Webgencia

Technical Specifications

Project	Menu Maker by Qwenta

Version	**Author**	**Date**	**Approval**
1.0	Rand Rahim	Jan 22, 2025	John - Qwenta's project manager Soufiane - product owner

The purpose of this document is to define and justify the technical specifications for "Menu Maker by Qwenta" tool.

I. Technology Decisions

- Overview of the functional requirements and their technical solutions:

Requirement	**Constraints**	**Solution**	**Description of the Solution**	**Supporting Arguments (2)**
Menu category creation	Must allow adding categories dynamically from a modal window	React.js Modal	React.js library will enable dynamic, responsive modal windows with minimal coding.	1) React.js aligns with the chosen front-end framework for the project.

Webgencia

User Authentication	Secure user login and registration processes	bcrypt, JWT, Token	User authentication will be managed using bcrypt for hashing passwords, JWT for token generation, and Token for session management.	1) Offers robust security and scalability for authentication.
Backend API	RESTful API to manage menu categories and items	Node.js & Express	Node.js allows for efficient handling of API calls; Express simplifies routing and middleware usage.	1) Node.js is a scalable and non-blocking solution.
Database	Store user and menu data efficiently	MongoDB	A NoSQL database like MongoDB provides scalability and flexibility for handling hierarchical data.	1) MongoDB's schemaless nature fits dynamic menu requirements.
PDF Export Feature	Allow menus to be exported in PDF format	jsPDF Library	jsPDF will be used to generate high-quality, printable PDFs of user-created menus.	1) Supports customization and styling options for seamless integration.
Image Upload for Dishes	Enable users to upload images for dishes	Multer & Sharp	Multer will handle file uploads, while Sharp will optimize and process images.	1) Provides efficient and secure handling of image files.

Webgencia

Font and Color Customization	Allow users to customize fonts and colors for their menus	CSS & SCSS	CSS and SCSS will enable efficient styling and dynamic updates for user customization.	1) Provides flexibility and integrates seamlessly with React.js.
Develop Pages	Develop landing, login, and dashboard pages	React.js	React.js will enable the creation of dynamic, component-based pages that are easy to maintain.	1) Provides reusability and modularity for large-scale web applications.

II. Links With the Backend

- Which server language? *Node.js*

- Is an API needed? If yes, which one? Yes
- RESTful API for managing user accounts, menu categories, and items.
- Endpoints for handling PDF generation and image uploads.

- Selected database: *e.g.: NoSQL – MongoDB*
  It allows for easy handling of large volumes of unstructured or semi-structured data, which is ideal for dynamic web applications.

Webgencia

III. Domain and Hosting Provider Recommendations

- Domain name: menu-maker-by-qwenta.com
- Hosting provider name
  Frontend: Netlify (ideal for React applications).
  Backend and Database: Google Cloud Platform (GCP) or Microsoft Azure.
  MongoDB: MongoDB Atlas (optimized for NoSQL).
- Email address:
  contact@qwenta-menu.com
  support@qwenta-menu.com

IV. Accessibility

- Browser compatibility: The site will be compatible with the following browsers:
  - Google Chrome, Firefox, Safari.
- Device types: The project will be developed to be accessible on:
  - Desktop, Tablet, Mobile.

V. Third-Party Services

Service	Supporting Arguments
bcrypt	Ensures secure password hashing.
JWT	Provides scalable and secure token-based authentication.
Multer & Sharp	Handles and processes image uploads effectively and efficiently.
jsPDF Library	Allows seamless PDF generation for user-created menus.
Google Analytics	Provides detailed insights into user behavior and site performance.
Paypal	If we need to handle online payments for our app's payment features, services like PayPal can make it easier to integrate payments.

Webgencia

VI. Security Recommendations

- **Account Access:** Ensure secure password storage using bcrypt for hashing.
- **Authentication:** Implement token-based authentication using JWT for secure and scalable user sessions.
- **Encryption:** Use HTTPS for all data transmission.
- **Image Uploads:** Validate and sanitize all uploaded files using Multer and Sharp to prevent malicious content.
- **Latest Versions:** Regularly check the Node.js dependency versions to avoid known vulnerabilities.
- **Plugins & Libraries:** Install only the required ones and keep them up to date.

VII. Website Maintenance and Future Updates

- Regular updates to ensure compatibility with the latest browsers and devices.
- Monthly security audits to identify and resolve vulnerabilities.
- Scalability enhancements as the user base grows.
- Regular database backups and performance optimizations.
- Dedicated support for resolving bugs and implementing new features.
- Monitoring and updating third-party libraries to prevent outdated dependencies.
- Addition of new features such as menu sharing and advanced analytics as per client feedback.

Webgencia

VIII. Branding and UI Design

- The project will use these colors for Branding and UI Design.
    - Green: #8BC7B1
    - Beige: #FFF4E8
    - Black: #000
    - White: #FFF
    - Brown: #C5A073

IX. Additional Notes

- The project will follow Agile development principles, allowing iterative improvements based on client feedback.
- Initial deployment is planned for Q2 2025, with a beta testing phase in Q1 2025.
- Integration with social platforms like Instagram for menu sharing is under consideration.