**INDIVIDUAL CONTRIBUTIONS**

**Kayis – User Registration & Login Module**

Kayis was responsible for designing and implementing the secure user authentication system for the application. He built the registration and login modules using Spring Boot and Spring Security, incorporating JWT-based token authentication to ensure secure access to protected resources. The registration form he developed included robust validations, such as strong password enforcement and email format verification. Kayis also implemented role-based access control, assigning the default role of USER upon registration, with potential for future expansion to include admin or premium roles. His configuration of CORS settings and integration of password encryption using BCrypt ensured both security and flexibility for frontend integration. Furthermore, he built middleware to intercept incoming requests and validate JWT tokens, allowing only authenticated users to access sensitive endpoints such as diet generation and preference submission. His expertise in security architecture helped establish the foundational authentication system of the platform, ensuring all modules operated securely under a unified access control framework. Beyond authentication, Kayis also designed key backend components to support user persistence using PostgreSQL, handling exceptions like duplicate registrations and invalid credentials gracefully. He worked closely with other team members to integrate user authentication into the overall system flow, ensuring smooth transitions between login, registration, and personalised features. Overall, his contribution provided the backbone of user identity and access security, enabling the app to protect user data and customise features based on the authenticated user’s profile. His meticulous work on the security layer allowed the rest of the system to function reliably and securely.

**Barath – Diet Generator & AI Integration**

Barath played a central role in the intelligent core of the application—the diet generator module. He was responsible for connecting user input with OpenAI’s GPT model to dynamically generate personalised weekly meal plans. Using Java and the OpenAI Java SDK, Barath crafted structured prompts based on user goals, dietary preferences, allergies, and food restrictions. He handled the transformation of backend data into natural language prompts and processed the AI’s response into structured JSON, suitable for frontend rendering. This required close attention to API call formatting, handling edge cases such as missing inputs, and ensuring graceful degradation in case of network issues or API rate limits. In addition to the AI integration, Barath developed the Angular component to display the generated diet plan using responsive Material Design cards. Each card included detailed nutritional information such as calories, proteins, fats, and carbs, which were formatted for a better user experience. He implemented utility services to highlight macronutrients and provided prep time estimates to match the user’s meal preparation preferences. The logic Barath implemented allowed for a smooth pipeline between user input and AI output, making the app not just functional but intelligent and personalised. He also added caching mechanisms to avoid redundant API calls and fine-tuned the prompt construction to increase the accuracy and usability of the AI’s diet suggestions. His work formed the heart of the application’s value proposition, blending technology and nutrition science through AI.

**Kani – Meal Preference Form with Multi-Accordion UI**

Kani took charge of building the user-facing meal preference form, one of the most critical touchpoints of the application. She implemented an intuitive, responsive form using Angular and Angular Material that allowed users to enter detailed dietary preferences. The form was split into multiple sections using accordion panels for a better user experience and clarity. Each panel covered a category such as personal details, food allergies, preferred cuisines, and foods to include or exclude. Kani integrated Angular Reactive Forms with dynamic FormArray controls to allow users to input multiple food items or restrictions, with chip-list support for intuitive interaction. She designed dropdowns, sliders, and multi-select components to capture structured data like meal count, activity level, meal prep time, and budget preferences. In addition to the frontend development, she implemented form validations to prevent incomplete or invalid data entries and ensured that data was accurately submitted to the backend via secure HTTP requests. CRUD functionality was fully supported—users could submit, review, and delete their preferences. Kani also coordinated closely with the backend team to map form inputs to a domain model that aligned with the AI prompt logic. She tested the form’s responsiveness across devices and optimised its usability for students and working professionals alike. Her thoughtful UI design made it simple and engaging for users to articulate their health and nutrition goals, which directly impacted the accuracy and personalisation of the generated diet plans. Kani’s contribution provided a polished and user-centric interface, turning complex user inputs into a seamless experience.

**Aakash – Cloud Deployment, Testing & Hosting**

Aakash was instrumental in bringing the entire system live by managing cloud deployment, backend hosting, and infrastructure setup. He deployed the full-stack application—consisting of a Spring Boot backend and Angular frontend—on a Microsoft Azure Virtual Machine, ensuring scalability and reliability. Aakash configured the PostgreSQL database on the same Azure VM and implemented secure environment handling using env files to store credentials and API keys. He installed and configured Nginx as a reverse proxy to handle HTTPS redirection and SSL certificate management using Let’s Encrypt and Certbot. With this setup, he enabled users to securely access the site over HTTPS, protecting user data during transmission. Aakash also handled the deployment of the Angular build by serving static files through Nginx. His deployment scripts ensured that system services like the Spring Boot application restarted automatically on server reboot using systemd. In addition to deployment, he conducted API and behavioral testing using Postman and other tools to verify endpoint responses, authentication flows, and database interactions. He wrote test cases simulating both valid and invalid scenarios, including login failures, API misuse, and preference submission errors. These tests ensured that the system was robust under various user conditions. Aakash also managed CI/CD scripts for updating code deployments quickly and safely. His work ensured the project was not only accessible but also production-ready, fast, and secure. Through his expertise in DevOps and cloud infrastructure, Aakash ensured that the team’s collective effort reached end users without disruption or compromise in performance.