

FINAL PROJECT PROPOSAL

Viet Food

April 6th, 2025

PREPARED FOR

COM S 3190 - Construction of User Interfaces

Iowa State University Computer Science Department

PREPARED BY

Tam Minh Nguyen (tamminh@iastate.edu)

Bach Nguyen (ntbach@iastate.edu)

Table of Contents

1. Introduction
2. Purpose of the proposal
3. Goals & Objectives
4. Project Description
5. Project Path selection
6. Feature Ownership & Responsibility
7. Resources and tools
8. Data Sources and Management
9. File Structure and Project Organization
10. User Experience Views
11. Final Comments

1. Introduction

We, Tam Minh Nguyen and Bach Nguyen, are from team MN_6 and are both sophomores majoring in Computer Science and Data Science. Through our coursework in COM S 319, we have developed web development skills using HTML, CSS, and JavaScript, with practical experience building a product catalog website. In COM S 309, we also worked on frontend and backend collaboration to develop an Android app. Combining the knowledge from these courses, we are now embarking on our final project, which is a website for ordering Vietnamese food online.

In this project, users will be able to browse the menu, place orders, pay online, and leave reviews and feedback on the dishes they try. The website will be built using React (JSX) for the front end, Node.js for the backend, and MongoDB for the database to create a seamless, interactive, and user-friendly online food ordering and review submission platform.

2. Purpose of the proposal

This proposal aims to outline the development of an online Vietnamese food ordering website designed to offer users a seamless experience in ordering authentic Vietnamese cuisine. The platform will allow users to browse a wide variety of dishes, place orders, make payments, and leave reviews and feedback on their culinary experiences. This website will serve as a one-stop destination for food enthusiasts who wish to explore traditional Vietnamese dishes and enjoy the convenience of ordering them online.

By providing a user-friendly interface, secure payment options, and detailed information about each dish—including ingredients, pricing, and food categories—the platform aims to enhance the experience of dining on Vietnamese cuisine from the comfort of home. This project seeks to bridge the gap between food lovers and authentic Vietnamese food by offering an intuitive, interactive, and engaging online platform. Through this initiative, we aim to deliver a reliable, easy-to-use system that simplifies the ordering process while celebrating Vietnamese food's rich flavors and diversity.

3. Goals and Objectives

Our project's goals and objectives include:

- Developing an online food ordering system that allows users to explore a variety of Vietnamese dishes, place orders, and make payments seamlessly.
- Creating a user-friendly and interactive interface using React (JSX) that enables users to browse dishes filtered by categories such as region, dish type, ingredients, and cooking techniques.
- Providing detailed information for each dish, including its history, nutritional information, preparation methods, ingredient list, and price.
- Implementing a review and feedback system where users can rate dishes and share their experiences, contributing to a dynamic community around Vietnamese cuisine.
- Ensuring regular content updates by incorporating new dishes, seasonal specials, and culinary

insights to keep the platform relevant and engaging.

- Demonstrating proficiency in web development technologies, including React (JSX) for frontend development, Node.js for backend implementation, and MongoDB for reliable data storage to create a functional and engaging platform.
- Ensuring secure and seamless payment processing by integrating a payment gateway allows users to complete their food orders online easily.

4. Project Description

Our project is a Vietnamese food ordering platform that allows users to explore dishes, place orders, make payments, and leave reviews. The application is designed to be responsive, user-friendly, and fully functional, built with React for the front end and Node.js with Express for the back end. The backend will connect to a MongoDB database to store food data, user information, and order history.

Functionality and User Interactions:

1. Frontend (React): The frontend will provide a dynamic, interactive user interface. It will allow users to browse food items, view detailed information about each dish, add items to their cart, and proceed to checkout. Users can also log in, place orders, and leave reviews.
2. Backend (Node.js/Express): The backend will handle user authentication, food management, order processing, and payment integration. It will provide RESTful API endpoints to communicate with the frontend and perform CRUD operations on food items, user accounts, and orders.
3. Database (MongoDB): MongoDB will store food items, user details, orders, and reviews. Each collection will be structured to allow efficient CRUD operations and secure data management.

Planned Pages:

1. Login and Signup Pages: Users can create accounts and log in to manage orders, save preferences, and write reviews.
2. Home/Main View: Displays a list of available food items with brief descriptions, prices, and categories. Users can filter dishes based on region, ingredients, or dish type.
3. Entity Details/View: Each food item will have its own detailed page displaying the ingredients, price, allergy information, vegan status, and an "Add to Cart" button.
4. Checkout Page: Allows users to review their cart, enter delivery information, and select a payment method.
5. Order Confirmation Page: Confirms the user's order, shows the summary of the order, and provides estimated delivery time.
6. About/Team Info Page: Displays course details and team member information.

How Parts Work Together:

1. The frontend communicates with the backend via RESTful APIs to handle actions such as retrieving food data, submitting orders, and managing user authentication.

2. The backend interacts with the MongoDB database to fetch, update, or delete data based on user actions. For example, when a user places an order, the backend updates the database with the order details and status.
3. The frontend receives data from the backend (e.g., food list, order confirmation) in JSON format and displays it dynamically using React components.

By combining these technologies, the application will provide a smooth and interactive experience for users to explore, order, and review Vietnamese food

5. Project Path selection

For this project, we have chosen Option 1: Extend Midterm Project. Our goal is to take the Vietnamese food website we developed for the midterm and transform it into a full-fledged online food ordering platform. This new project will include features like food ordering, payment processing, and a review system, making the platform interactive and functional for users who want to explore and order Vietnamese food.

New Features and Enhancements:

1. Login and Signup Pages: We will add user authentication and account creation functionalities to allow users to sign in, track their orders, and leave reviews.
2. Home/Main View: The homepage will display a variety of Vietnamese dishes, with a brief introduction to Vietnamese cuisine. It will also feature filters for searching dishes by ingredients, region, or cooking methods.
3. Entity Details/Edit View: Clicking on a dish will show detailed information, such as its price, ingredients, allergies, vegan status, and nutritional information. Admins will be able to manage the menu by editing and removing items that are unavailable.
4. Checkout Page: This page will allow users to make payments using multiple methods and enter their delivery address for the order.
5. Order Confirmation Page: After completing the checkout process, users will be directed to a confirmation page summarizing the order details and the estimated delivery time.
6. Reviews System: Users will be able to leave reviews and ratings for each dish, helping others make informed decisions.
7. Admin Panel: Admins will have the ability to manage the menu, including adding, editing, and removing food items, as well as viewing user reviews.
8. About/Team Info Page: This page will provide course details and information about the team members, including their names and ISU emails.

Rebuilding the Existing Code:

1. Frontend: The existing HTML, CSS, and JavaScript code from the midterm will be fully redeveloped using React (JSX) components, ensuring a dynamic, responsive, and user-friendly experience.
2. Backend: The backend will be rebuilt using Node.js with Express, replacing the previous static functionality. The new backend will handle user authentication, food management, orders, and payments.
3. Database: We will integrate MongoDB to store food data, user information, and order history, ensuring data persistence and reliable communication between the front end and back end.

By redeveloping the project with these new features and using modern technologies, we aim to create a comprehensive and functional food ordering platform that showcases Vietnamese cuisine while providing users with a seamless ordering experience.

6. Feature Ownership & Responsibility

To ensure balanced learning and individual accountability, each team member will take complete ownership of specific features in the project, handling both frontend and backend tasks for each feature.

1. Login & Signup
 - Feature Description: This feature allows users to create an account (Signup) and log into the system (Login). It is crucial for user authentication and enabling personalized user experiences, such as order history and saved preferences.
 - Assigned Developer: Tam Minh Nguyen
 - Tech Involvement: Tam will handle both the frontend (React components, form handling, and validation) and backend (Node.js/Express API routes, JWT authentication) aspects of this feature.
2. Food Menu and Details
 - Feature Description: This feature displays a list of food items with brief descriptions, prices, and images. Users can click on an item to see detailed information, including ingredients, allergy information, and a button to add it to the cart. It is essential for presenting food options and enabling users to explore the menu.
 - Assigned Developer: Bach Nguyen
 - Tech Involvement: Bach will handle this feature's frontend (React components, food item display, and detail view) and backend (Node.js/Express routes for fetching food data from the database) aspects.
3. Cart Management
 - Feature Description: This feature enables users to add food items to their cart, adjust quantities, and proceed to checkout. It is vital to manage the user's order before finalizing the transaction.
 - Assigned Developer: Tam Minh Nguyen
 - Tech Involvement: Tam will handle this feature's frontend (React components for the cart, updating quantities, and displaying the cart) and backend (Node.js/Express routes for managing cart data in the session or database) aspects.
4. Checkout and Payment
 - Feature Description: This feature summarizes the items in the user's cart, collects delivery information, and processes payment via multiple methods (credit card, PayPal). It is a key part of ordering, allowing users to complete their purchases.
 - Assigned Developer: Bach Nguyen
 - Tech Involvement: Bach will handle both the frontend (React components for the checkout page, form validation, and payment integration) and the backend (Node.js/Express routes for order creation, payment processing, and storing order details) aspects of this feature.
5. Order Confirmation
 - Feature Description: After completing the checkout, users are shown an order confirmation page that includes a summary of the items ordered, total price, and estimated delivery time. This feature reassures users that their order has been successfully placed.
 - Assigned Developer: Tam Minh Nguyen
 - Tech Involvement: Tam will handle this feature's frontend (React components to display the order details and confirmation message) and backend (Node.js/Express routes to fetch and send the order details after payment) aspects.
6. Review and Rating System
 - Feature Description: This feature allows users to leave reviews and rate food items after they've ordered them. It helps improve the platform's interactivity and gives users a way to share their experiences with others.
 - Assigned Developer: Bach Nguyen
 - Tech Involvement: Bach will handle this feature's frontend (React components for the review submission form and displaying reviews) and backend (Node.js/Express routes to store and retrieve reviews in MongoDB) aspects.

7. Resources and tools

For this project, we will use the following technologies and tools:

1. Frontend:
 - React (JSX): For building the user interface.
 - React Router: For navigation between pages.
 - Axios: For making API requests to the backend.
2. Backend:
 - Node.js: For building the server and API endpoints.
 - JWT: For user authentication.
3. Database:
 - MongoDB: For storing food data, user info, and orders.

Time Commitments and Responsibilities:

1. Tam Minh Nguyen:
 - Frontend development with React, React Router, and Axios.
 - Design of Checkout and Order Confirmation pages and wireframes.
2. Bach Nguyen:
 - Backend development with Node.js, Express, and MongoDB.
 - User authentication with JWT and API development.

Weekly Commitment:

1. Each team member will dedicate 10-15 hours per week for frontend, backend, and collaboration.
2. Each week, we will have a team meeting to address problems when developing apps.

8. Data Sources and Management

The data for our Vietnamese food ordering platform will come from multiple sources, including:

1. User Input: Users will provide data such as login credentials, order details such as selected dishes, delivery address, and reviews.
2. Admin Input: Admins will manage food data by adding, editing, or removing dishes from the menu.
3. Database: All data, including food details, user information, orders, and reviews, will be stored in MongoDB. This ensures scalable and persistent storage of data.

Sample Data Formats

1. Food Item:

```
1  {
2    "id": "12345",
3    "name": "Pho",
4    "description": "Vietnamese noodle soup",
5    "price": 12.99,
6    "ingredients": ["beef", "noodles", "herbs", "broth"],
7    "allergies": ["gluten"],
8    "vegan": false
9  }
10
```

2. User:

```
1  {
2    "userId": "1",
3    "name": "Bach Nguyen",
4    "email": "bachnguyen@example.com",
5    "password": "hashedpassword",
6    "orderHistory": ["12345", "67890"]
7  }
8
```

3. Order:

```
1  {
2    "orderId": "54321",
3    "userId": "1",
4    "foodItems": [
5      { "foodId": "12345", "quantity": 2 },
6      { "foodId": "67890", "quantity": 1 }
7    ],
8    "totalPrice": 38.97,
9    "status": "processing",
10   "deliveryAddress": "123 Main St, City, State"
11 }
```

4. Review:

```
1  {  
2    "reviewId": "98765",  
3    "userId": "1",  
4    "foodId": "12345",  
5    "rating": 4,  
6    "comment": "Delicious and hearty!"  
7  }  
8
```

9. File Structure and Project Organization:

This is the structure of the project:

├── frontend/	# Contains all React-related files
│ ├── src/	# Source files for the frontend application
│ │ ├── assets/	# Static assets like images, styles, etc.
│ │ ├── components/	# React components for different parts of the app
│ │ ├── App.jsx	# Main React component that contains routing
│ │ ├── main.jsx	# Entry point of the React application
│ │ ├── App.css	# Global styles for the application
│ │ └── index.js	# React DOM rendering
├── backend/	# Contains Express server, API routes, and database logic
│ ├── server.js	# Main Express server setup
│ ├── routes/	# API routes for handling user requests
│ │ ├── foodRoutes.js	# Routes for managing food items
│ │ ├── userRoutes.js	# Routes for user authentication and management
│ │ └── orderRoutes.js	# Routes for handling orders
│ ├── models/	# Database models for MongoDB collections
│ │ ├── Food.js	# Schema for food items
│ │ ├── User.js	# Schema for users
│ │ └── Order.js	# Schema for orders
│ ├── controllers/	# Business logic for handling requests
│ │ ├── foodController.js	# Functions for handling food CRUD operations
│ │ ├── userController.js	# Functions for handling user registration/login
│ │ └── orderController.js	# Functions for handling orders and payment
│ └── config/	# Configuration files
│ ├── db.js	# MongoDB database connection
│ └── auth.js	# JWT authentication middleware
├── Documents/	# Contains planning sketches, architecture, final report, etc.
│ ├── wireframes/	# Sketches and wireframes of the UI
│ ├── software-architecture.md	# Document describing the software architecture
│ └── final-report.pdf	# Final report and project summary
└── README.md	# Project documentation

Frontend and Backend Communication (RESTful APIs):

1. Frontend (React):
 - The frontend will make GET, POST, PUT, and DELETE requests to the backend using Axios or fetch API.
 - When a user logs in, the frontend sends a POST request to the backend's authentication route (**/api/auth/login**) with the user credentials.
 - The frontend will display food items by sending a GET request to the backend's food route (**/api/food**) and rendering the response (food list) on the UI.
2. Backend (Node.js):
 - The backend will handle these requests using routes defined in the Express server. Each route will be responsible for handling specific actions.
 - **POST /api/auth/login**: Accepts user login credentials and returns a JWT token upon successful authentication.
 - **GET /api/food**: Fetches a list of all food items from the database and returns it to the frontend.
 - **POST /api/order**: Creates a new order based on user selections and stores it in the database.

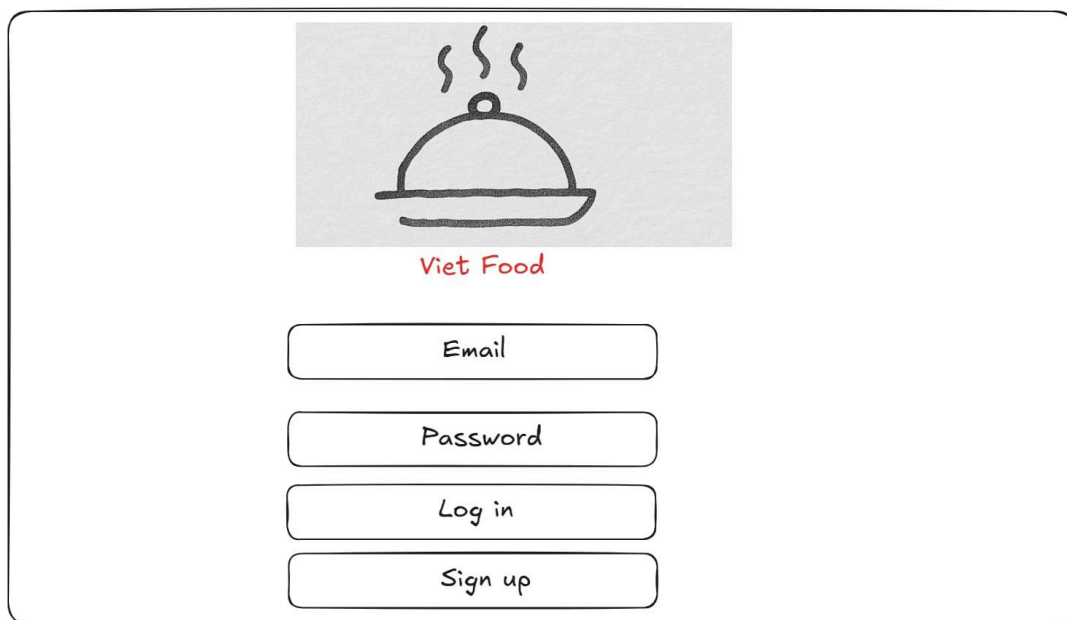
3. Database (MongoDB):

- The backend will interact with the MongoDB database using Mongoose models. The food data will be stored in the Food model, user information in the User model, and order details in the Order model.
- When the backend receives a request to fetch food items, it queries the MongoDB database and returns the data in a JSON format to the frontend.

By using RESTful APIs and JSON responses, the frontend and backend will work in tandem to provide a smooth and interactive user experience, ensuring that data is consistently synced between the client and server.

10. User Experience Views

Login View: The Login page presents users with a simple form to enter their email and password, along with a "Login" button. Users can also navigate to the "Signup" page if they don't have an account. This page serves as the entry point for users to access their accounts and proceed with placing orders or reviewing their order history.



The image shows a login form within a rounded rectangular container. At the top center is a logo consisting of a hand-drawn illustration of a steaming bowl of food, with three wavy lines above it representing steam. Below the illustration, the text "Viet Food" is written in a red, slightly irregular font. Underneath the logo, there are four rounded rectangular input fields stacked vertically. The first field is labeled "Email", the second "Password", the third "Log in", and the fourth "Sign up". Each label is centered within its respective field.

Signup View: The Signup page displays a registration form where users can enter their name, email, and password. After filling out the form, users can click "Signup" to create their account or navigate to the "Login" page if they already have one. This page is for new users to create an account before logging in.



Viet Food

Sign up

Name

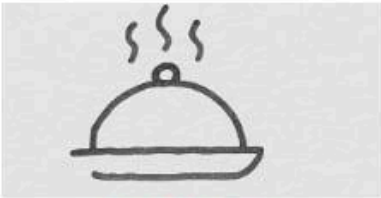
Email

Password

Sign up

Log in

View for Displaying All Food: The Food Display page shows a list of food items with images, names, and prices. Users can filter dishes based on categories like region or ingredients and click on any item to view more details. This page serves as the main menu, allowing users to explore available dishes and choose items to add to their cart.




Food Display

Food img	Name Description	Price
----------	---------------------	-------

Food img	Name Description	Price
----------	---------------------	-------

View for Specific Food (Description, Allergy, Price, Add to Cart): The specific food page provides detailed information about a dish, including its ingredients, price, allergy info, and vegan status. Users can review this information and add the item to their cart. After this, they can return to the food list or

checkout.

Fooding
Price

Description: _____

Allergy info: _____
Vegan: Yes or No

Add to cart

Checkout Page: The Checkout page summarizes the user's cart, displaying selected items, quantities, and the total price. Users can enter their delivery address, select a payment method, and review their order before submitting. This page is where users finalize their order details before proceeding to payment.

Order Summary

Spring rolls2\$8.00

Pho2\$8.00

Total\$20.99

Payment Method

Card number

Expiration date

Security code

Name on card

Check out

Delivery Address

Address

Apartment, suite

City

State

Phone

ZIP

Order Confirmation Page: The Order Confirmation page thanks the user for their purchase and provides an order summary, including food items, total cost, and delivery details. Users can view their order again or return to the homepage to continue browsing. This page confirms that the order has been successfully placed and provides the final details.

ORDER CONFIRMATION PAGE

Thank you for your purchase!

Order Summary

Spring rolls	2	\$8.00
Pho	2	\$8.00

Total

\$20.99

Delivery To:

123 Main St
Apt 4B
Anytown, CA 12345

View Order

Continue Browsing

11. Final Comments

This project is an exciting opportunity for us to apply the web development skills we've learned throughout our coursework, specifically in creating a fully functional online food ordering platform. Through this experience, we aim to improve our proficiency in React, Node.js, and MongoDB and gain hands-on experience in implementing user authentication, payment systems, and CRUD operations.

By the end of the project, we want to understand how to integrate the frontend and backend, manage databases, and optimize user experience. Additionally, we are eager to learn more about best practices in full-stack development and how to collaborate on a team project effectively.

You can reach out to us at:

1. Tam Minh Nguyen(tamminh@iastate.edu)
2. Bach Nguyen(ntbach@iastate.edu)