

Potential Risk	Risk Level	Mitigation Strategy
Team members unfamiliar with React.js / PostgreSQL	Medium	Schedule early learning sessions, use online tutorials, and simplify functionality if needed.
Limited time due to short project duration and coursework workload	High	Create a weekly plan, prioritize MVP (core features), and conduct weekly progress checks.
Team member absence (illness, scheduling conflicts)	Medium	Assign backup responsibilities, ensure at least two members understand each module, and share code via GitHub.
Requirement changes (feedback from instructor)	Medium	Adopt iterative development (Sprints), reserve buffer time, and use modular design to reduce change cost.
Data loss or version conflicts in team collaboration	Medium	Use Git/GitHub for version control, follow a branching strategy, and back up the database regularly.
UI/UX complexity delays progress	Low	Focus on functional correctness first, polish the UI at later stages.
Insufficient testing leads to defects	Medium	Perform unit and manual tests after each feature completion to validate incrementally.
Deployment or environment compatibility issues	Low	Standardize development environment (Node.js version, DB setup) early and provide README deployment guide.