# Campus One Delivery Project Proposal

## 1. Project Title

CampusOne Delivery: A Web-Based Food Ordering & Campus Delivery Management System

Project Type: Web Application

POC for Senior Project: No

## 2. Team Members

Member 1: Paing Min Thant  (6622055)

Member 2: Khant Nyi Thu (6632108)

Member 3: Pyae Zin Phyo (6715150)

## 3. Problem Statement & Motivation

Students on campus usually need to walk to the canteen, AU Mall, or AU Plaza to buy their meals. For students who have tight class schedules, this can be inconvenient because these food areas are located far from some faculty buildings. In many cases, students do not have enough time during short breaks to walk there, wait in line, and return before their next class.

The main users of this system are **students with limited break time**, along with **campus food vendors and administrators**.

This problem matters because a campus food delivery system allows students to order food online and have it delivered directly to their faculty buildings. Students can then eat comfortably on the first floor while resting before their next class, helping them save time and manage their schedules more effectively.

## 4. Project Scope & Features

- User registration and login (Student, Vendor, Admin)
- Browse campus food vendors and menu items
- Place food orders within the campus
- Track order status (Pending, Preparing, Delivered)
- Admin management of users, orders, and vendors
- Payment record tracking per order
- View most ordered vendor
- View most ordered item
- Monitor peak ordering time
- View total daily / weekly sales

## 5. Data Models

The system consists of multiple entities that support full CRUD (Create, Read, Update, Delete) operations and reflect the real workflow of campus food ordering and delivery.

---

### Entity 1: User

(Stores all system users including Admin, Vendor, and Customer)

**Fields:**
_id, name, email, phone, role (Admin, Vendor, Customer), campus, locationId

**Operations:**
• Create (Register User)
• Read (Login / View Profile)
• Update (Edit Profile Information)
• Delete (Delete Account)

---

### Entity 2: Vendor

(Represents food vendors operating within the campus)

**Fields:**
_id, userId, shopName, locationId

**Operations:**
• Create (Register Shop)
• Read (View Shop Details)

• Update (Edit Shop Information)
• Delete (Close Shop)

---

### Entity 3: MenuItem

(Stores food items offered by each vendor)

**Fields:**
MenuItem: _id, vendorId, itemName, category, price

**Operations:**
• Create (Add Menu Item)
• Read (View Menu Items)
• Update (Edit Item Details or Price)
• Delete (Remove Menu Item)

---

### Entity 4: Order

(Represents a food order placed by a customer)

**Fields:**
_id, customerId, vendorId, orderDate, orderStatus, totalPrice

**Operations:**
• Create (Place Order)
• Read (Track or View Order)
• Update (Update Order Status)
• Delete (Cancel Order)

---

### Entity 5: OrderItem

(Stores individual food items within an order)

**Fields:**
_id, orderId, menuItemId, quantity, price

**Operations:**
• Create (Add Item to Order)
• Read (View Items in Order)
• Update (Modify Quantity or Item Details)
• Delete (Remove Item from Order)

---

### Entity 6: Location

(Represents delivery locations within the campus)

**Fields:**
_id, locationName

**Operations:**
• Create (Add Location)
• Read (View All Locations)
• Update (Edit Location Name)
• Delete (Remove Location)

---

### Entity 7: Payment

(Records payment transactions for orders)

**Fields:**
_id, orderId, paymentMethod, paymentStatus, amount,paymentdate

**Operations:**
• Create (Process Payment)
• Read (View Payment Details / Receipt)
• Update (Update Payment Status)
• Delete (Void Transaction)

## 6. Technology Stack

Frontend: React.js, Tailwind CSS

Backend: Node.js with Next.js API Routes

Database: MongoDB

Deployment: Vercel, MongoDB Atlas