**Tooling Selection and Justification**

**Programming Languages**

**JavaScript**

**Purpose: Primary language for both front-end and back-end development.**

**Justification: Universally supported, enabling a full-stack JavaScript solution. Ideal for seamless integration between front-end (React) and back-end (Next.js).**

**Integration: Works perfectly with frameworks like React and Next.js, ensuring consistency across the project.**

**SCSS (Sassy CSS)**

**Purpose: CSS preprocessor to create maintainable and scalable styles for the front-end.**

**Justification: Provides advanced styling options (e.g., variables, nesting) essential for building a visually appealing interface.**

**Integration: Easily integrates into the Next.js framework through SCSS module support.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Frameworks**

**React**

**Purpose: Component-based JavaScript library for front-end development.**

**Justification: Encourages modular development, simplifies state management, and provides excellent performance for interactive user interfaces.**

**Integration: Fully integrated with Next.js for server-side rendering and dynamic routing.**

**Next.js**

**Purpose: Full-stack framework for React, providing server-side rendering (SSR), static site generation (SSG), and API routes for backend logic.**

**Justification:**

**SSR/SSG: Improves performance and SEO by rendering pages on the server.**

**API Routes: Built-in backend capabilities eliminate the need for a separate server, reducing complexity.**

**Routing: Simplifies dynamic routing without additional libraries.**

**Integration: Handles both front-end and back-end functionality, working seamlessly with React for UI and providing a robust API layer.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Integrated Development Environment (IDE)**

**Visual Studio Code (VS Code)**

**Purpose: Central tool for writing, debugging, and managing code.**

**Justification: Rich extension ecosystem, including support for React, Next.js, and SCSS development.**

**Integration: Works with Git, GitHub, and project management tools, making it the hub for all development activities.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project Management Software**

**Trello**

**Purpose: Task management and milestone tracking.**

**Justification: Simple and effective for small teams. Customizable boards align perfectly with agile methodologies.**

**Integration: Linked with GitHub to track commits and progress alongside tasks.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Version Control System**

**Git + GitHub**

**Purpose: Version control and collaboration platform.**

**Justification: Industry standard for managing code repositories, enabling efficient collaboration and tracking.**

**Integration: Used alongside VS Code, GitHub Actions, and Trello for seamless project management and CI/CD workflows.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CI/CD System**

**GitHub Actions**

**Purpose: Automates tasks such as testing, building, and deploying the application.**

**Justification: Direct integration with GitHub, simplifying setup and reducing costs for small-scale projects.**

**Integration: Automatically tests and deploys changes made to the Next.js application, ensuring high code quality and performance.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Design Tools**

**Figma**

**Purpose: Collaborative design tool for creating UI/UX mockups.**

**Justification: Enables real-time collaboration and quick feedback loops, essential for visually appealing designs.**

**Integration: Designs can be exported directly into SCSS and React components, speeding up the front-end implementation.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Collaboration Platform**

**Slack**

**Purpose: Communication and team coordination.**

**Justification: Provides integration with GitHub, Trello, and other tools, ensuring smooth communication throughout the project.**

**Integration: Centralizes communication and alerts from GitHub Actions and Trello.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Backend-Specific Tools**

**Postman**

**Purpose: API testing and debugging tool.**

**Justification: Simplifies the validation of Next.js API routes, ensuring that endpoints function as expected.**

**Integration: Works alongside Next.js to test and debug API routes during development.**

**Next.js API Routes**

**Purpose: Provides server-side logic directly within the Next.js framework.**

**Justification: Eliminates the need for a separate back-end server, simplifying the stack while allowing for backend logic, such as user authentication or database queries.**

**Integration: Fully integrated into the Next.js framework, enabling unified development for front-end and back-end within a single repository.**

**Database (Optional – MongoDB)**

**Purpose: NoSQL database for storing user data, beer ratings, and social interactions.**

**Justification: Scalable and flexible schema, ideal for dynamic applications like Pivove.**

**Integration: Works with Next.js API routes using libraries such as Mongoose for seamless data handling.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**How Tools Integrate and Complement Each Other**

**Front-End and Back-End: React (front-end) and Next.js (front-end and back-end) form a unified full-stack solution, ensuring consistency in development and smooth communication between the UI and the API.**

**Development Workflow: VS Code acts as the central workspace, with Git and GitHub managing version control. GitHub Actions automates the CI/CD pipeline to test and deploy the Next.js application.**

**Collaboration and Design: Figma provides design prototypes, which are implemented in SCSS and React. Slack ensures communication between team members, while Trello tracks tasks and progress.**

**Backend Development: Postman is used to test and validate Next.js API routes, which can handle authentication, database queries, and other server-side logic.**