Project Status Report I

Date of Report Issue/Prepared: Feb/01/2025

Report Prepared By Team Members: Fatima Arab, Nigar Ahmadova, Anar Samadzadeh, Elizabeth Thomas

Employer/Organization (if applicable) : George Brown College

Project Name:	Biz Horizon	
Project Team:	T101	
Period Reporting:	Start Date: **Feb 01, 2025**	End Date: **March 09, 2025**
Overall Project Health	Green (Good) - Yellow (Warning) - Red (Bad)	

Summary

Project Status Summary
The project, Biz Horizon, has made steady progress, focusing on foundational work in backend development, payment integration, and frontend structure. The backend APIs for registration, payments, and catalog fetching were set up, and Stripe integration was successfully tested in a sandbox environment. On the frontend, basic components such as the user registration form, product catalog layout, and booking form skeleton were developed, though full integration with the backend is still pending. The layout and initial features, like managing customers and orders using dummy data, were implemented for the admin dashboard. While some tasks, such as payment workflow testing and UI-backend integration, remain incomplete, the project is on track to achieve its objectives with focused efforts in the next phase.

Accomplishments As Planned	Planned but not Accomplished
Nigar • Implemented transaction handling (request validation, response formatting, error handling). • Developed Payment API endpoints (POST /api/payment) and validated with Postman. • Successfully integrated Stripe sandbox environment, ensuring payments process correctly. • Improved error-handling for faile9d payments (insufficient funds, incorrect details). • Ensured consistent API response structure across all payment endpoints. • Tested payment failure scenarios. • Ensured successful payment status updates in the database. • Unified System Integration (Backend, Frontend, APIs, Admin Dashboard, Payments) and Connected all separately implemented Sprint 5 & 6 components into one project: • Backend APIs (Elizabeth's work). • Frontend UI (Anar's work). • Admin Dashboard (Fatima's work). • Payment System (Her own work) • Merged API functionalities into a single structured backend. • Fixed inconsistencies in data flow between frontend & backend. • Ensured successful data transactions (Orders, Bookings, Payments) across modules.	• The Payment UI needs to be connected to the backend for real user interactions. • End-to-end testing of the fully merged system is still required. • Live Payment Testing is incomplete due to webhook configuration delays.

Accomplishments As Planned	Planned but not Accomplished
Anar • Developed basic UI navigation for Customers, Orders, and Payments. • Ensured responsive design & sidebar menu for the dashboard. • Connected Login & Registration UI with backend authentication API. • Integrated basic product data fetching from API. • Developed product listing UI with grid layout. • Added basic order status tracking UI in the Admin Panel. • Created UI structure for booking form & order tracking. • Developed and checked UI for Admin Dashboard.	• Product management page still lacks full CRUD operations. • No UI to add, update, or delete products yet. • Booking form validation & backend submission is missing. • Live order status updates are not implemented.

Accomplishments As Planned	**Planned but not Accomplished**
Elizabeth • Develop Analytics API (Generate business insights). • IntegrateD Orders, Products, Payments, Customers, and Analytics APIs with the database. • Developed API for Orders (Create, Read, Update, Delete). • Tested API endpoints in postman. • Optimized API performance & security (JWT authentication, request validation). • Conducted backend performance testing (Optimize database queries, prevent API bottlenecks). • Finalized backend security improvements (Ensure secure API endpoints, role-based access control). • Fixed any API-related issues before passing the project to Fatima. • Created Dashboard Analytics Section (Display sales reports. • Designed and added a launch screen. • Implemented logged-in profile page to current admin page.	• Develop Analytics API for top-selling products, customer stats. • Data validation and error handling for all API responses. • Create Dashboard Analytics Section (top-selling products, customer activity). • Filters & search for orders in the Admin Panel

Accomplishments As Planned	**Planned but not Accomplished**
Fatima **Feedback (Backend & Frontend):** • Developed the backend for the feedback system, including API endpoints for: o Retrieving feedback for display o Managing feedback in the admin panel • Implemented the frontend UI for: o Admin to view and manage feedback • Integrated the feedback system with the database, ensuring proper data storage and retrieval. • Tested feedback API endpoints using Postman to verify successful interactions between frontend and backend. • Adjusted the UI to match the admin dashboard theme for consistency. • Created basic admin panel layout for managing users and customers. • Developed Admin Dashboard backend and frontend functionalities.	• Some CSS elements may still need revisions for better responsiveness and consistency. • Minor UI adjustments may be needed to improve user experience and make the feedback section more intuitive. • Connecting the Orders/Bookings section to the backend API and implementing order management UI improvements.

Upcoming Objectives for March 10, 2025, to Mar 27, 2025

Planned Activities/Tasks for Next Period			
Activity/Task	Assigned To <name/s>	Duration <days>	Date
• Implement API endpoints for generating sales analytics, customer insights, and product statistics. • Optimize database queries for orders (indexing, caching, query optimization). • Implement API rate limiting and request validation to prevent abuse. • Test and fix data flow issues and ensure correct real-time updates.	Elizabeth	7 days	March 17
• Ensure users can leave reviews & ratings for products/services. • Verify API functionality (Orders, Products, Feedback, Analytics). • Fix any errors in API calls, request validation, and security. • Implement search & filtering features for better order tracking.	Fatima	7 days	March 17
• Test the admin panel for usability issues (navigation, responsiveness, accessibility). • Implement layout fixes and enhance UI components for better user experience. • Ensure full integration of Orders & Product UI with backend APIs. • Explicitly test Orders & Payments UI flow.	Anar	7 days	March 17
• Test full system workflow: User Registration → Catalog Browsing → Booking → Payment → Feedback → Analytics Dashboard. • Identify and fix any remaining issues before project submission.	All team members	5 days	March 22
• Deploy backend, frontend, and database to AWS/Azure. • Set up CI/CD pipelines for automated deployment and testing. • Write API documentation, database schema, and setup guide. • Provide step-by-step installation and deployment instructions.	Nigar	5 days	March 27

	Milestones for Next Period	
Milestone (Objective)	Assigned To	Delivery Date
• Develop API routes for creating, updating, and deleting orders. • Implement search and filtering features for better order tracking. • Ensure database integration for storing and retrieving order details. • Validate Orders, Payments, and Feedback APIs for response codes, authentication, and security issues. • APIs, validating response codes, error handling, and authentication mechanisms.	Fatima	March 15
• Develop API endpoints for Sales Analytics, Customer Insights, and Product Statistics. • Optimize database queries (Indexing, Caching, Query Optimization) for Orders & Analytics. • Implement API rate limiting, request validation, and centralized logging for security.	Elizabeth	March 14
• Design a functional UI for admins to manage orders easily. • Ensure full integration of Orders & Product UI with backend APIs, testing proper data flow between UI forms and database transactions. • Check, identify, and fix UI responsiveness issues in the Admin Dashboard.	Anar	March 17
• Test and validate API functionality by checking CRUD operations, request validation, response time, security vulnerabilities, and data consistency for Orders, Products, Feedback, Payments, and Analytics APIs. • Conduct Full End-to-End Testing (User Registration → Booking → Payment → Feedback → Analytics Dashboard).	All team members	March 22
• Deploy backend, frontend, and database to AWS/Azure, set up CI/CD pipelines for automated deployment, monitor logs for errors, and ensure proper environment configuration. • Complete API documentation, database schema, and installation guide.	Nigar	March 27

Managing Issues and Risk (for current reporting period and next period)

Issues/Problems	Resolution Strategy	Due Date
CSS Styling Issues: The feedback section's UI, buttons and feedback table did not match the rest of the dashboard, affecting consistency.	Adjusted **CSS styles** to ensure proper alignment, fonts, and colors to match the overall theme.	March 13, 2025
Database Issues: Feedback entries were not saving correctly due to schema mismatches and validation errors.	Updated the database schema, ensured required fields were properly handled, and tested using real data.	Resolved
API Route & Endpoint Issues: Some feedback API routes were not working due to incorrect request handling.	Debugged and fixed feedback Routes.js, verified request methods, and ensured proper integration with the frontend.	Resolved
Frontend & Backend Integration: Feedback data for users' names was not displayed correctly after submission.	Fixed API call responses, handled state updates in the frontend, and ensured real-time data retrieval.	Resolved
API Performance Bottleneck: API response times are slow for large data queries (Orders, Payments, Analytics).	Optimize database queries with indexing & caching, reduce API payload size, and use pagination for large datasets.	March 15, 2025
Payment Status Sync Failure: Payment statuses in the Admin Panel are not updating after transactions.	Ensure backend webhook listeners properly update the database and implement a refresh trigger in UI.	March 16, 2025
Payment Gateway Integration Issues: Stripe/PayPal integration issues with live mode payments.	Verify all live API keys and webhook responses. Test end-to-end transactions with real payment methods.	March 20, 2025
Deployment Configuration Errors: AWS/Azure deployment issues due to environment variable misconfigurations.	Use a .env file & secrets manager, validate cloud configurations in a staging environment before production.	March 24, 2025
CI/CD Deployment Failures: CI/CD pipeline failing during automated deployment due to dependency conflicts.	Fix broken dependencies in package.json, update Dockerfile configurations, and review build logs for errors.	March 25, 2025

Upcoming Risks	Risk Ranking (Hi, Med, Low)	Risk Impact (Hi, Med, Low)	Mitigation Strategy
Unforeseen API Downtime: Sudden backend failures could disrupt frontend interactions.	High	High	Implement API health monitoring, set up automated failover mechanisms, and maintain server redundancy.
Live Payment Processing Errors: Payment gateway could reject live transactions unexpectedly.	High	High	Perform pre-launch live payment testing, maintain backup payment methods, and monitor real-time transactions.
Scalability Issues with Increased User Load: High traffic may overload the database and slow down performance.	Medium	High	Conduct load testing, use database optimization techniques, and enable horizontal scaling if needed.
Unresolved Bugs in Admin Dashboard: Post-deployment, critical admin functions might not work correctly.	Medium	High	Perform comprehensive post-launch testing, create a hotfix strategy, and have immediate patch releases ready.
Delays in Final Testing & Deployment: Unexpected technical issues may push back the final release timeline.	Medium	High	Set up strict testing deadlines, enforce a buffer period before deployment, and prioritize high-impact fixes first.
Misalignment Between Frontend & Backend Features: Some features may not work as expected due to changes in API or UI.	Medium	High	Keep frontend-backend members in sync with regular integration meetings and shared documentation updates.
Third-Party Service Downtime: External APIs (Stripe, AWS, PayPal) may go offline unexpectedly.	Medium	High	Implement fallback mechanisms, cache last successful responses, and set up alternative service options.
Team Availability Issues: Any team member being unavailable (illness, unexpected workload) could delay critical tasks.	Medium	High	Assign backup responsibilities, maintain clear documentation, and ensure cross-functional team knowledge sharing.

NOTE: Attach additional sheets if insufficient space is available.

Submission Guidelines:

Please submit as “W25_T<team number>_ProjectReport1”.
For e.g., W25_T99_ProjectReport1

This is a group submission i.e. one per group.

Due Dates:

Sunday, March 09, 2025 (11:59 p.m.)