



**Computer Science Department**  
**Faculty of Computer and Information Technology**  
**Sana'a University**

### **Frappe best practices**

Frappe is an open-source framework for building web applications in Python, primarily known for its use in ERPNext. Here are some best practices to consider when working with Frappe:

1. Follow the Frappe Framework Conventions: Frappe has its own set of conventions and patterns, known as the "Frappe Way." Adhering to these conventions ensures consistency and maintainability in your code. This includes naming conventions, folder structure, file organization, and code formatting. Familiarize yourself with these conventions and follow them throughout your development process.
2. Leverage Frappe's Built-in Features: Frappe provides a rich set of built-in features and functionalities that can save development time and effort. Explore and utilize these features whenever possible instead of reinventing the wheel. For example, Frappe offers user authentication, role-based permissions, email integration, RESTful APIs, and more. Understanding and leveraging these features will enhance your development process.
3. Use Frappe's ORM (Object-Relational Mapping): Frappe's ORM, called "DocTypes," provides an abstraction layer for interacting with the database. Utilize the ORM to define data models and perform database operations such as creating, retrieving, updating, and deleting records. It simplifies database management and ensures data integrity. Follow best practices for defining DocTypes, including proper field types, validation rules, and relationships.
4. Write Custom Apps: Frappe encourages building custom apps to extend its functionality. Instead of modifying the core code directly, create separate apps to add custom features or modify existing behavior. This approach allows for easier upgrades and maintenance. Utilize Frappe's hooks and events to integrate your custom apps seamlessly into the framework.



**Computer Science Department**  
**Faculty of Computer and Information Technology**  
**Sana'a University**

5. **Test Your Code:** Writing tests is crucial to ensure the reliability and stability of your Frappe applications. Frappe provides testing utilities and frameworks to facilitate unit testing and functional testing. Write test cases to cover different aspects of your codebase and aim for good test coverage. Regularly run your tests to catch potential bugs early and maintain the quality of your code.

6. **Optimize Database Operations:** Frappe uses a document-based database model called "MariaDB/MySQL." To optimize database operations, follow best practices such as using appropriate indexes, avoiding excessive database queries, and utilizing Frappe's caching mechanisms. Be mindful of database performance and query efficiency to ensure a smooth user experience.

7. **Secure Your Application:** Security is crucial for any web application. Follow security best practices such as protecting against common vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Implement proper authentication and authorization mechanisms using Frappe's built-in features. Regularly patch and update your Frappe installation to address any security vulnerabilities.

8. **Version Control and Deployment:** Use a version control system like Git to manage your Frappe projects effectively. Keep your codebase under version control and follow best practices for branching, merging, and collaboration. Implement a robust deployment process to ensure smooth updates and rollbacks. Use deployment tools like Docker or Ansible to automate the deployment process and maintain consistency across different environments.

9. **Engage with the Frappe Community:** Frappe has an active and helpful community of developers. Engage with the community through forums, mailing lists, chat groups, and GitHub. Seek help, ask questions, and share your knowledge and experiences. Participating in the community will keep you updated on the latest developments, best practices, and solutions to common challenges.