**Lab Rats – Project Deliverable Report**

## FastBite

**Contributing Members:**

* Neel Panajkar
* Kaksh Patel
* Tyler Quach
* Roger Lin
* Aayush Niroula

**Project Description** - *FastBite* is a full‑stack fast‑food ordering platform that lets customers browse a live menu, customize items, pay online, and track orders in real time. For the restaurant, FastBite provides an operational dashboard to manage incoming orders, assign delivery drivers, and update inventory, ensuring the menu remains accurate. The application supports two synchronized portals (Customer and Restaurant) backed by a single database and real‑time event stream.

**Technology Stack:**

* **Programming Language:-** JavaScript
* **Frontend Framework:-** React (Vite), React Router, Tailwind CSS
* **Backend Framework:-** [Express.js](http://express.js), WebSockets
* **Database:-** MongoDB (Atlas, Mongoose)
* **DevOps/VCS:-** GitHub

**Users & Roles:**

* **Customer:-** Signup/ Login, browse menu, configure items, cart/checkout, track order status, order history, profile & saved payment methods.
* **Restaurant:-** View Orders, Accept/ Reject Orders, set ETA, update order status, update inventory, assign driver, manage menu

## Functional Requirements

**Customer Portal**

1. Account management: register/login/logout (JWT), profile, saved addresses.
2. Browse/search menu; item detail with modifiers (size, toppings, spice level, add‑ons).
3. Cart management (add/edit/remove, coupon code, tax/shipping estimates).
4. Checkout with payment (Phase 2 Stripe); cash on delivery (Phase 1 option).
5. Real‑time order status updates + notifications (web push/email/SMS).
6. Order history & re‑order.

**Restaurant Portal**

1. Authentication with role‑based access (Manager, Staff, Driver).
2. Live order queue with filtering (new/in‑progress/ready/out‑for‑delivery).
3. Status transitions with timestamps; estimated prep time.
4. Driver assignment
5. Menu management: categories, items, options, pricing, taxes.
6. Reporting: sales, top items, cancellations, and delivery lead time.

## User Interface (Wireframe‑Level Description)

**Customer Views**

* Home/Menu: Category chips, menu cards, badges for *New*, *Spicy*, *Sold Out*.
* Item Detail: Modifiers (radio/checkbox/quantity), allergen tags
* Cart & Checkout: Address selector, slot selection, payment method, order notes.
* Order Tracker: Progress bar (5 states), live ETA, and driver assigned card.
* Profile: Addresses, preferences.

**Restaurant Views**

* Orders Dashboard: Columns by status; drag‑and‑drop or action buttons to advance state; sound alerts on new orders.
* Menu Builder: CRUD for categories/items/options with pricing and tags.
* Drivers: Roster, assignment panel.
* Analytics: Sales summary, cancelled orders, and delivery SLA.

## Risk Management

1. Incompatibility issue between design and software development. The software doesn’t match the interface or functionality planned in the design.
   1. **Probability**: Medium
   2. **Impact**: High
   3. **Mitigation**: Members meet together to discuss their implementations and ensure they are on the same track. Details will be defined and documented.
2. Team members fall behind and are unable to complete their part before the deadline.
   1. **Probability**: High
   2. **Impact**: Medium
   3. **Mitigation**: The project coordinator and other members will consistently check in with each other and their progress, as well as assist each other with any needs.
3. Poor user interface design could frustrate users and reduce the adoption of the app.
   1. **Probability:** Medium
   2. **Impact:** Moderate
   3. **Mitigation:** Perform usability testing and iterate on designs to improve clarity and ease of use. Ensure that the user interface follows usability heuristics and is intuitive for users.

## Timeline

Project Proposal and Risk Management

* Identify risks and risk management methods Deadline: 8/30
* Project Proposal Deadline: 9/01
* Schedule Creation Deadline: 9/01

System Specification and Requirements

* Create documentation for the system Deadline: 9/18
* Create UML diagram Deadline: 9/10

Design

* User Interface Mockup Deadline: 9/22
* Database design Deadline: 9/22
  + Create collections for items Deadline: 10/11
  + Create collections for users Deadline: 10/11

Development

* Implement core Front End Functionality Deadline: 10/22
* Implement core Back End Functionality Deadline: 10/22
* Additional Front End Deadline: 11/10
* Additional Back End Deadline: 11/10

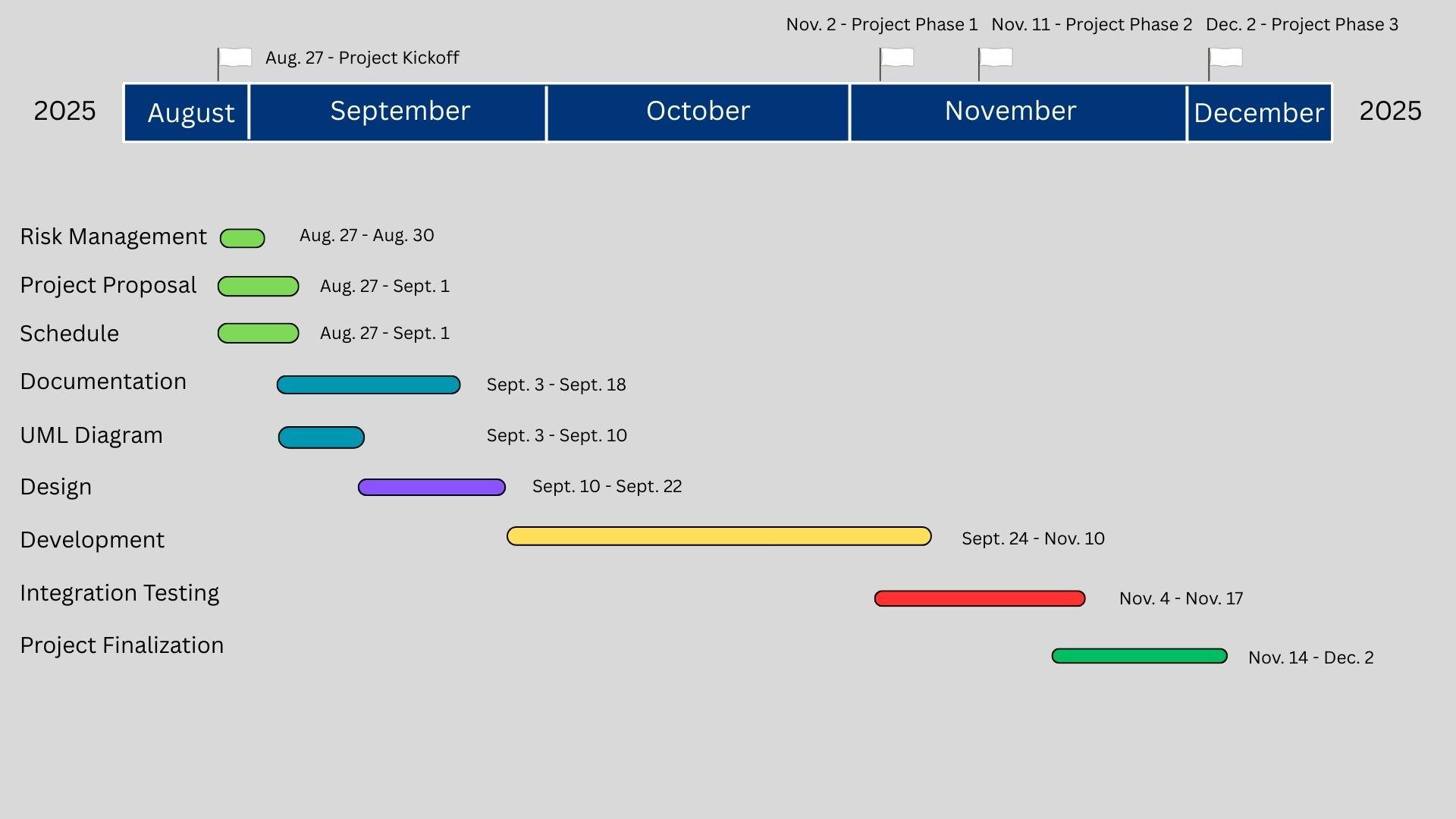
Testing

* Assess components and create tests Deadline: 10/30
* Integration testing for core functionalities Deadline: 10/30
* Assess components and create tests Deadline: 11/13
* Integration testing for the whole application Deadline: 11/17

Launch

* Project Finalization Deadline: 11/30
* Launch and host Deadline: 12/1

[**FastBite - Gantt Chart**](https://docs.google.com/spreadsheets/d/1eMBpwVfds4G6vnvVk3gXf1sNwPDa3Rgv-o-c-LAkqnM/edit?usp=sharing)

****

**Team Contributors**

| **Name** | **Contribution Description** | **Overall Contribution (%)** |
| --- | --- | --- |
| Neel Panajkar | Project Management Lead,  Implementation Lead for Back-end | 20% |
| Kaksh Patel | Requirement Lead, Documentation Lead | 20% |
| Tyler Quach | Design Lead  Implementation Lead for Front-end | 20% |
| Roger Lin | Testing Lead  Configuration Management Lead | 20% |
| Aayush Niroula | System Administrator Lead, Demo and Presentation Lead | 20% |