**Tooling Selection and Justification**

**Programming Languages**

1. **JavaScript**
   * **Purpose:** Primary language for front-end development to create dynamic, interactive features.
   * **Justification:** Widely supported by browsers, has an extensive ecosystem (e.g., libraries like React), and is beginner-friendly for collaboration.
   * **Integration:** Seamlessly integrates with SCSS for styling and Node.js if needed for back-end development.
2. **SCSS (Sassy CSS)**
   * **Purpose:** CSS preprocessor to enhance styling capabilities with variables, nesting, and mixins.
   * **Justification:** Simplifies complex styling and supports maintainable and reusable styles. Ideal for a visually appealing interface.
   * **Integration:** Compiles into standard CSS and works directly with JavaScript-driven front-end frameworks.

**Frameworks**

1. **React**
   * **Purpose:** JavaScript library for building user interfaces.
   * **Justification:** Efficient component-based architecture ensures scalability and reusability. Its ecosystem includes tools like React Router and state management libraries.
   * **Integration:** React easily integrates with JavaScript, SCSS, and REST APIs, ensuring a cohesive development flow.

**Integrated Development Environment (IDE)**

1. **Visual Studio Code (VS Code)**
   * **Purpose:** IDE for writing, debugging, and managing code.
   * **Justification:** Lightweight, feature-rich, and supports extensions like ESLint, Prettier, and SCSS Lint.
   * **Integration:** Works seamlessly with Git and project management extensions.

**Project Management Software**

1. **Trello**
   * **Purpose:** Tool for tracking tasks and milestones.
   * **Justification:** Simple, intuitive interface with Kanban-style boards perfect for managing project workflows.
   * **Integration:** Integrates with GitHub for tracking commits and provides a shared platform for the team.

**Version Control System**

1. **Git + GitHub**
   * **Purpose:** Version control and code repository.
   * **Justification:** Ensures collaborative development with features like branching and pull requests. GitHub provides additional benefits such as issue tracking and project boards.
   * **Integration:** Works with VS Code, CI/CD pipelines, and collaboration tools like Slack.

**CI/CD System**

1. **GitHub Actions**
   * **Purpose:** Automates testing, building, and deployment.
   * **Justification:** Fully integrated into GitHub repositories and provides free-tier pipelines for small-scale projects.
   * **Integration:** Triggers automated builds and tests on every push, ensuring code quality.

**Design Tools**

1. **Figma**
   * **Purpose:** Collaborative design and prototyping platform.
   * **Justification:** Enables team members to design mockups and gather feedback in real-time. Cloud-based, making it accessible for everyone.
   * **Integration:** Export assets directly into SCSS and JavaScript workflows.

**Collaboration Platform**

1. **Slack**
   * **Purpose:** Communication and collaboration tool.
   * **Justification:** Real-time messaging, integration with tools like GitHub and Trello, and support for custom notifications.
   * **Integration:** Centralizes communication for all team members, ensuring smooth project coordination.

**Other Relevant Tools**

1. **Postman**
   * **Purpose:** API testing and debugging.
   * **Justification:** Simplifies testing API endpoints and ensures proper integration with the front-end.
   * **Integration:** Works alongside React and GitHub Actions to test and validate APIs.

**How Tools Integrate and Complement Each Other**

* **Development Workflow:** VS Code serves as the central workspace, with Git and GitHub managing version control. React, JavaScript, and SCSS power the front-end, with Trello tracking development milestones.
* **Collaboration:** Slack facilitates team communication, while Figma provides a shared design platform.
* **Quality Assurance:** GitHub Actions automates CI/CD pipelines, ensuring that changes are tested and deployed efficiently. Postman validates API functionality, guaranteeing smooth integration between back-end and front-end.