**Tech Stack:**

React Native - code for creating app

Nodejs -for backend

PostgreSQL - database

Cloudinary- cloud service to store images and videos

Railway- API hosting platform

AWS- database storage

**Plan to learn and build projects:**

Week 1: Introduction to Basics

Objectives:

- Set up your development environment.

- Learn the basics of each technology.

Activities:

- Day 1-2: Set up accounts and install tools (Node.js, React Native CLI, PostgreSQL, AWS, Cloudinary, Railway).

- Day 3-4: Learn the basics of React Native:

- Follow the [React Native Getting Started Guide](https://reactnative.dev/docs/getting-started).

- Build a simple "Hello World" app.

- Day 5-6: Learn the basics of Node.js:

- Follow the [Node.js Getting Started Guide](https://nodejs.dev/learn).

- Build a simple API that returns a message.

- Day 7: Learn the basics of PostgreSQL:

- Follow a basic tutorial on PostgreSQL ([Tutorialspoint PostgreSQL Tutorial](https://www.tutorialspoint.com/postgresql/index.htm)).

- Set up a local PostgreSQL database and create a simple table.

Week 2: Deeper Dive into Frontend and Backend

Objectives:

- Build a simple React Native app.

- Create a backend with Node.js and connect it to your React Native app.

Activities:

- Day 1-3: Develop a basic React Native app (e.g., a to-do list).

- Use state management.

- Style the app using basic components.

- Day 4-5: Expand your Node.js API to handle CRUD operations.

- Connect the API to your React Native app.

- Day 6-7: Integrate PostgreSQL with your Node.js API.

- Store and retrieve data from PostgreSQL.

Week 3: Cloudinary and Railway

Objectives:

- Learn to store and retrieve images with Cloudinary.

- Deploy your backend using Railway.

Activities:

- Day 1-2: Set up Cloudinary and integrate it with your Node.js backend.

- Follow [Cloudinary Documentation](https://cloudinary.com/documentation).

- Implement image upload and retrieval functionality.

- Day 3-4: Deploy your Node.js API on Railway.

- Follow [Railway Documentation](https://docs.railway.app/).

- Day 5-7: Update your React Native app to support image uploads.

- Connect to your deployed backend.

Week 4: AWS and Deployment

Objectives:

- Learn the basics of AWS.

- Deploy your database to AWS.

Activities:

- Day 1-2: Learn the basics of AWS services ([AWS Training](https://aws.amazon.com/training/)).

- Focus on AWS RDS for PostgreSQL.

- Day 3-5: Set up an RDS instance and migrate your local PostgreSQL database to AWS.

- Connect your Node.js API to the AWS RDS database.

- Day 6-7: Update your backend and deploy any changes using Railway.

Week 5: Advanced Features and Optimization

Objectives:

- Add advanced features to your app.

- Optimize your code and database.

Activities:

- Day 1-3: Implement user authentication (e.g., JWT).

- Secure your API endpoints.

- Day 4-5: Add pagination and search functionality to your app.

- Optimize database queries.

- Day 6-7: Conduct performance testing and optimize the app.

Week 6: Full Project Development

Objectives:

- Start a full project using all technologies learned.

Activities:

- Day 1-3: Plan your project (e.g., a photo-sharing app).

- Define requirements and architecture.

- Day 4-7: Start development:

- Set up the frontend with React Native.

- Develop the backend with Node.js and connect to PostgreSQL.

Week 7: Project Development Continued

Objectives:

- Continue building your project.

- Implement additional features.

Activities:

- Day 1-3: Integrate Cloudinary for media storage.

- Ensure media uploads and retrieval work seamlessly.

- Day 4-5: Add more features based on your project plan.

- E.g., user profiles, comments, likes.

- Day 6-7: Deploy updates to Railway and AWS.

Week 8: Testing, Debugging, and Final Touches

Objectives:

- Test and debug your project.

- Deploy the final version.

Activities:

- Day 1-3: Conduct thorough testing (unit, integration, and end-to-end tests).

- Fix any bugs and optimize performance.

- Day 4-5: Prepare documentation for your project.

- Write a README, set up API documentation, etc.

- Day 6-7: Final deployment.

- Ensure everything is running smoothly on Railway and AWS.

- Share your project on GitHub and showcase it in your portfolio.