LESSONS LEARNED REPORT	
Project Name	Cookify
Project Sponsor	Yogesh Sharma
Project Manager	Hashir Owais
What went well during the project?	
Effective Problem Identification: The team clearly defined the problem of food waste and lack of recipe personalization, which helped align the project goals with societal and environmental needs.	
Environmental and Societal Impact Considerations: The application promotes sustainable cooking practices by reducing food waste and encouraging home-cooked meals, which aligns with the project's goal of contributing to environmental and societal betterment.	
Solution Design and Implementation: The hybrid solution combining an external API for real-time data and a local database for user personalization successfully addressed key project goals, including scalability and user-friendliness.	
Successful Feature Integration: Core features like real-time ingredient matching and personalized recipe recommendations were seamlessly integrated, enhancing the user experience.	
Technology Stack Selection: Using open-source tools such as MongoDB, Express.js, and Passport.js minimized costs and time while delivering a robust and secure application.	
Team Coordination: Regular meetings and clear division of tasks facilitated smooth project management and allowed milestones to be achieved on schedule.	
Scalability Considerations: The design and implementation allowed for future expansion, such as integrating additional APIs or advanced filtering options.	
What did not go well during the project?	
Learning Curve for Technology Stack: Some team members faced challenges adapting to new technologies like Passport.js and MongoDB, which delayed development timelines and introduced initial inefficiencies but were overcome with the guidance of the ENSE lab.	

Unclear Milestones in Early Stages: Early project stages lacked well-defined milestones, which created minor delays in aligning the team's efforts. This was due to confusion regards to the GitHub submissions and commits.	
Minor Git Conflicts: Frequent merging of changes without proper synchronization caused minor Git conflicts, leading to additional resolution time being needed.	
What should we do differently next time?	**How will this be done?**
Improve Git Management Practices:	Standardize Git usage with clear guidelines, such as branch naming conventions, commit message protocols, and frequent pull requests.
Strengthen Stakeholder Involvement:	Engage stakeholders more frequently through progress meetings and interactions to ensure alignment with their expectations and receive valuable feedback.
Improve Time Management by Addressing Priorities Early:	Use a priority matrix to evaluate tasks based on impact and urgency, ensuring that critical tasks like setting up core features are completed first.