**Web-Dev 2.0 LinkedIn Calendar (120 Days Challenge)**

**120-Day Full-Stack Web Development Challenge + LinkedIn Posting Calendar**

**Instructions for Use**

* Each day, follow the “Task Focus” to guide your learning and coding.
* After completing the day’s task, write a **LinkedIn post** using the suggested topic and structure under “LinkedIn Post Idea.”
* Feel free to adjust or expand any day’s plan based on your pacing or specific interests.
* Keep track of your mini-projects, GitHub repos, and deployments to showcase them in your portfolio.

**Month 1: HTML, CSS, Git, and Core JavaScript (Days 1–30)**

**Week 1: HTML Essentials + Introduction to Git/GitHub**

**Day 1**

**Task Focus**

* Understand basic HTML structure (<!DOCTYPE html>, <html>, <head>, <body>).
* Learn semantic tags (<header>, <section>, <footer>, etc.).
* Start your GitHub repository for this challenge.

**LinkedIn Post Idea**

* **Title**: “Day 1 of #100DaysOfCode (Plus 20 Bonus!) — Kicking Off My Full-Stack Journey”
* **Content**:
  + Introduce your 120-day challenge and why you’re committing to it.
  + Mention that you learned about HTML’s skeleton and semantic tags today.
  + Add a code snippet of a basic HTML page.
  + End with a question: “Any advice for staying motivated during a long coding challenge?”

**Day 2**

**Task Focus**

* Dive deeper into HTML forms (<form>, <input>, <select>, <textarea>).
* Embed images, audio, or video.
* Practice pushing changes to your GitHub repo.

**LinkedIn Post Idea**

* **Title**: “Day 2: Exploring HTML Forms and Media”
* **Content**:
  + Show a screenshot or gist of a simple contact form you created.
  + Discuss one challenge you faced (e.g., aligning form labels) and how you solved it.
  + Mention your GitHub repo link for anyone who wants to follow along.

**Day 3**

**Task Focus**

* Install Git (if not already).
* Learn basic Git commands: git init, git add, git commit, git push.
* Create/push a test repository on GitHub.

**LinkedIn Post Idea**

* **Title**: “Day 3: Version Control Adventures with Git & GitHub”
* **Content**:
  + Explain why version control is essential for developers.
  + Include a short code snippet or screenshot showing your commit history.
  + Ask your network how they handle branching strategies.

**Day 4**

**Task Focus**

* Master Git branching (git checkout -b, git merge).
* Learn how to resolve merge conflicts.
* Experiment with a feature branch workflow.

**LinkedIn Post Idea**

* **Title**: “Day 4: Branching Out in Git”
* **Content**:
  + Demonstrate a merge conflict you intentionally created and solved.
  + Share a helpful resource on Git branching (e.g., Atlassian tutorials).
  + Encourage others to comment with their top Git tips.

**Day 5**

**Task Focus**

* Deeper look into **semantic HTML**: <figure>, <figcaption>, <aside>.
* Add basic SEO meta tags (<meta name="description" ...>).
* Continue refining your HTML pages.

**LinkedIn Post Idea**

* **Title**: “Day 5: Going Semantic with HTML”
* **Content**:
  + Show how semantic tags improve accessibility and SEO.
  + Include a small before-and-after snippet comparing <div> soup vs. semantic tags.
  + Encourage readers to check their own code for improved semantics.

**Day 6**

**Task Focus**

* **Mini Project #1**: Build a multi-page static site (Home, About, Contact).
* Use Git for version control; finalize or host on GitHub Pages.

**LinkedIn Post Idea**

* **Title**: “Day 6: My First Multi-Page Website!”
* **Content**:
  + Provide a live demo link (GitHub Pages or Netlify).
  + Mention one design or structure choice you’re proud of.
  + Ask for feedback on UI or layout improvements.

**Day 7**

**Task Focus**

* **Review & Reinforcement**: Revisit everything from Days 1–6.
* Fix any code issues, reorganize repositories, or catch up if behind.

**LinkedIn Post Idea**

* **Title**: “Day 7: Week 1 Recap — HTML & Git Mastery”
* **Content**:
  + Reflect on how you tackled HTML and Git basics.
  + Share a quick summary of your biggest lesson learned.
  + Thank anyone who provided advice or feedback.

**Week 2: CSS – Styling the Web**

**Day 8**

**Task Focus**

* CSS Basics: Linking external stylesheets, selectors, specificity.
* Box Model: margin, border, padding.

**LinkedIn Post Idea**

* **Title**: “Day 8: Stepping into CSS”
* **Content**:
  + Show a quick snippet demonstrating the box model.
  + Explain how you resolved any layout confusion.
  + Ask for best practices in naming CSS classes.

**Day 9**

**Task Focus**

* **CSS Positioning** (static, relative, absolute, fixed, sticky).
* display types (block, inline, inline-block, flex, grid).

**LinkedIn Post Idea**

* **Title**: “Day 9: Wrangling CSS Position & Display”
* **Content**:
  + Before/after screenshots of a div that changes position from static to absolute.
  + Mention how understanding positioning helps create neat UIs.

**Day 10**

**Task Focus**

* **Flexbox**: display: flex;, justify-content, align-items, flex-direction.
* Build a small responsive navbar or card layout with Flexbox.

**LinkedIn Post Idea**

* **Title**: “Day 10: Flexbox Deep Dive + Mini Layout Demo”
* **Content**:
  + Show a short video/gif or screenshot of the responsive navbar you built.
  + Talk about one “gotcha” moment with Flexbox and how you overcame it.
  + Invite others to share their favorite Flexbox resources.

**Day 11**

**Task Focus**

* **CSS Grid**: Basic properties (grid-template-columns, grid-template-rows, gap).
* Compare Grid vs. Flexbox usage scenarios.

**LinkedIn Post Idea**

* **Title**: “Day 11: Grid vs. Flexbox Showdown”
* **Content**:
  + Post a side-by-side comparison of a layout in Flexbox vs. Grid.
  + Explain which approach you find more intuitive.
  + Ask your network which they prefer and why.

**Day 12**

**Task Focus**

* **Responsive Design** & Media Queries (@media rules).
* Use relative units (%, em, rem, vw, vh).

**LinkedIn Post Idea**

* **Title**: “Day 12: Making Websites Responsive”
* **Content**:
  + Show a quick phone vs. desktop screenshot of your layout.
  + Mention any device testing strategy you used (e.g., Chrome DevTools).

**Day 13**

**Task Focus**

* **CSS Best Practices**: BEM methodology for class naming.
* Basic Animations with transition, transform, @keyframes.

**LinkedIn Post Idea**

* **Title**: “Day 13: Cleaner CSS + Subtle Animations”
* **Content**:
  + Show a short clip or code snippet of a hover animation.
  + Discuss how BEM naming improved your code readability.

**Day 14**

**Task Focus**

* **Mini Project #2**: A responsive portfolio page.
* One-page design with sections for “Home,” “About,” “Projects,” “Contact.”
* Deploy on Netlify or GitHub Pages.

**LinkedIn Post Idea**

* **Title**: “Day 14: Launching My Responsive Portfolio”
* **Content**:
  + Provide the live link; highlight your sections.
  + Ask for design or UX feedback.
  + Encourage viewers to star your GitHub repo if they find it helpful.

**Week 3: JavaScript Fundamentals**

**Day 15**

**Task Focus**

* **JavaScript Basics**: Variables (let, const, var), data types, arithmetic operators.
* Simple console logging to see outputs.

**LinkedIn Post Idea**

* **Title**: “Day 15: JavaScript Under the Hood”
* **Content**:
  + Include a small snippet demonstrating variable declarations with let and const.
  + Share an interesting fact about JS data types.

**Day 16**

**Task Focus**

* **Control Flow**: conditionals (if/else, switch), loops (for, while, do...while).
* Write a small script to handle user input in the console.

**LinkedIn Post Idea**

* **Title**: “Day 16: Making Decisions with JS”
* **Content**:
  + Show a quick code snippet for a loop-based challenge you solved.
  + Mention a real-world scenario where control flow is handy (e.g., form validations).

**Day 17**

**Task Focus**

* **Functions & Scope**: function declarations, expressions, arrow functions, global vs. local scope.
* Create a utility library with a few simple helper functions.

**LinkedIn Post Idea**

* **Title**: “Day 17: Functional Fun in JavaScript”
* **Content**:
  + Post a snippet of your favorite helper function (e.g., random number generator).
  + Explain the difference between function declarations vs. arrow functions.

**Day 18**

**Task Focus**

* **Arrays & Objects**: push, pop, map, filter, reduce.
* Create an address book array of objects; let users add/search.

**LinkedIn Post Idea**

* **Title**: “Day 18: Mastering Arrays & Objects in JS”
* **Content**:
  + Share a short snippet of using .reduce() or .filter() creatively.
  + Talk about how data structures shape real-world applications (e.g., contact lists).

**Day 19**

**Task Focus**

* **DOM Manipulation**: selecting elements, modifying styles/content, adding event listeners.
* Build a small to-do list or counter app.

**LinkedIn Post Idea**

* **Title**: “Day 19: Bringing Webpages to Life with the DOM”
* **Content**:
  + Include a screenshot of your interactive to-do list.
  + Explain a tricky DOM event handling scenario you solved.

**Day 20**

**Task Focus**

* **JSON and Local Storage**: JSON.stringify(), JSON.parse().
* Persist your to-do list or a small user setting in localStorage.

**LinkedIn Post Idea**

* **Title**: “Day 20: Saving Data in the Browser”
* **Content**:
  + Show how you used localStorage to persist data.
  + Reflect on the importance of storing user preferences for better UX.

**Day 21**

**Task Focus**

* **Mini Project #3**: Interactive Web App (e.g., Quiz App or Recipe Manager).
* Use DOM manipulation and localStorage; share the final code on GitHub.

**LinkedIn Post Idea**

* **Title**: “Day 21: Launching My Interactive Web App”
* **Content**:
  + Provide a demo link or a GitHub repo.
  + Mention your biggest JavaScript “aha” moment this week.
  + Ask for feedback on your code quality.

**Week 4: Advanced JS Concepts & Consolidation**

**Day 22**

**Task Focus**

* **ES6+ Features**: Destructuring, spread operator, rest parameters, template literals.
* Refactor old code using these modern features.

**LinkedIn Post Idea**

* **Title**: “Day 22: Leveling Up with ES6”
* **Content**:
  + Show a snippet that uses destructuring or the spread operator.
  + Note how it makes your code cleaner vs. older JS syntax.

**Day 23**

**Task Focus**

* **Asynchronous JavaScript**: callbacks, Promises, async/await.
* Fetch data from a public API (fetch() or axios).

**LinkedIn Post Idea**

* **Title**: “Day 23: Going Async — Fetching My First API”
* **Content**:
  + Display the result of your API call (e.g., jokes, weather data).
  + Explain one concept about asynchronous flow that initially confused you.

**Day 24**

**Task Focus**

* **Error Handling & Debugging**: try...catch, using DevTools breakpoints.
* Practice debugging a broken script step by step.

**LinkedIn Post Idea**

* **Title**: “Day 24: Battling Bugs & Errors”
* **Content**:
  + Show a snippet of a try...catch block that handled an API error.
  + Mention a tip for new developers to speed up their debugging.

**Days 25–26**

**Task Focus**

* **Mini Project #4**: API-Driven Web App (Movie Search, Weather App, etc.)
* Integrate loading states, error states, and deploy the final project.

**LinkedIn Post Idea (Day 25)**

* **Title**: “Day 25: Building My API-Powered Web App — Part 1”
* **Content**:
  + Share your approach for structuring API calls and state management.
  + Ask your audience for suggestions on UI/UX improvements.

**LinkedIn Post Idea (Day 26)**

* **Title**: “Day 26: Wrapping Up My API App + Deployment”
* **Content**:
  + Include a link to the live app.
  + Discuss a real-world scenario where such an app can be useful.
  + Invite feedback and potential collaborators.

**Day 27**

**Task Focus**

* **Git/GitHub Deep Dive**: Pull requests, code reviews, issues, releases.
* Create a feature branch and open a PR for your existing project.

**LinkedIn Post Idea**

* **Title**: “Day 27: Collaborating Like a Pro on GitHub”
* **Content**:
  + Demonstrate your PR process with screenshots.
  + Discuss how code reviews improve code quality.

**Day 28**

**Task Focus**

* **Refactoring & Code Organization**: ES Modules or CommonJS, better file structure.
* Split large scripts into multiple modules.

**LinkedIn Post Idea**

* **Title**: “Day 28: Cleaner Code, Happy Developer”
* **Content**:
  + Show your new folder/file structure.
  + Share how modularization makes your life easier.

**Days 29–30**

**Task Focus**

* **Final Review of Front-End Foundations**: HTML, CSS, JS recap.
* Polish your portfolio projects; push final changes to GitHub.

**LinkedIn Post Idea (Day 29)**

* **Title**: “Day 29: Revisiting the Fundamentals”
* **Content**:
  + Quick bullet list of your biggest takeaways in HTML/CSS/JS.
  + Reflect on your improvement from Day 1 to Day 29.

**LinkedIn Post Idea (Day 30)**

* **Title**: “Day 30: Front-End Milestone Complete!”
* **Content**:
  + Showcase your portfolio so far.
  + Invite connections to view your GitHub and share feedback.

**Month 2: React.js & Advanced Front-End (Days 31–60)**

**Week 5: React Basics**

**Day 31**

**Task Focus**

* **Introduction to React**: Set up via Create React App or Vite.
* Understand JSX, component structure, and project folder setup.

**LinkedIn Post Idea**

* **Title**: “Day 31: Hello React World”
* **Content**:
  + Screenshot of your “Hello World” in a React component.
  + Brief explanation of why React is so popular.

**Day 32**

**Task Focus**

* **Components & Props**: build functional components, pass data as props.
* Create a small “Profile Card” component.

**LinkedIn Post Idea**

* **Title**: “Day 32: Mastering Props in React”
* **Content**:
  + Show your Profile Card in action.
  + Describe how props make components reusable.

**Day 33**

**Task Focus**

* **State & Lifecycle (Hooks)**: useState, useEffect.
* Add interactivity (e.g., a “Follow” button toggling follow state).

**LinkedIn Post Idea**

* **Title**: “Day 33: Diving into React Hooks”
* **Content**:
  + GIF or screenshot of your interactive “Follow” button.
  + Mention the difference between useState vs. older class-based states.

**Day 34**

**Task Focus**

* **Event Handling** in React (onClick, onChange, onSubmit).
* Create a simple form to update state on user input.

**LinkedIn Post Idea**

* **Title**: “Day 34: Events + Forms in React”
* **Content**:
  + Show your form and how it updates the component’s state in real-time.
  + Reflect on how it differs from vanilla JS event handling.

**Day 35**

**Task Focus**

* **Conditional Rendering & Lists**: .map(), handling keys.
* Display a list of items; conditionally render messages if list is empty.

**LinkedIn Post Idea**

* **Title**: “Day 35: Rendering Dynamically in React”
* **Content**:
  + Example snippet of a list rendered via .map().
  + Share a challenge (e.g., forgetting the key prop) and how you fixed it.

**Day 36**

**Task Focus**

* **React Router Basics**: set up multiple routes with <BrowserRouter>, <Routes>, <Route>.
* Convert a multi-page layout into a single-page app.

**LinkedIn Post Idea**

* **Title**: “Day 36: Adding Navigation with React Router”
* **Content**:
  + A short video/gif showing transitions between routes.
  + Summarize why SPAs are great for user experience.

**Day 37**

**Task Focus**

* **Mini Project #5**: Simple React App (e.g., Product Catalog, Task Manager) with at least two routes.
* Deploy on Netlify or Vercel.

**LinkedIn Post Idea**

* **Title**: “Day 37: My First React App Deployed!”
* **Content**:
  + Demo link + GitHub repo.
  + Show how props, state, and routing come together.

**Week 6: Deeper into React**

**Day 38**

**Task Focus**

* **Advanced Hooks**: useEffect for side effects, useContext for global state, useReducer for complex logic.
* Convert or refactor a smaller component to use useContext for shared data.

**LinkedIn Post Idea**

* **Title**: “Day 38: Going Beyond useState — Advanced React Hooks”
* **Content**:
  + Brief code snippet of useContext in action.
  + Show how it simplifies passing data across components.

**Day 39**

**Task Focus**

* **Forms & Form Libraries** (Formik or React Hook Form).
* Build a “Sign Up” form with validation and error handling.

**LinkedIn Post Idea**

* **Title**: “Day 39: Making Forms Easier with Formik (or React Hook Form)”
* **Content**:
  + Screenshot of your sign-up form with validation error messages.
  + Talk about how a library streamlined form handling.

**Day 40**

**Task Focus**

* **Styling in React**: CSS Modules, Styled Components, or Tailwind CSS.
* Apply consistent styling across your mini projects.

**LinkedIn Post Idea**

* **Title**: “Day 40: Styling My React App — Which Approach Reigns Supreme?”
* **Content**:
  + Compare two approaches you tried (e.g., CSS Modules vs. Tailwind).
  + Ask your network which styling library they prefer.

**Day 41**

**Task Focus**

* **State Management**: Redux or advanced Context/Reducer pattern.
* Implement global store for something (e.g., cart items in an e-commerce mock app).

**LinkedIn Post Idea**

* **Title**: “Day 41: Taming State with Redux (or Context + Reducer)”
* **Content**:
  + Short snippet of your reducer or Redux store.
  + Explain how it streamlined state across multiple components.

**Days 42–43**

**Task Focus**

* **Mini Project #6**: React CRUD App (e.g., Contact Manager, Blog Posts).
* Use React Router, advanced hooks, form validations, state management.

**LinkedIn Post Idea (Day 42)**

* **Title**: “Day 42: Building a CRUD App in React — Part 1”
* **Content**:
  + Show your initial setup with routing and data structure.
  + Mention one key challenge in setting up the project.

**LinkedIn Post Idea (Day 43)**

* **Title**: “Day 43: CRUD App Complete + Deploy”
* **Content**:
  + Share the final app link, highlight the CRUD operations.
  + Mention a snippet focusing on how you handle Create vs. Update logic.

**Day 44 on next page**

**Day 44**

**Task Focus**

* **Review & React Best Practices**: optimization with memoization or pure components, folder structure.
* Prepare to transition to Node.js.

**LinkedIn Post Idea**

* **Title**: “Day 44: Polishing My React Skillset”
* **Content**:
  + List your biggest takeaways from building React apps.
  + Reflect on how you improved performance (memoization, code splitting).

**Month 3: Node.js, Express, MongoDB (Days 45–90)**

**Week 7: Node.js Fundamentals**

**Day 45**

**Task Focus**

* **Node.js Intro**: architecture, V8 engine, installing Node, npm vs. yarn.
* Run a simple .js file in Node.

**LinkedIn Post Idea**

* **Title**: “Day 45: Stepping into the Back-End with Node.js”
* **Content**:
  + Show your basic Node script output in the terminal.
  + Mention a concept about how Node processes JavaScript differently from browsers.

**Day 46**

**Task Focus**

* **Node Modules & NPM**: importing/exporting modules, built-in modules (fs, path).
* Create a small utility module and import it.

**LinkedIn Post Idea**

* **Title**: “Day 46: Modularizing Node.js Code”
* **Content**:
  + Code snippet of a custom module you wrote.
  + Discuss how reusability helps in larger projects.

**Day 47**

**Task Focus**

* **Express.js Basics**: set up an Express server, handle basic routes (GET, POST).
* Use simple middleware functions.

**LinkedIn Post Idea**

* **Title**: “Day 47: Hello Express — My First Server”
* **Content**:
  + Show how you defined your routes.
  + Mention a scenario where Express’s simplicity shines.

**Day 48**

**Task Focus**

* **Templating Engines & Static Files**: EJS, Handlebars, or Pug.
* Serve static files (CSS, images) via Express.

**LinkedIn Post Idea**

* **Title**: “Day 48: Server-Side Rendering Basics”
* **Content**:
  + Show a screenshot of a page rendered by EJS or Handlebars.
  + Discuss how this differs from a React-based front-end.

**Days 49–50**

**Task Focus**

* **Express Router & MVC**: separate routes, controllers, and possibly views.
* Refactor your simple Express app into an MVC-ish structure.

**LinkedIn Post Idea (Day 49)**

* **Title**: “Day 49: Structuring My Express App — MVC Style”
* **Content**:
  + Show folder layout and how it clarifies responsibilities.
  + Encourage readers to share their folder structures for back-end apps.

**LinkedIn Post Idea (Day 50)**

* **Title**: “Day 50: Reflecting on Express App Architecture”
* **Content**:
  + Talk about the importance of scalable architecture.
  + Share a resource or tutorial link that helped you.

**Day 51**

**Task Focus**

* **Mini Project #7**: RESTful API (e.g., “Books” or “Movies”).
* Test endpoints with Postman or a similar tool. Deploy on Railway or Heroku if possible.

**LinkedIn Post Idea**

* **Title**: “Day 51: Building & Deploying My First REST API”
* **Content**:
  + Share a snippet or screenshot of a successful Postman request.
  + Provide the live endpoint if deployed.

**Week 8: MongoDB & Mongoose**

**Day 61**

**Task Focus**

* **MongoDB Introduction**: NoSQL concepts, installing MongoDB or using MongoDB Atlas.
* Insert test documents into a database.

**LinkedIn Post Idea**

* **Title**: “Day 61: Exploring the NoSQL World with MongoDB”
* **Content**:
  + Summarize the difference between SQL vs. NoSQL.
  + Show a screenshot of your first inserted document in MongoDB.

**Day 62**

**Task Focus**

* **Mongoose ODM**: Connect to MongoDB, define schemas and models.
* Create a “User” model with basic CRUD operations.

**LinkedIn Post Idea**

* **Title**: “Day 62: Schemas, Models & Magic with Mongoose”
* **Content**:
  + Show your user schema snippet.
  + Mention how it simplifies database interactions.

**Day 63**

**Task Focus**

* **Advanced Queries**: filtering, projections, pagination, sorting.
* Implement pagination in an Express route for your data.

**LinkedIn Post Idea**

* **Title**: “Day 63: Taking Control of Data — Mongoose Queries”
* **Content**:
  + Show code that handles pagination.
  + Reflect on how pagination improves performance for larger datasets.

**Day 64**

**Task Focus**

* **Data Validation & Error Handling**: Mongoose schema validation, Express error handling.
* Enforce data integrity in your models.

**LinkedIn Post Idea**

* **Title**: “Day 64: Locking Down Data Quality with Validation”
* **Content**:
  + Give an example of a custom validation you wrote.
  + Mention how you handle errors gracefully in your app’s routes.

**Day 65**

**Task Focus**

* **Authentication (JWT Basics)**: generating and verifying tokens, securing routes.
* Implement a signup/login route returning JWT.

**LinkedIn Post Idea**

* **Title**: “Day 65: Securing My API with JWTs”
* **Content**:
  + Diagram explaining how token-based auth works.
  + Share a snippet of your route that generates tokens.

**Days 66–67**

**Task Focus**

* **Mini Project #8**: Secure REST API with MongoDB.
* Users can only read/write their own data; implement error handling and validation.

**LinkedIn Post Idea (Day 66)**

* **Title**: “Day 66: Building an Auth-Enabled REST API — Part 1”
* **Content**:
  + Show the basic routes you’ve set up.
  + Mention how you structure authentication middleware.

**LinkedIn Post Idea (Day 67)**

* **Title**: “Day 67: My Auth-Enabled CRUD App Is Live!”
* **Content**:
  + Deploy link plus endpoints documentation.
  + Ask for testers or feedback on API design.

**Week 9: Integrating Front-End & Back-End**

**Day 68**

**Task Focus**

* **Connecting React to Express**: making API calls from the React front-end (fetch, axios).
* Handle CORS configurations.

**LinkedIn Post Idea**

* **Title**: “Day 68: Bridging Front-End and Back-End”
* **Content**:
  + Screenshot showing data fetched from your API in a React component.
  + Mention the CORS problem you faced and how you solved it.

**Day 69**

**Task Focus**

* **Client-Side Authentication**: storing JWT in localStorage/cookies, protecting routes in React.
* Possibly use React context or Redux for auth state.

**LinkedIn Post Idea**

* **Title**: “Day 69: Implementing JWT Auth on the Front-End”
* **Content**:
  + Show how you secure private routes in React.
  + Ask if others prefer cookies vs. localStorage for tokens.

**Day 70**

**Task Focus**

* **Handling File Uploads** with Multer (Express), file input in React forms.
* Provide upload functionality in your app (e.g., profile pictures).

**LinkedIn Post Idea**

* **Title**: “Day 70: File Uploads Made Easy — Node + React”
* **Content**:
  + Show a quick clip of a file getting uploaded and displayed.
  + Mention how you secure uploaded files or store them (cloud storage?).

**Days 71–72**

**Task Focus**

* **Mini Project #9**: Full-Stack Application (e.g., Task Manager, Dashboard) with user auth and CRUD.
* Deploy front-end (Netlify/Vercel) and back-end (Railway/Heroku).

**LinkedIn Post Idea (Day 71)**

* **Title**: “Day 71: Full-Stack Project Kickoff”
* **Content**:
  + Outline your project’s goals and features.
  + Show your initial front-end and back-end repos.

**LinkedIn Post Idea (Day 72)**

* **Title**: “Day 72: Full-Stack App Deployed!”
* **Content**:
  + Provide the live links for both front-end and back-end.
  + Encourage your network to test the app and report any bugs.

**Day 73**

**Task Focus**

* **GitHub Project Management**: issues, project boards, labeling tasks.
* Create a Kanban board for your mini-project tasks.

**LinkedIn Post Idea**

* **Title**: “Day 73: Streamlining Development with GitHub Projects”
* **Content**:
  + Screenshot of your Kanban board.
  + Explain how it helps you stay organized.

**Day 74**

**Task Focus**

* **Code Review & Optimization**: environment variables, reduce bundle size.
* Implement lazy loading or caching for your front-end or back-end as appropriate.

**LinkedIn Post Idea**

* **Title**: “Day 74: Code Cleanup & Performance Tuning”
* **Content**:
  + Show a performance improvement metric (e.g., smaller bundle size).
  + Invite your network to share optimization tips.

**Week 10: Advanced Back-End & Third-Party Integrations**

**Day 75**

**Task Focus**

* **Advanced Express Concepts**: rate limiting (express-rate-limit), security best practices (Helmet, sanitizing inputs).
* Strengthen your existing Express app.

**LinkedIn Post Idea**

* **Title**: “Day 75: Securing My Express App”
* **Content**:
  + Summarize how you implemented rate limiting.
  + Mention why security is non-negotiable in production apps.

**Day 76**

**Task Focus**

* **Third-Party APIs**: OAuth flows or payments (Stripe/PayPal).
* Implement a route that interacts with a popular third-party service.

**LinkedIn Post Idea**

* **Title**: “Day 76: Integrating Stripe (or Another API) into My App”
* **Content**:
  + Show a snippet of successful API response or a mock payment.
  + Reflect on the importance of integrating external services.

**Day 77**

**Task Focus**

* **Logging & Monitoring**: use Morgan for request logging, error logging, or Sentry for error tracking.
* Add logs to your app to track usage.

**LinkedIn Post Idea**

* **Title**: “Day 77: Keeping Tabs on My App with Logging Tools”
* **Content**:
  + Describe how you set up Morgan to log requests.
  + Share why real-time monitoring helps in production.

**Days 78–79**

**Task Focus**

* **Mini Project #10**: Extend a previous full-stack app with a third-party API (e.g., payment, social login).
* Ensure proper security and error handling.

**LinkedIn Post Idea (Day 78)**

* **Title**: “Day 78: Adding New Features with Third-Party Integrations”
* **Content**:
  + Outline the feature you’re integrating (e.g., Google Maps).
  + Mention potential challenges like API rate limits.

**LinkedIn Post Idea (Day 79)**

* **Title**: “Day 79: Third-Party Integration Complete!”
* **Content**:
  + Showcase a demo with the new feature in action.
  + Talk about how you handled potential security concerns.

**Day 80**

**Task Focus**

* **Testing Basics**: unit testing with Jest or Mocha/Chai, integration tests for routes.
* Write tests for at least one model and route in your API.

**LinkedIn Post Idea**

* **Title**: “Day 80: Testing the Waters with Jest”
* **Content**:
  + Show a test file that checks a simple route or function.
  + Encourage others to adopt TDD or at least some testing in their workflow.

**Day 81**

**Task Focus**

* **Review & Prep for Advanced Topics**: re-check full-stack mini-projects, note improvements.
* Plan how to finalize them for your portfolio.

**LinkedIn Post Idea**

* **Title**: “Day 81: Taking Stock Before the Final Stretch”
* **Content**:
  + Summarize the major back-end concepts you’ve covered.
  + Tease your final Capstone Project idea.

**Month 4: Final Polish, Advanced Topics & Capstone (Days 82–120)**

**Week 11: Advanced Deployment & Production Readiness**

**Day 82**

**Task Focus**

* **Deployment Options**: comparing AWS, DigitalOcean, Render, Vercel.
* (Optional) Briefly explore Docker for containerization.

**LinkedIn Post Idea**

* **Title**: “Day 82: Exploring My Deployment Toolbox”
* **Content**:
  + Talk about pros/cons of each hosting platform.
  + Ask your network about their favorite hosting service.

**Day 83**

**Task Focus**

* **CI/CD Pipelines**: GitHub Actions or GitLab CI, automating tests and deployments.
* Set up a simple pipeline for one project.

**LinkedIn Post Idea**

* **Title**: “Day 83: Automating Deployments with CI/CD”
* **Content**:
  + Show your .yml config for GitHub Actions or a pipeline screenshot.
  + Explain how CI/CD fosters continuous improvement.

**Days 84–85**

**Task Focus**

* **Performance Optimization**: caching strategies, code splitting in React, analyzing memory usage in Node.
* Implement lazy-loading for a large component or caching an API route.

**LinkedIn Post Idea (Day 84)**

* **Title**: “Day 84: Performance Gains — Code Splitting in React”
* **Content**:
  + Show how you used React’s lazy/suspense to load components on demand.
  + Compare before/after performance metrics.

**LinkedIn Post Idea (Day 85)**

* **Title**: “Day 85: Caching My Node API for Speed”
* **Content**:
  + Snippet of your caching mechanism (e.g., Redis or in-memory).
  + Mention how it improved API response times.

**Days 86–87**

**Task Focus**

* **Final Mini Project #11**: Make one of your previous mini-projects fully production-ready.
* Add test coverage, CI/CD, environment variables, security, monitoring.

**LinkedIn Post Idea (Day 86)**

* **Title**: “Day 86: Upgrading a Mini-Project for Production — Part 1”
* **Content**:
  + Outline the steps you’re taking (tests, env vars, etc.).
  + Ask your network to share essential steps for production readiness.

**LinkedIn Post Idea (Day 87)**

* **Title**: “Day 87: Production-Ready Project Launch!”
* **Content**:
  + Final link to your project + highlights of security/performance improvements.
  + Encourage peers to check out the code or open a PR.

**Day 88**

**Task Focus**

* **Portfolio & Resume Review**: update your portfolio site, pinned GitHub repos, LinkedIn summary.
* Ensure everything is well-documented and displayed.

**LinkedIn Post Idea**

* **Title**: “Day 88: Polishing My Portfolio & Resume”
* **Content**:
  + Show a screenshot of your updated portfolio home page.
  + Ask for feedback on any design or content aspects.

**Weeks 12–13 (Days 89–102): Final Capstone — Planning & Kickoff**

**Day 89**

**Task Focus**

* **Capstone Project Planning**: choose an idea (e-commerce, social media, job board, etc.).
* Draft user stories, features, and a GitHub project board.

**LinkedIn Post Idea**

* **Title**: “Day 89: The Grand Finale — Capstone Project Plan”
* **Content**:
  + Describe your project concept and target users.
  + Invite suggestions from your network on features.

**Days 90–91**

**Task Focus**

* **Designing the Architecture**: plan DB schema (MongoDB), outline API endpoints, React front-end.
* Finalize project scope based on timeline.

**LinkedIn Post Idea (Day 90)**

* **Title**: “Day 90: Scheming My Capstone’s Database & APIs”
* **Content**:
  + Show an ERD or schema diagram.
  + Mention your approach to route design.

**LinkedIn Post Idea (Day 91)**

* **Title**: “Day 91: Front-End Architecture for My Capstone”
* **Content**:
  + Show a quick wireframe or mockup of the front-end.
  + Talk about your approach to state management.

**Days 92–94**

**Task Focus**

* **Setting Up the Back-End**: Node/Express server, user authentication with JWT, basic models.
* Validate data for your main domain models.

**LinkedIn Post Idea (Pick any day in this range)**

* **Title**: “Capstone Progress: Building the Server Foundations”
* **Content**:
  + Show your initial code structure for controllers, routes, models.
  + Reflect on an authentication challenge you solved.

**Days 95–97**

**Task Focus**

* **Front-End Scaffolding**: React app with routes for login, signup, main dashboard.
* Integrate with back-end auth, set up global state (Redux/Context).

**LinkedIn Post Idea (Pick any day in this range)**

* **Title**: “Capstone Progress: Front-End Integration”
* **Content**:
  + GIF or screenshot of logging in from the React app.
  + Talk about how you manage auth states across components.

**Days 98–100**

**Task Focus**

* **Core Features Implementation**: main CRUD features (e.g., posts, products, tasks).
* Add file uploads if relevant, ensure robust error handling.

**LinkedIn Post Idea (Pick any day in this range)**

* **Title**: “Capstone Progress: Core Features Taking Shape”
* **Content**:
  + Summarize a major feature you just completed.
  + Show a snippet or short video demo.

**Days 101–102**

**Task Focus**

* **Basic Testing & Refinement**: write unit/integration tests, do manual QA.
* Prepare for final deployment soon.

**LinkedIn Post Idea (Pick any day in this range)**

* **Title**: “Capstone Progress: Testing & Polishing”
* **Content**:
  + Screenshot of test results or coverage.
  + Mention any performance or UI tweaks.

**Final 18 Days (Days 103–120): Capstone Completion & Polish**

**Days 103–105**

**Task Focus**

* **Advanced/Optional Features**: Payment integration (Stripe), real-time with Socket.io, email notifications (Nodemailer), or any other feature that adds value.

**LinkedIn Post Idea (Pick one day to post)**

* **Title**: “Capstone Update: Adding Extra Spark with [Feature Name]”
* **Content**:
  + Demo the new feature (e.g., Stripe payments, real-time chat).
  + Talk about the complexities you faced and overcame.

**Days 106–110**

**Task Focus**

* **UI/UX Polish & Performance**: refine layouts, add loading states/skeletons, image optimizations.
* Make sure it looks good across all devices.

**LinkedIn Post Idea (Pick one day to post)**

* **Title**: “Capstone Update: Improving the Look & Feel”
* **Content**:
  + Before/after screenshots of the UI.
  + Show any performance metrics or Lighthouse scores.

**Days 111–112**

**Task Focus**

* **Security & Deployment**: finalize JWT security, hashing passwords with bcrypt, deploy front-end + back-end.
* Use environment variables for production.

**LinkedIn Post Idea**

* **Title**: “Capstone Update: Securing and Deploying the Final Build”
* **Content**:
  + Link to your production environment.
  + Briefly mention how you protect sensitive data with .env.

**Days 113–115**

**Task Focus**

* **Thorough Testing & Final Bug Fixes**: get feedback from peers, fix any major issues.
* Confirm performance is acceptable in production.

**LinkedIn Post Idea**

* **Title**: “Capstone Update: Bug Hunting & Final Touches”
* **Content**:
  + Mention key bugs you squashed and how.
  + Ask your network to test your app and share feedback.

**Days 116–117**

**Task Focus**

* **Documentation & README**: clear setup instructions, usage guide.
* Document API endpoints in a simple markdown or with Swagger.

**LinkedIn Post Idea**

* **Title**: “Capstone Update: Documenting for Future Devs (and Myself)”
* **Content**:
  + Screenshots of your README or Swagger docs.
  + Emphasize why good documentation matters to teams and recruiters.

**Days 118–119**

**Task Focus**

* **Final Presentation & Portfolio Update**: create a short demo video, updated LinkedIn highlights.
* Add the project to your portfolio site, pin the repo on GitHub.

**LinkedIn Post Idea (Day 118)**

* **Title**: “Day 118: Capstone Demo & Portfolio Revamp”
* **Content**:
  + Link to your portfolio with a short teaser video.
  + Encourage viewers to share it with anyone looking for a full-stack dev.

**LinkedIn Post Idea (Day 119)**

* **Title**: “Day 119: Ready to Launch — Final Capstone Teaser”
* **Content**:
  + Summarize the journey from day 1 to now.
  + Thank mentors, friends, or the community that supported you.

**Day 120**

**Task Focus**

* **Completion & Reflection**: reflect on your entire 120-day journey.
* Identify areas for continued growth. Celebrate!

**LinkedIn Post Idea**

* **Title**: “Day 120: Challenge Complete — My Full-Stack Journey So Far”
* **Content**:
  + Celebrate your progress and achievements.
  + List major skills gained (HTML, CSS, JS, React, Node.js, Express, MongoDB).
  + Encourage others to embark on a similar journey.

**Additional Engagement Tips for Each Post**

1. **Ask for Feedback**: End with a question (“Any tips on advanced debugging?”).
2. **Use Hashtags**: #100DaysOfCode, #WebDevelopment, #FullStack, #JavaScript, #NodeJS, #React, etc.
3. **Tag Influencers or Mentors**: If you used a specific tutorial or resource, tag the author.
4. **Polls**: Occasionally run a LinkedIn poll asking about user preferences (e.g., “Docker vs. Vagrant?”).
5. **Video Demos / GIFs**: Show short visuals of your app in action to increase engagement.

**How to Use This Single Prompt**

* Copy this entire text into a document or note-taking app.
* Each day, come back to see what you’re supposed to learn/code and how to frame your LinkedIn post.
* Tweak the daily tasks or the post ideas as needed to fit your style or schedule.
* Over 120 days, you’ll build a robust portfolio, **document your process**, and **showcase your problem-solving abilities** to potential employers or internship recruiters.

**That’s it!** You now have a **120-day, day-by-day plan** complete with **LinkedIn post ideas** to document your progress. By consistently following these tasks and sharing your journey, you’ll **grow your skill set, build a portfolio** worth showing off, and make your LinkedIn profile stand out in the eyes of **hiring managers** and **fellow developers**. Good luck and enjoy the ride!