

Full-Stack Product Development Bootcamp (Web + Mobile)



Overview

Master the skills necessary to become a successful full-stack product developer using the PERN stack (PostgreSQL, Express, React, Node.js) with React Native for mobile. Learn how to build NestBoard, a co-living booking and space management platform, by developing web and mobile user interfaces, creating APIs, implementing server-side business logic, managing relational data, and deploying production-ready applications across both platforms. This program is designed as a comprehensive journey where students learn concepts as solutions to real-world problems encountered during the development of a two-sided product ecosystem.



Learning Objectives

- **Build Dynamic Web Applications:** Create responsive vendor dashboards with React 19. Manage state using hooks and connect to APIs to deliver interactive user experiences.
- **Develop Cross-Platform Mobile Applications:** Build mobile apps with React Native. Implement navigation, device integrations, and offline-ready features for tenant-facing apps.
- **Create and Consume RESTful APIs:** Use Express.js to design and build REST APIs that handle property management, booking logic, and dashboard data.
- **Design and Manage Relational Databases:** Use PostgreSQL with Prisma ORM. Define schemas, run CRUD operations, and build queries for revenue and occupancy insights.
- **Authenticate and Authorize Users:** Implement secure authentication with Firebase Authentication. Apply role-based access control to manage vendor and tenant access.
- **Implement Payment Flows:** Integrate Stripe to process booking payments, manage secure checkouts, and store transaction records.
- **Visualize Business Metrics:** Build dashboards with charts to display revenue trends, booking analytics, and occupancy data.
- **Integrate Device Capabilities:** Use camera for QR scanning, add maps for property discovery, and enable geolocation-based search.
- **Build Offline-Ready Mobile Apps:** Use AsyncStorage for local data, detect network status, and handle connectivity changes smoothly.
- **Deploy and Maintain Applications:** Deploy web apps with Vercel, host APIs on Railway, and test mobile builds through APK sideloading and TestFlight.
- **Leverage AI for Engineering Productivity:** Use Claude Code, GitHub Copilot, and AI tools to refactor code, generate documentation, and speed up development.

Program Information

Estimated Time

4 months at 15hrs/week*
3-6h session
12h self-practice

Skill Level

Beginner

Medium

English

Session Delivery

Live Delivered

*Recordings available

Required Software

- VS Code: <https://code.visualstudio.com/>
- Android Studio: <https://developer.android.com/studio>
- PostgreSQL: <https://www.postgresql.org/>
- Postman: <https://www.postman.com/>
- Vercel: <https://vercel.com/>
- Prisma: <https://www.prisma.io/>
- NodeJS - <https://nodejs.org/en/download>
- Git - <https://git-scm.com/downloads>
- Cursor: <https://www.cursor.com/>

(The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 10-12 hours per week working through the program, you should finish within the time provided. Actual hours may vary.)



Program Flow

Main Track - Web

Phase 01 : Web app Frontend development
6 lessons

Each **Saturday** From April 04 - May 09



Phase 02: Web app Backend development
6 lessons

Each **Saturday** From May 16 - June 20



Phase 03 : Web integration & deployment
3 lessons

Each **Saturday** From June 27 - July 11

Main Track - Mobile

Phase 01: Mobile essentials
6 lessons

Each **Sunday** From May 03 - June 5



Phase 02 : Mobile integration & deployment
3 lessons

Each **Sunday** From June 28 - July 26



Common Track

Orchestration and quality
2 lessons (combined)

Each **Sunday** From July 18 - July 25

Sub Tracks

Session 1: Micro-frontend architecture

May 19, Tuesday

Session 2: Product innovation

June 09, Tuesday

Bonus Tracks

Career coaching and job readiness

Main Track - Web

Web Application Development

- Front-End
- Back-End
- Integration and Deployment



Foundations of Frontend Web Application Development

In this phase, you will build a strong foundation in JavaScript and React. You will learn core JavaScript concepts needed for modern web development, then progress to building interactive user interfaces with React 19. This includes learning about JSX, components, props, state, lifecycle management, and TypeScript for type-safe development.



Project

NestBoard Vendor Dashboard Frontend

You will develop the foundational structure of the NestBoard vendor dashboard by implementing basic layout elements and reusable components. You will use state manipulation to filter and display property data dynamically, define TypeScript interfaces for all core entities, and render the dashboard shell with mock data.

Lesson 1

Core JavaScript

- **Basic Programming Concepts with JavaScript:** Understand variables, data types, control structures, and functions for efficient coding and data management.
- **Manipulating the DOM:** Learn to select, modify, and create HTML elements, and handle events for interactive web pages.
- **JavaScript Operators and Array Methods:** Use operators for calculations and logic, and array methods like map, filter, and reduce for data manipulation.
- **Concurrency with Promises & async/await:** Manage asynchronous operations using promises and async/await for handling API calls and time-based logic.

Lesson 2

React Foundations & Project Setup

- **Introduction to ReactJS:** Introduce what ReactJS is and why it is required for building modern, interactive user interfaces. Compare React's component model to traditional DOM manipulation.
- **Creating the React App:** Begin by setting up a new React project using Vite. Explain the folder structure and the purpose of key files like main.jsx and App.jsx.
- **Introduction to JSX:** Introduce JSX syntax and demonstrate how it allows embedding HTML within JavaScript. Create a simple component to display a welcome message.
- **Static Components and Styling:** Create static UI components using JSX. Learn the basics of styling using CSS modules and inline styles. Build simple banner and card layout components.

Lesson 3

Components, Props & Stateful UI

- **Component Creation:** The concept of reusable components by building the NestBoard dashboard. Explain component structure, composition, and the separation of concerns.
- **Props:** Pass data to components using props. Use hardcoded property and room category data to populate dashboard cards initially.
- **State Management with useState:** The useState hook to manage component state. Create functionality for toggling property active status and filtering room categories based on user interactions.
- **Rendering Lists and Dynamic JSX:** Create a PropertyCardList component that renders multiple property cards. Teach how to loop through an array of properties and pass data to each card dynamically using the map function.
- **Deriving from State and Lifting State Up:** Show how values like occupancy percentages and total room counts can be derived from state. Demonstrate lifting state up to a parent component for coordinating filters across property and room views.

Lesson 4

Routing, Side Effects & Data Fetching

- **Routing with React Router v7:** Understand the basic concepts of client-side routing and integrate React Router into the project. Set up routes for the dashboard home, property detail, and room category views.
- **Layouts and Dynamic Routes:** Understand advanced routing concepts such as shared layouts for the dashboard shell and dynamic routes for individual property pages using URL parameters.
- **Fetching Data with useEffect :** Simulate fetching property and booking data from an API. Explain how to use the useEffect hook to perform side effects like data fetching. Use a hardcoded list of properties and bookings to simulate API responses.
- **Handling Loading and Error States:** Teach how to handle loading states while fetching data, displaying skeleton loaders or messages until property data is available. Implement error handling for failed API calls with user-friendly feedback.

Lesson 5

Global State Management, Form Management & Backend Integration

- **Global State Management:** Introduce the need for global state in larger applications. Discuss the limitations of prop drilling and why centralizing state can improve scalability, readability, and maintainability when multiple dashboard components need access to the same property and booking data.
- **Redux Toolkit:** Introduce the Redux Toolkit for managing state across multiple dashboard components. Set up slices for properties, room categories, and bookings with actions and reducers.
- **RTK Query:** Refactor the useEffect-based data fetching to RTK Query. Define API endpoints for fetching properties, room categories, and booking data from the NestBoard backend.
- **Form Management :** Demonstrate how to use react hook form and manage form inputs. Form management will be done when creating property information and when creating user profiles.
- **Connecting to Backend:** Demonstrate how to connect the dashboard frontend to the NestBoard backend using RESTful APIs. Implement functionality to fetch property and booking data from the live API, handle authentication headers, and manage user sessions.
- **Handling Errors and Validations:** Teach how to handle errors from API calls and validate user input on property and room category creation forms. Display error messages and feedback to vendors.

Foundations of Backend Web Application Development

In this phase you'll cover the backend development, building the API and database layer that powers NestBoard.



Project

NestBoard Backend API

You will create the backend structure for the NestBoard API and implement the endpoints required to support both the vendor dashboard and the tenant mobile app. You will get hands-on experience implementing CRUD logic for properties, room categories, and bookings, designing a relational database schema, and building production-ready service layers with authentication and authorization.

Lesson 6

Backend Foundations

- **Setting Up Express:** Express.js and learn the basics of setting up a server and creating a simple route that responds with a test message.
- **Basic Routing:** Learn how to define routes and handle HTTP methods (GET, POST, PATCH, DELETE). Implement routes for properties, room categories, and bookings.
- **Route Grouping:** Route grouping to organize routes under /api/properties, /api/rooms, and /api/bookings.
- **Middleware:** Learn middleware functions in Express.js and how they can be used for logging, authentication, and error handling. Create custom middleware for request logging.

Lesson 7

Middleware and Validation

- **Firestore JWT Validation:** Build middleware that validates Firestore JWT tokens on every incoming API request.
- **Request Validation:** Implement request validation using Zod to define strict schemas for property and room category creation payloads.

Lesson 8

Database Engineering I

- **Relational Data Design:** Understand the principles of normalization (1NF through 3NF) and ACID properties.
- **ERD Design:** Design the complete NestBoard entity-relationship diagram. Normalize the schema by separating Property from RoomCategory from RoomInstance.
- **Data Modeling Decisions:** Discuss why certain values like occupancy are derived through aggregation rather than stored as counters.

Lesson 9

Database Engineering II

- **Prisma ORM:** Write the Prisma schema for all NestBoard entities. Connect to PostgreSQL and run migrations.
- **Indexed Queries:** Implement indexed queries for property search by city and aggregation queries for monthly revenue and occupancy rates.
- **Seat Availability Logic:** Implement the booking overlap check to ensure no two confirmed bookings occupy the same seat for overlapping date ranges.
- **Authentication:** Implement Firebase Authentication on the backend. Validate JWT tokens server-side and resolve users from the token payload.

Lesson 10

Security and Access Control

- **Role-Based Access Control:** Implement role-based middleware so that Admin-only endpoints are protected. Store user roles in PostgreSQL and check them before allowing access.
- **CORS and Row-Level Security:** Configure CORS policies for web and mobile origins. Implement RLS policies so vendors can only query their own properties and bookings.

Lesson 11

Production Readiness

- **Service Layer Architecture:** Implement a dedicated service layer for booking logic that validates seat availability, calculates total amounts, and performs transactional writes.
- **File Uploads:** Build a cover image upload endpoint for property listings.
- **Environment Configuration:** Configure environment variables for Railway deployment, separating development and production configurations.

Integration and Deployment

In this phase, you will merge the independent systems into a cohesive product ecosystem. The web dashboard connects to the live backend API. The mobile app establishes its own server connection with secure session management. Both platforms are deployed to production environments and tested on real hardware.



Project

Production NestBoard Ecosystem

You will connect the React vendor dashboard and the React Native tenant app to the live NestBoard API. The web track implements payment handling with Stripe, data visualization for the admin dashboard, and deploys to Vercel and Railway. The mobile track establishes secure session persistence, hardens the booking submission flow, and generates builds for physical device testing.

Lesson 12

The API Bridge and Dashboard Visualization

- **Connecting Frontend to Backend:** Connect the React dashboard to the live NestBoard API using RESTful calls. Handle CORS preflight requests and manage global loading states during data hydration.
- **Data Visualization:** Integrate a charting library to build revenue charts and occupancy breakdowns for the admin dashboard. Display key metrics including total properties, active bookings, and occupancy percentages.

Lesson 13

Payment Handling

- **Payment Fundamentals:** Learn the fundamentals of secure payment handling and basic checkout flows.
- **Stripe SDK:** Integrate the Stripe SDK to implement the booking payment flow by applying the learned concepts.

Lesson 14

Production Hosting and Pipelines

- **Deploying the Frontend:** Deploy the React vendor dashboard to Vercel with automatic deployments on Git push.
- **Deploying the Backend:** Deploy the Express API to Railway with environment-based configuration.
- **CI/CD and Migrations:** Set up the Prisma migration pipeline as part of the deployment process. Configure CI/CD so that pushing to the main branch triggers automated builds and deployments.

Main Track - Mobile

Mobile Application Development

Foundation to React Native

This phase introduces React Native development, building the tenant-facing mobile application. Here the same backend API will be used as in Web, giving you the experience of building a unified product ecosystem across two client platforms.



Project (Mobile)

NestBoard Tenant Mobile App

You will build the tenant-facing mobile application using React Native. The app will allow users to search for co-living spaces, browse room occupants, navigate to properties on a map, scan QR codes for quick property lookup, and manage their bookings with offline-resilient caching.

Lesson 1

Mobile Ecosystem Foundations

- **The Mobile Paradigm:** Understand the differences between App Store binaries and web browsers. Explore the React Native architecture including the Bridge and the New Architecture with JSI and Fabric.
- **Project Setup:** Initialize a React Native project, walk through the file structure, and set up Android and iOS emulators.
- **React Concepts Review:** Briefly review React concepts as they apply to the mobile environment.
- **Native UI Components:** Map web tags to native components including View, Text, Image, Button, and ScrollView. Build a PropertySearchCard component with theming and styling.

Lesson 2

UI Components, Lists, and Theming

- **FlatList and Optimization:** Implement FlatList with data, key extractor, and render item callbacks. Apply lazy loading and optimization techniques for smooth scrolling.
- **Skeleton Placeholders:** Implement skeleton placeholder loading states for property listings while data is being fetched.

Lesson 3

Navigation and Deep Linking

- **React Navigation:** Integrate React Navigation and set up Stack and Tab navigation patterns.
- **Screen Flow:** Implement Tab navigation for Search, My Bookings, and Profile. Set up Stack navigation for the Property to Room to Booking flow. Handle back navigation and screen-to-screen data passing.
- **Deep Linking:** Configure deep linking so that a shared property URL opens the app directly to that property's detail page.
- **Axios Setup:** Set up the Axios library and configure a base instance for making API calls to the NestBoard backend.

Lesson 4

Data Fetching and State Management

- **Populating Data:** Fetch and display property data including text and images in the mobile app. Handle loading and error states during API calls.
- **Local Data Persistence:** Introduce AsyncStorage for basic local data persistence. Store and retrieve data to support offline access patterns.
- **Sign Up and Login:** Implement user registration and login screens. Connect to the backend authentication endpoints.

Lesson 5

Authentication

- **JWT Token Management:** Manage access and refresh tokens using Axios interceptors. Implement auto-redirect on app launch based on token validity.
- **Environment Configuration:** Manage sensitive data such as API keys and base URLs using environment files. Store authentication tokens in device secure storage.

Lesson 6

Search, Listing, and Detail View

- **Search Feature:** Build the property search filter with location and date inputs. Populate the search results listing with data from the API.
- **Property Detail View:** Navigate to the property details page and fetch property data including room categories and available seats.
- **QR Code Scanning:** Implement a QR code scanner for quick property lookup. Connect the scanner to navigation so scanning directs to the property detail page.
- **Map Integration:** Integrate Google Maps using the Google Cloud Platform. Render property locations on a native map component with pins at GPS coordinates.

Lesson 7

Map Integration and Geolocation

- **Directions:** Allow tenants to tap a property pin to open device navigation for directions.
- **Geolocation:** Use device geolocation to calculate distance from the tenant's current location to search results for distance-based property discovery.

Lesson 8

Offline-First and Local Persistence

- **Local Data Caching:** Implement AsyncStorage to cache property listings and search results locally. Display cached results with a "Last updated" indicator when offline.
- **Connection-Aware UI:** Detect network status and display a "You are offline" banner when connectivity is lost. Disable booking submissions when offline while allowing browsing of cached listings.
- **Data Synchronization:** Sync cached data when network status returns to online. Populate data from local storage when the app launches without connectivity.

Integration and Deployment

In this phase, you will merge the independent systems into a cohesive product ecosystem. The web dashboard connects to the live backend API. The mobile app establishes its own server connection with secure session management. Both platforms are deployed to production environments and tested on real hardware.



Project

Production NestBoard Ecosystem

You will connect the React vendor dashboard and the React Native tenant app to the live NestBoard API. The web track implements payment handling with Stripe, data visualization for the admin dashboard, and deploys to Vercel and Railway. The mobile track establishes secure session persistence, hardens the booking submission flow, and generates builds for physical device testing.

Lesson 9

The Native Handshake

- **Server Connection:** Connect the mobile app to the live NestBoard API. Set up Axios with interceptors for attaching Firebase JWT tokens to every request.
- **Secure Session Persistence:** Store Firebase tokens in device secure storage (Keychain/Keystore). Implement auto-redirect on app launch based on token validity.
- **Environment Management:** Configure environment variables to manage API base URLs and sensitive data across development and production.

Lesson 10

Reliable Submissions

- **Transactional Booking Flow:** Harden the booking submission to ensure seat reservation, payment simulation, and booking confirmation are handled atomically.
- **Error Boundaries:** Implement error boundary components to catch rendering failures gracefully. Build a centralized API error handler with user-friendly error messages.
- **Connection Resilience:** Ensure bookings are never queued offline. Display a clear failure screen on network drop and allow the tenant to retry manually.

Lesson 11

Sideloaded and Hardware Verification

- **Android Build Process:** Generate APK and AAB builds using Gradle. Understand the SHA-1 keystore signing process. Sideload the APK to a physical Android device.
- **iOS Build Process:** Explore the iOS build process and TestFlight distribution for development builds.
- **Hardware Verification:** Test the full tenant flow on real hardware: search, browse rooms, complete a booking. Verify that all hardware integrations function correctly on physical devices.

Main Track - Common

Web + Mobile



Orchestration and Quality

In this phase, you will learn to leverage AI tools for engineering productivity and establish comprehensive testing across the entire NestBoard ecosystem. These lessons apply to both the web and mobile tracks.

Lesson 1

AI Orchestration

- **AI-Powered Development:** Use Claude Code, GitHub Copilot, and Cursor for accelerating development workflows. Practice using AI agents to refactor code and generate documentation.
- **Practical Application:** Apply AI tooling to refactor the booking service layer. Generate CLAUDE.md documentation for the NestBoard codebase and expand features such as additional property search filters.

Lesson 2

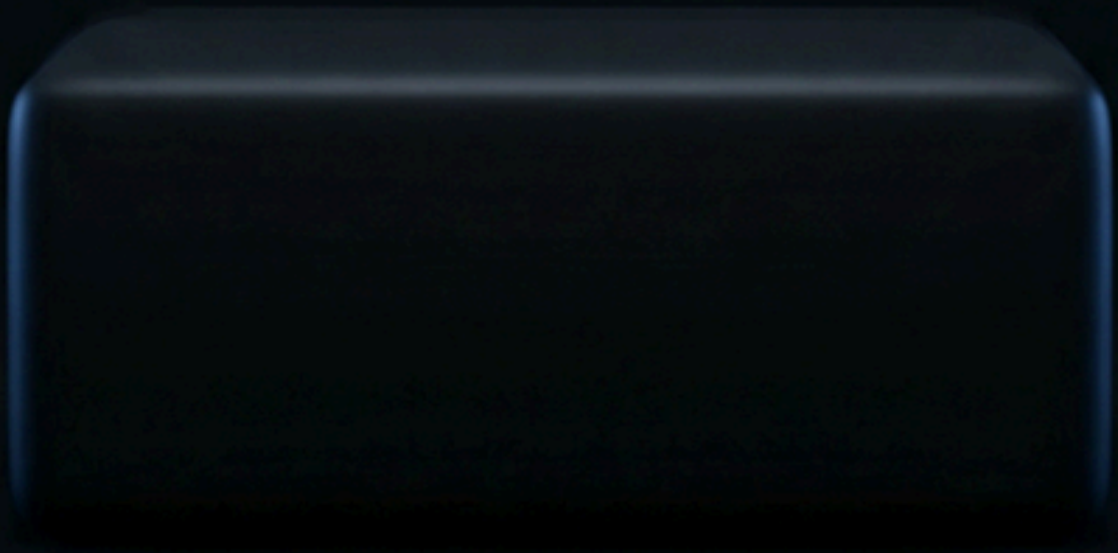
Quality Engineering

- **Backend Testing:** Establish test coverage using Jest and Supertest. Write API route tests covering booking creation, seat conflict prevention, and revenue aggregation accuracy.
- **Web End-to-End Testing:** Implement Playwright E2E tests for the web dashboard covering login, property creation, room category creation, and dashboard data verification.
- **Mobile End-to-End Testing:** Implement Maestro E2E flows for the mobile app covering login, property search, room browsing, and booking completion.

Sub Tracks

+

Bonus Sessions



Session 1: Micro-frontend Architecture

Explore advanced frontend architecture by splitting the NestBoard vendor dashboard into independently deployable modules. Implement Shell and Remote paradigms using Module Federation, allowing the Property Management module and the Booking Analytics module to be developed and deployed independently. Understand how this pattern enables engineering teams to scale without monolithic coupling.

Session 2: Product Innovation

Transition from engineering to product thinking. Led by an entrepreneur-in-residence, apply MoSCoW prioritization to the NestBoard feature backlog. Calculate the ROI of adding features such as tenant reviews, maintenance ticketing, or real payment integration. Present a product roadmap that balances technical feasibility with business value.

Career Coaching and Job Readiness

This final phase helps you transition from a student into a job-ready professional with skillset relevant to get placed in the industry. We'll guide you through how to present your technical expertise and project work with confidence, whether in your CV, portfolio, or interviews. You'll also learn targeted strategies to navigate the rapidly growing job market across industries, from startups to conglomerate & enterprises.

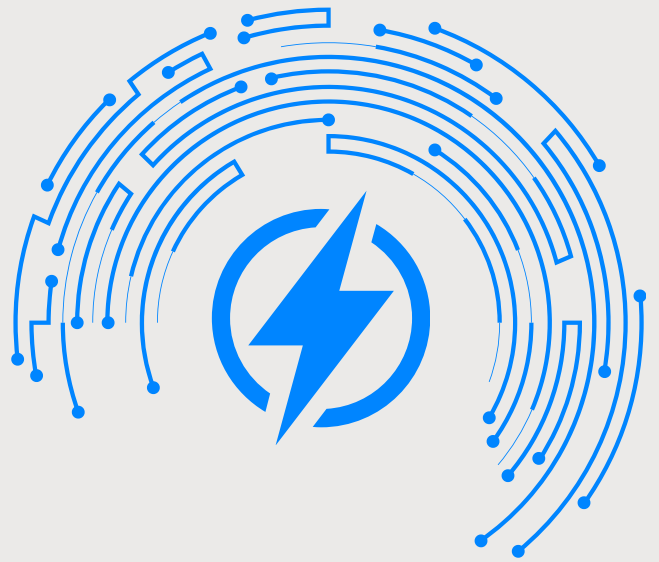
Professional Portfolio and CV Audit

You will work on crafting a standout CV and a structured portfolio that tells a compelling story of your technical growth. Under the guidance of our career coach, you will submit your materials for individual review, ensuring your profile highlights your experience and aligns perfectly with modern industry expectations.

What you'll do:

- Use curated CV templates to build a professional tech resume.
- Highlight specialized experience in development and engineering
- Submit your CV for individual review and feedback from industry experts.
- Learn how to structure your portfolio storytelling for maximum impact.
- Prepare for both behavioral and technical interviews with real question frameworks.
- Explore application strategies across product innovation companies and startups

STEMLink Learning Experience



Hands-on Projects

Experiential projects are designed to reflect actual workplace challenges. Learners can return to lessons at any time during the course to refresh concepts.



Live mentorship from young Entrepreneurs

Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.



Live -program Delivery

All the sessions are conducted in live and we're providing recordings for all of them for additional references.



Professional Certification

Providing a professional certificate to all the students who're complete the program



Learn from experienced engineers

Connect individually with our engineering team and discover how to solve the challenges that you encounter & tracking your progress.



Industry focused programs

Our course aligns with industry trends, providing real-world projects to ensure graduates are job-ready and competitive in the tech field.

