**CS691, Capstone Project - Retrospective Report**

**Iteration: Sprint 1**

Team No: 2

Project Manager: Hani Thummar

Application Name: Bitebuddy

Meeting Date:03/25/2025

Mandatory points to address:

* Velocity (effort) estimation for iterations
* Following the schedule of deliverables
* Slack communication
* Peer-reviews

|  |  |  |
| --- | --- | --- |
| **What Worked Well** | **What can be Improved** | **Suggested Improvement Actions** |
| * Successfully integrated React on the front end with Node.js, Express, and MySQL on the backend, demonstrating the ability to build a complete web application. * Restricted restaurant owner and diner privileges using middleware. * Labeled the API names to be meaningful.   **Reliable Deliverables**   * We delivered all the features decided in Sprint 1 as per timeline.   **Velocity Estimation**   * We successfully completed planned story points, providing a solid baseline for planning upcoming sprints. * Code review meetings and standups used to happen on time and we had full attendance. | * Implementing redux for state management. * At times, communication on the Slack channel was not synchronized. | * Load images & components only when needed. |
| **Jenkins**   * Jenkins was successfully installed on the server without major issues. * Installation process was smooth with clear documentation and support from the team. * Jenkins' configuration and setup were completed in a reasonable timeframe. * Automation through Jenkins helped ensure consistency and repeatability across deployments.   **Slack Integration**   * Slack integration for build notifications was successfully set up and was helpful for the team to track deployment status. | * Initial setup could have been automated or streamlined to avoid manual configuration steps. * There were minor configuration errors during the first setup that delayed the process. * The deployment times were longer than expected, especially for larger builds. * Slack notifications could have been more informative (e.g., including specific error logs, detailed information on build failures). | * Create an automated script for Jenkins installation and configuration to ensure a faster and error-free setup. * Consider using pre-configured Jenkins Docker images for faster setup in future projects. * Optimize Jenkins pipelines to improve deployment speed (e.g., parallelizing tasks, reducing build steps). * Create and maintain environment-specific configurations to avoid discrepancies between Dev and QA. * Customize Slack notifications to include more detailed build logs and error messages. |
| **Peer-Reviews**   * Peer reviews provided valuable feedback that enhanced the clarity and quality of documentation. | * Feedback loops were sometimes inefficient, causing delays in finalizing documentation. | * Create a systematic review process with defined guidelines and a checklist. |