

Vancouver Events Hub - 3-Day Development Roadmap

Project Setup (Do This First - 30 minutes)

1. Install Required Tools

```
bash

# Check if Node.js is installed
node --version
npm --version

# If not installed, download from nodejs.org
```

2. Create Your Project

```
bash

# Navigate to your desired folder
cd Desktop # or wherever you want your project

# Create React app
npx create-react-app vancouver-events-hub
cd vancouver-events-hub

# Start development server
npm start
```

3. Get API Keys (Sign up for these NOW)

- Eventbrite API: <https://www.eventbrite.com/platform/api/>
- Meetup API: <https://www.meetup.com/api/>
- Create accounts and get your API keys

DAY 1: Foundation & Basic Event Display

Morning (3-4 hours)

Goal: Learn React basics and display some events

Step 1: Learn React Fundamentals (1 hour)

- Watch: "React in 30 minutes" on YouTube
- Key concepts: Components, JSX, State, Props

Step 2: Clean Up & Set Up Structure (30 minutes)

```
bash

# In your project folder
rm src/App.css src/logo.svg src/reportWebVitals.js src/setupTests.js

# Install additional packages
npm install axios
```

Step 3: Create Basic Components (1 hour)

Create these files in `src/`:

- `EventCard.js` - Display individual event
- `EventList.js` - Display list of events
- `SearchBar.js` - Search functionality

Step 4: Mock Data Display (1 hour)

Start with fake event data to get the UI working

Afternoon (2-3 hours)

Goal: Integrate real API data

Step 5: Eventbrite API Integration

- Set up API calls with axios
- Display real Vancouver events
- Handle loading states

Evening Goal

- Working React app that shows real events from Eventbrite
- Basic event cards with title, date, location
- Understanding of React components and state



DAY 2: Features & Styling

Morning (3-4 hours)

Goal: Add search, filtering, and better UI

Step 6: Search Functionality

- Add search by event name
- Filter by date range
- Filter by category

Step 7: Responsive Design

- Make it look good on mobile
- Use CSS Grid/Flexbox
- Clean, modern styling

Afternoon (3-4 hours)

Goal: Add more APIs and polish

Step 8: Meetup API Integration

- Add second data source
- Combine events from multiple APIs
- Remove duplicates

Step 9: Location Features

- Filter by distance from UBC
- Add map integration (optional)
- Sort by proximity

Evening Goal

- Beautiful, responsive design
 - Multiple data sources
 - Search and filtering working
 - Mobile-friendly interface
-

DAY 3: Backend API & Deployment

Morning (3-4 hours)

Goal: Create backend API

Step 10: Node.js Backend Setup

```
bash

# Create backend folder
mkdir backend
cd backend
npm init -y
npm install express cors axios dotenv
```

Step 11: API Routes

- Create Express server
- API endpoints for events
- Handle CORS for frontend
- Environment variables for API keys

Afternoon (3-4 hours)

Goal: Deploy and polish

Step 12: Connect Frontend to Backend

- Update React app to use your API
- Handle errors gracefully
- Add loading spinners

Step 13: Deployment

- **Frontend:** Deploy to Netlify
- **Backend:** Deploy to Railway or Heroku
- Test live version

Evening Goal

- Live, working application
- Full-stack architecture

Professional deployment

Ready for resume/demos

Key Learning Resources

React Basics

- [Free Code Camp React Course](#) (first 2 hours)
- [React Official Tutorial](#): reactjs.org/tutorial

APIs & JavaScript

- [Axios Documentation](#): axios-http.com
- [JavaScript Promises](#): MDN Web Docs

Styling

- [CSS Flexbox](#): css-tricks.com/snippets/css/a-guide-to-flexbox/
- [CSS Grid](#): css-tricks.com/snippets/css/complete-guide-grid/

Deployment

- [Netlify](#): netlify.com (drag & drop deployment)
 - [Railway](#): railway.app (backend deployment)
-

Success Metrics

By end of Day 3, you should have:

- Live demo URL to show recruiters
 - GitHub repository with clean code
 - Understanding of React, APIs, Node.js
 - Impressive project for resume
 - Real full-stack experience
-

Daily Check-ins

End of each day, ask yourself:

1. Can I explain what I built today?

2. What was the hardest part?
 3. What would I do differently?
 4. What should I focus on tomorrow?
-

Pro Tips

- **Start simple:** Get something working before making it pretty
 - **Commit often:** Push to GitHub multiple times per day
 - **Document as you go:** Add comments and README
 - **Test on different devices:** Make sure it works on mobile
 - **Ask for help:** Use ChatGPT/Claude when stuck
-

Ready to start? Begin with Project Setup above! 