Web Roadmap Overview

Common Track: Web Foundations for Absolute Beginners

Workshop 1: How the Web Works

Key Topics:

- **Internet Basics**: "The web is like a global postal system for computers."
- **Browsers & Servers**: Requests/responses analogy (e.g., ordering food delivery).
- **IP Addresses & DNS**: IPs as "GPS coordinates," DNS as "translator for website names."
- **HTTP/HTTPS**: Secure vs. insecure connections (padlock icon explanation).
- **Activity**: Use Chrome DevTools to inspect network requests for YouTube.com.

Workshop 2: HTML & CSS Basics

Key Topics:

- **HTML Essentials**: Tags (`<div>` , `<h1>`), forms, lists, semantic elements.
- **CSS Fundamentals**: Box model, flexbox basics, external stylesheets.
- **Responsive Design**: Media queries for mobile/desktop layouts.
- **Activity**: Build a recipe page with images, ingredients list, and styled headings.

Workshop 3: Introduction to JavaScript

Key Topics:

- **JS Role**: "The brain that adds logic to websites."
- **Variables & Operators**: `let` vs. `const` , arithmetic/logical operators.
- **Functions & Events**: Basic click handlers, `alert()` interactions.
- **DOM Basics**: Change text/styles with `getElementById` .
- **Activity**: Create a "dark mode" toggle button for a webpage.

Workshop 4: Advanced JavaScript (Frontend/Backend Basics)

Key Topics:

- **ES6+ features:** template literals, destructuring.
- **Array methods:** `map`, `filter`, `reduce`.
- **Closures & Scope**: Variable accessibility in functions.
- **Async/Await**: Fetching data without freezing the UI.
- **Modules**: `import/export` for code organization.
- **Error Handling**: `try/catch` for API failures.
- **Activity**: Build a weather widget using the OpenWeather API.

Workshop 5: Introduction to Databases

Key Topics:

- **Database Types**: SQL (Excel-like tables) vs. NoSQL (JSON-like documents).
- **CRUD Operations**: Create, read, update, delete (practice with a mock library system).
- **Relationships**: One-to-many (e.g., users → blog posts).
- **Activity**: Design a database for a Twitter-like app (users, tweets, likes).

Workshop 6: Frontend/Backend Roles & API Communication

Key Topics:

- **Frontend**: UI building, form validation, DOM updates.
- **Backend**: Server logic, database connections, security.
- **APIs**: REST endpoints as "waiters" between frontend/backend.
- **JSON Structure**: Nesting objects/arrays for complex data.
- **Activity**: Diagram a login flow: frontend form → backend validation → database check.

Workshop 7: Git Basics & Real-World Collaboration (After front and back tracks)

Key Topics:

- **Git Workflow**: `clone`, `branch`, `commit`, `push`, `pull`.
- **GitHub Features**: Pull requests, code reviews, issues.
- **Team Conventions**: Commit messages, branching strategies.
- **Activity**: Collaborate on a group project (e.g., shared calendar app) using forks/PRs.

Workshop 8: Switching from JavaScript to TypeScript (After front and back tracks)

Key Topics:

- **Why TypeScript?**: Catching bugs early with static typing.
- **Basic Types**: `string`, `number`, `boolean`, custom interfaces.
- **Type Annotations**: Functions, arrays, and objects.
- **TS Config**: Compiling to JavaScript.
- **Activity**: Convert a JavaScript calculator app to TypeScript.

Frontend Track: From Zero to Frontend Hero

Workshop 1: Advanced CSS & Preprocessors

Key Topics:

- CSS Flexbox/Grid mastery.
- Responsive design with media queries.
- CSS frameworks (Tailwind CSS/Bootstrap).
- Preprocessors: SASS/SCSS (variables, mixins, nesting).
- **Activity**: Build a responsive e-commerce product card using Tailwind/SASS.

Workshop 2: Advanced DOM Manipulations & Events

Key Topics:

- Event delegation/bubbling/capturing.
- Dynamic DOM element creation/removal.
- Local/Session Storage for state persistence.
- Drag-and-drop API basics.
- **Activity**: Create an interactive kanban board with draggable tasks.

Workshop 3: Intro to JS Frameworks (React)

Key Topics:

- React core concepts: components, JSX, virtual DOM.
- Props vs. state.
- Functional components.
- React Developer Tools.
- **Activity**: Build a static portfolio site with React components.

Workshop 4: Mastering React Hooks (Advanced State & Side Effects)

Key Topics:

- Core Hooks: `useState`, `useEffect`, `useContext`, `useReducer`, `useRef`.
- Performance Hooks: `useMemo`, `useCallback`, `React.memo`, `useTransition`.
- Custom Hooks: Design patterns for reusable logic (e.g., data fetching, form handling).
- Edge Cases: Hook dependencies, cleanup functions, avoiding stale closures.
- **Activity**: Build a real-time task manager with drag-and-drop reordering, animations, and a backend sync system (uses `useReducer` for complex state, `useEffect` for API calls, `useContext` for shared themes/auth, and custom hooks for WebSocket connections).

Workshop 5: Data Fetching & Backend Communication

Key Topics:

- REST vs. GraphQL (with Apollo Client).
- Authentication (JWT, cookie-based sessions).
- Error handling and loading states.
- File uploads with `FormData`.
- **Activity**: Build a social media feed with real API integration.

Workshop 6: React 19 Latest Features

Key Topics:

- Server Components architecture.
- React Actions for form handling.
- New hooks (`use`, `useOptimistic`).
- Asset loading improvements.
- **Activity**: Migrate a class component app to React 19 features.

Workshop 7: Intro to Next.js

Key Topics:

- App Router vs. Pages Router.
- Server-Side Rendering (SSR) / Static Site Generation (SSG).
- API routes for full-stack development.
- Deployment with Vercel.
- **Activity**: Build a blog with markdown support using Next.js.

Workshop 8: Frontend Optimization & Best Practices

Key Topics:

- Lighthouse performance audits.
- Code splitting with dynamic imports.
- Image optimization (WebP, lazy loading).
- SEO and accessibility (ARIA labels).
- **Activity**: Audit and optimize a slow React app (e.g., reduce CLS, FCP).

Backend Track: From Zero to Backend Hero

Workshop 1: Intro to Express (Routes & Middleware)

Key Topics:

- Node.js fundamentals and npm packages.
- Express.js setup and RESTful routing.
- Middleware concepts (logging, error handling, `next()`).
- **Activity**: Build a bookstore API with CRUD endpoints for books.

Workshop 2: ORMs & ODMS (Drizzle & Mongoose)

Key Topics:

- Relational DBs with Drizzle (PostgreSQL).
- NoSQL with Mongoose (MongoDB).
- Schema design and migrations.
- Transactions and validation.
- **Activity**: Create a user + product management system using both Drizzle and Mongoose.

Workshop 3: Authentication, Authorization & RBAC

Key Topics:

- JWT authentication flow.
- Password hashing (bcrypt, argon2).
- Role-based access control (admin/user/guest).
- OAuth2 basics (Google/GitHub login).
- **Activity**: Build a blog API where admins delete posts, users comment.

Workshop 4: Advanced APIs (Filtering, Pagination, File Handling)

Key Topics:

- Query filtering (`?category=tech&price_gte=100`).
- Pagination with `limit` / `offset` or cursor-based methods.
- File uploads (Multer for Express, S3 integration).
- **API Documentation**: OpenAPI/Swagger specs, Postman collections, automated docs generation.
- **Activity**: Develop a product catalog API with image uploads, search filters, and Swagger documentation.

Workshop 5: WebSockets & Real-Time Communication

Key Topics:

- Socket.io fundamentals (events, rooms).
- Real-time use cases (chat, live updates).
- Scaling WebSockets (Redis adapter).
- **Activity**: Create a collaborative whiteboard app with real-time sync.

Workshop 6: Caching & Security (Redis Cache)

Key Topics:

- Redis for caching (session storage, API responses).
- Security headers (CORS, CSP).
- Rate limiting and DDoS prevention.
- **Activity**: Cache high-traffic API endpoints and implement rate limits.

Workshop 7: Intro to GraphQL

Key Topics:

- GraphQL vs REST comparison.
- Schemas, resolvers, and queries/mutations.
- Apollo Server setup with Express.
- **API Documentation**: Leveraging GraphQL's self-documenting schema and tools like GraphiQL.
- **Activity**: Convert a RESTful e-commerce API to GraphQL and generate schema docs.

Workshop 8: Backend Optimization & Best Practices

Key Topics:

- Database indexing and query profiling.

- Load balancing with Nginx/PM2.
- Logging (Winston) and monitoring (Prometheus).
- **API Documentation Best Practices**: Versioning, changelogs, and developer portals.
- **Activity**: Optimize a legacy API's response time by 50% and document performance improvements.