

Overview

This is the Next.js 14 App Router frontend for Bet On Talent. It uses TypeScript, React 18, TailwindCSS, and a small UI primitives layer. Business logic lives in controller and service files, while network access is isolated in `src/models` and `src/services`.

Authentication is performed against `POST /auth/login`. A returned token is stored in `sessionStorage` and then attached as `Authorization: Bearer <token>` on subsequent requests via a centralized HTTP client.

Getting Started

- Node 18+ and npm 9+
- Install deps: `npm ci`
- Run dev server: `npm run dev` → `http://localhost:3000`
- Build for production: `npm run build`
- Start production server: `npm start`
- Lint / Format: `npm run lint` / `npm run format`

Environment

- Local: set `.env.local` → `NEXT_PUBLIC_API_BASE_URL=http://localhost:3001`
- Production: set `.env.production` → `NEXT_PUBLIC_API_BASE_URL=https://api.example.com`
- In production, `NEXT_PUBLIC_API_BASE_URL` is *required* and a missing value will throw during boot.

Defaults

<code>APP_CONFIG.name</code>	Bet On Talent
<code>APP_CONFIG.env</code>	<code>process.env.NODE_ENV</code> OR <code>development</code>
<code>APP_CONFIG.apiBaseUrl</code>	<code>process.env.NEXT_PUBLIC_API_BASE_URL</code> if set; otherwise in development defaults to <code>http://localhost:3001</code> ; in production it must be provided.
<code>APP_CONFIG.graphStroke</code>	<code>1</code>
HTTP default headers	<code>{ 'Content-Type': 'application/json' }</code>
HTTP timeout	<code>15000</code> ms per request
401 handling	Global redirect to <code>/not-authorized</code> for all routes except the login request itself
Session key	<code>auth_token</code> in <code>sessionStorage</code>
Theme	System default via <code>next-themes</code> ; toggle between light/dark

Scripts

- `npm run dev` — start Next.js in development
- `npm run build` — build production bundle
- `npm start` — run production server
- `npm run lint` — ESLint for `.ts/.tsx`
- `npm run format` — Prettier write
- `npm test` — run Vitest suite
- `npm run test:watch` — Vitest watch mode

Testing

We use Vitest with `@testing-library/react` and `jsdom`.

- Run all: `npm test`
- Watch: `npm run test:watch`
- Single file: `npx vitest run tests/auth/login.page.test.tsx`

Current tests

- `tests/auth/login.page.test.tsx` — verifies the Login page UI (title and button)
- `tests/main/manage.view.test.tsx` — verifies Manage header renders

Vitest setup mocks Next.js navigation in `vitest.setup.ts` to make UI components test-friendly.

Directory Structure

- `src/app` — App Router pages and layouts
- `src/app/(auth)` — auth routes (`/login`, `/not-authorized`)
- `src/app/(main)` — main layout with top action bar and manage view
- `src/app/(main)/widgets/top-form` — participant creation form, theme toggle, logout
- `src/app/(main)/manage` — table and donut graph widgets
- `src/components/ui` — reusable UI primitives (e.g., `Button`)
- `src/models` — typed API models (`users`, `participants`)
- `src/services/http` — `HttpClient`, types, and events
- `src/services/storage` — session storage helpers
- `src/services/event` — lightweight event bus
- `src/utils` — helpers (`cn` class merge)
- `src/config.ts` — centralized runtime config
- `src/strings.ts` — copy/labels in one place
- `src/routes.ts` — route constants

API Usage

- `POST /auth/login` — `body { email, password } → returns { access_token | token }`
- `GET /participants` — returns array of participants
- `POST /participants` — create participant `{ firstName, lastName, participation }`
- `DELETE /participants/:id` — delete by id

Import the Postman collection at `docs/postman_collection.json` for quick API testing.

Auth & Navigation

- Token is persisted to `sessionStorage` under `auth_token`
- All requests include `Authorization: Bearer <token>` if present
- Global 401 handler redirects to `/not-authorized` (login failures remain on page)
- Logout clears token and navigates to `/login`

Styling & Theming

- TailwindCSS with CSS variables defined in `src/app/globals.css`

- Dark mode via `next-themes` (class strategy, system default)
- Toggle provided in the top form widget

Config & Aliases

- React Strict Mode is disabled in `next.config.js`
- TS path aliases like `@/app/*`, `@/components/*`, etc. set in `tsconfig.json`
- Vitest config at `vitest.config.ts` with JSDOM and setup file

Development Notes

- Controllers (`*.controller.ts[x]`) manage state and side-effects; views (`*.view.tsx`) remain presentational
- HTTP client centralizes base URL, headers, timeouts, and lifecycle events
- Event bus enables simple cross-widget communication (e.g., notifying Manage after a new participant is created)
- Table supports optimistic delete; graph recalculates slices based on total participation

Troubleshooting

- Empty data on Manage → verify API base URL and that the backend participants endpoints are reachable
- Build error in production → set `NEXT_PUBLIC_API_BASE_URL` in `.env.production`
- Want to reset session quickly? Clear `sessionStorage` or use the built-in Logout