4) Create Lead Route

```
export async function POST(req: Request) {
  const user = await getUserFromReq(req);
  if (!user) return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
  ...
}
```

5) Edit Lead Route

```
export async function PATCH(req: Request) {
  const user = await getUserFromReq(req);
  if (!user) return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
  ...
}
```

6) SSR List Page

```
export default async function LeadsPage({ searchParams }) {
  const page = Number(searchParams?.page ?? 1);
  ...
}
```

7) View & Edit Pages

View page shows buyer details and history. Edit page has optimistic concurrency control.

8) CSV Import / Export

```
export async function POST(req: Request) {
  const user = await getUserFromReq(req);
  if (!user) return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
  ...
}
```

9) Auth & Ownership

Demo auth with cookie, ownership check on edit/delete, read allowed for all logged-in users.

10) Rate limiting

```
export async function rateLimit(key: string) {
  const now = Date.now();
  ...
}
```

11) Unit test Example

```
describe("CSV row validator", () => {
  it("accepts valid row", () => { ... });
  it("rejects short name", () => { ... });
});
```

12) Accessibility & Error boundary

Add labels, aria-invalid, role='alert' for form errors, focus management, error boundaries.

13) Nice-to-haves

Tag chips with typeahead, status quick-actions, basic full-text search.

14) README

```
# Buyer Lead Intake (mini)
## Setup
- Clone repo
- npm install
- npx prisma migrate dev
- npm run dev
...
```

15) Commit History

```
git commit -m "chore: initial project scaffold"
git commit -m "feat(db): add Prisma schema"...
```

16) Quick Tips

Use Postgres by changing DATABASE_URL. Arrays differ in Postgres vs SQLite. Redis for real rate limiting.