How do you find an IT career path that suits your interests and skills?

Authors:

- ENG. Adel Nasim
- ENG. Mohammed Khamis

TOC

- **Am I a developer?**
  - What is application development and what are its fields?
  - Do I like to be a developer?
- **What developer might I be?**
  - Prerequisites
  - What is Front End Development?
  - What is Back End Development?
  - What is the DevOps?
- **Hot development fields**
  - AI
  - Blockchain
- **Who are my colleagues?**
  - What do QA and QC involve?
- **What are my challenges?**
- **Production challenges?**
- **An idea for an application**

Paper's Objectives:

- Discovering the right path for you, based on your skills and inclinations.

Am I a developer?

Q1) What is application development and what are its fields?

- Application development refers to the process of creating software applications for various platforms, such as mobile devices, desktop computers, web browsers, and more.

- This process involves designing, coding, testing, and debugging applications to meet specific user needs or solve particular problems. Application development is a broad field that encompasses various specialized areas and technologies. Here are some key fields within application development:

Application Development fields:

- **Mobile App Development:**
  - **iOS Development**: typically using languages like Swift or Objective-C.
  - **Android Development**: using languages such as Java or Kotlin.

- **Web Development:**
  - **Front-End Development:** Focusing on the user interface and user experience of web applications. It involves technologies like HTML, CSS, and JavaScript.
  - **Back-End Development:** Involves server-side programming and managing databases. Common back-end languages include Python, Ruby, PHP, and Node.js.
  - **Full-Stack Development**: Combining both front-end and back-end development skills to work on the entire web application stack.

Application Development fields:

- **Game Development**:
  - **2D/3D Game Development**: Creating video games for various platforms using game development engines like Unity or Unreal Engine.
  - **Augmented Reality (AR) and Virtual Reality (VR) Development**: Building applications that incorporate AR or VR technologies.
- **DevOps (Development and Operations):**
  - Involves practices and tools for automating and improving the process of software development, including continuous integration, continuous delivery, and infrastructure as code.

Application Development fields:

- **Enterprise Application Development:** Building large-scale, complex applications for businesses to streamline processes, manage data, and improve overall efficiency.
- **Cloud Computing:** Developing applications that leverage cloud services and platforms, such as AWS, Azure, or Google Cloud.
- **Internet of Things (IoT) Development:** Creating applications that interact with and control IoT devices, collecting and analyzing data from connected devices.

Application Development fields:

- **Desktop Application Development:**
  - **Windows Application Development:** Creating software for the Windows operating system using languages such as C# or C++.
  - **Mac Application Development**: Developing applications for macOS using languages like Swift or Objective-C.
  - **Linux Application Development:** Building applications that run on Linux-based systems using languages like C or Python.
- **Cross-Platform Development**: Using frameworks like React Native, Flutter, or Xamarin to develop applications that can run on multiple platforms with a single codebase.

Do I like to be a developer?

Some key criteria that might influence your experience and satisfaction as a developer:

- **Interest in Problem-Solving.**
- **Curiosity and Learning.**
- **Attention to Detail.**
- **Communication Skills.**
- **Passion for Technology.**

Do I like to be a developer?

Some key criteria that might influence your experience and satisfaction as a developer:

- **Autonomy and Flexibility:** Some developers thrive in environments that offer a degree of autonomy and flexibility in their work.
- **Passion for Continuous Improvement:** Software development is an ever-evolving field, and the best developers often have a passion for continuous improvement. If you enjoy refining your skills and staying up-to-date with industry trends, you might find the work more rewarding.
- **Job Market and Opportunities:** The demand for developers can vary depending on the region and the specific skills in demand. Assessing job opportunities and market conditions may influence your decision to pursue a career in development.

What developer might I be?

- **Prerequisites**
- **What is Front End Development?**
- **What is Back End Development?**
- **What is the DevOps?**

Prerequisites:

- **Core Programming Languages:**
  Learn C++ and Java in-depth, covering all fundamental concepts such as variables, loops, functions, classes, and exception handling.

- **Problem-Solving and Competitive Programming:**
  Practice problem-solving on platforms like HackerRank, Codeforces, LeetCode, and TopCoder. This will enhance your algorithmic and coding skills.

- **Object-Oriented Programming (OOP):**
  Understand and master the principles of OOP. Apply them in projects to solidify your understanding.

Prerequisites:

- **Data Structures:**
  Learn and implement various data structures such as arrays, linked lists, stacks, queues, trees, and graphs. Understand the time and space complexities of operations on these structures.

- **Algorithms:**
  Study and implement common algorithms like searching, sorting, dynamic programming, and graph algorithms. Understand their time and space complexities.

- **Design Patterns:**
  Familiarize yourself with common design patterns like Singleton, Factory, Observer, and MVC. Learn how and when to apply them to solve design problems.

- **Database Management Systems (DBMS)**

What is Front End development?

- Front-end web development is the process of designing and building the graphical user interface (GUI) of any website.
- Sometimes called client-side development, front-end development refers to the development of the parts of a website the user interacts with, such as buttons and text boxes. We usually perform front-end development using HTML, CSS, and JavaScript.
- The front end of a website is everything the user either sees or interacts with when they visit the website. It is responsible for the total look and feel of an online experience.

Front End developer Roadmap:

- **Basic Web Technologies:**
  - HTML: Learn the fundamentals of HTML for structuring web content.
  - CSS: Master CSS for styling and layout of web pages.
  - JavaScript + TypeScript : Understand the basics of JavaScript for adding interactivity to web pages.

- **Advanced JavaScript:**
  - DOM Manipulation: Learn to manipulate the Document Object Model for dynamic web pages.
  - ES6+ Features: Familiarize yourself with the latest ECMAScript features.
  - Asynchronous JavaScript: Understand promises, async/await, and callbacks for handling asynchronous operations.

Front End developer Roadmap:

- **Frontend Frameworks:**
  - Choose a JavaScript framework/library:
  - React.js: A popular library for building user interfaces.
  - Angular: A comprehensive frontend framework.
  - Vue.js: A progressive JavaScript framework.
    How can we choose? (https://github.com/facebook/react)

- **Package Managers:**
  - Learn tools like npm or yarn to manage project dependencies.
    (https://www.npmjs.com/package/mongo-aggregation-builder)

- **Build Tools:**
  - Webpack: Understand how to bundle and optimize your code using webpack.
  - Babel: Use Babel to transpile modern JavaScript to browser-compatible code.

Front End developer Roadmap:

- **Testing:**
  - Unit Testing: Learn tools like Jest for testing JavaScript code.
  - End-to-End Testing: Explore tools like Cypress or Selenium for testing the entire application flow.

- **APIs:**
  - Understand how to make asynchronous requests to APIs using tools like Fetch or Axios.

- **State Management:**
  - Learn state management libraries/tools such as Redux (for React), Vuex (for Vue), or NgRx (for Angular).

Front End developer Roadmap:

- **Web Security:**
  - Understand and implement best practices for web security, including HTTPS, Content Security Policy (CSP), and secure coding practices.

- **Build and Deployment:**
  - Understand the deployment process and tools like Netlify, Vercel, or GitHub Pages.
  - Babel: Use Babel to transpile modern JavaScript to browser-compatible code.

- **Common Skills: Git & (GitHub, GitLab)**

What is Back End development?

- Back-end developers focus on the server side of websites. They use technical skills to perform the behind-the-scenes work that creates a website's structure and overall functionality, allowing a site's front end to exist. These professionals create a site's operations, databases, and application programming interface (API).

- Mixed between Front-end and back-end like offline mode

Back End developer Roadmap:

- **Programming Languages:**
  - Choose a backend programming language:
  - **Node.js (JavaScript/TypeScript):** Known for its asynchronous, event-driven architecture.
  - **Python:** Popular for its simplicity and readability.
  - **Java:** Widely used in enterprise environments.
  - **Ruby:** Known for its simplicity and productivity.
  - **Go (Golang):** Known for performance and scalability.

- **Web Frameworks:**
  - Learn a backend web framework for your chosen language:
  - **Node.js:** Express.js, Nest.js.
  - **Python:** Django, Flask.
  - **Java:** Spring Boot.
  - **Ruby:** Ruby on Rails.
  - **Go:** Gin, Echo.
-

Back End developer Roadmap:

- **Databases:**
  - Understand different types of databases:
  - **Relational Databases:** MySQL, PostgreSQL.
  - **NoSQL Databases:** MongoDB, Cassandra, Redis.
- **Database Query Language:**
  - Learn SQL for relational databases.
  - Understand how to interact with NoSQL databases.
- **API Design:**
  - Learn how to design RESTful APIs.
  - Understand API documentation and OpenAPI/Swagger.
- **Authentication and Authorization:**
  - Implement user authentication using techniques like JWT (JSON Web Tokens) or OAuth.
  - Understand and implement role-based access control.
-

Back End developer Roadmap:

- **Testing:**
  - Implement unit testing and integration testing.
  - Explore tools like JUnit, Mocha, or PyTest.
- **Logging and Monitoring:**
  - Implement logging in your applications.
  - Understand monitoring tools like Prometheus or Grafana.
- **Caching:**
  - Learn caching strategies using tools like Redis or Memcached.
- **Message Queues:**
  - Understand message queue systems like RabbitMQ or Apache Kafka for handling asynchronous tasks.
- **Security Best Practices:**
  - Implement secure coding practices.
  - Understand and mitigate common security vulnerabilities.

Back End developer Roadmap:

- **GraphQL:**
  - Familiarize yourself with GraphQL for efficient and flexible API querying.
- **Serverless Architecture:**
  - Explore serverless computing platforms like AWS Lambda or Azure Functions.
- **Build and Deployment:**
  - Understand the deployment process and tools like Jenkins, Travis CI, or GitLab CI.
- **Continuous Integration/Continuous Deployment (CI/CD):**
  - Learn how to set up automated build and deployment pipelines.
- **Web Services:**
  - Understand concepts like RESTful web services and SOAP.

Back End developer Roadmap:

- **Soft Skills:**
  - Develop good communication skills, collaborate effectively with team members, and stay updated with industry trends.
- **Documentation:**
  - Emphasize the importance of clear and concise documentation for your code and APIs.
- **Scalability:**
  - Understand principles of scalable architecture for handling growing user loads.
- **Reliability and Disaster Recovery:**
  - Learn strategies for ensuring system reliability and disaster recovery.
- **Microservices Architecture:**
  - Explore microservices architecture principles for building modular and scalable systems.

What is the DevOps?

- **DevOps (short for Development and Operations):** is a set of practices that aims to automate and improve the collaboration and communication between software development and IT operations teams.

- **The goal of DevOps:** is to shorten the development lifecycle, deliver high-quality software continuously, and foster a culture of collaboration and shared responsibility.

DevOps Roadmap:

- **Understand DevOps Principles:**
  - Familiarize yourself with the key principles of DevOps, such as collaboration, automation, continuous integration, continuous delivery, and continuous monitoring.

- **Learn Version Control:**
  - Master version control systems like Git for tracking changes in code.

- **Scripting and Automation:**
  - Learn scripting languages like Bash, PowerShell, or Python for automating repetitive tasks.

DevOps Roadmap:

- **Infrastructure as Code (IaC):**
  - Understand the concept of treating infrastructure as code, and learn tools like Terraform or Ansible for automating infrastructure provisioning.

- **Continuous Integration (CI):**
  - Learn CI concepts and tools like Jenkins, GitLab CI, or Travis CI for automating the integration of code changes.

- **Continuous Deployment (CD):**
  - Understand CD practices to automate the deployment of applications. Tools like Jenkins, GitLab CI, or Spinnaker can be used for this purpose.

DevOps Roadmap:

- **Configuration Management:**
  - Learn tools like Puppet, Chef, or Ansible for configuring and managing servers.

- **Containerization:**
  - Understand containerization using Docker to package applications and their dependencies.

- **Container Orchestration:**
  - Learn container orchestration tools like Kubernetes for automating the deployment, scaling, and management of containerized applications.

DevOps Roadmap:

- **Cloud Platforms:**
  - Familiarize yourself with cloud platforms like AWS, Azure, or Google Cloud for hosting and deploying applications.

- **Monitoring and Logging:**
  - Learn about monitoring tools like Prometheus, Grafana, and logging tools like ELK Stack (Elasticsearch, Logstash, Kibana) for tracking application performance and troubleshooting.

- **Collaboration and Communication:**
  - Emphasize collaboration and communication skills within development and operations teams.

DevOps Roadmap:

- **Security Practices:**
  - Understand security best practices for both application and infrastructure security.
- **Microservices Architecture:**
  - Explore the principles of microservices architecture for building scalable and modular applications.
- **Collaboration Tools:**
  - Use collaboration tools like Slack, Microsoft Teams, or others to facilitate communication and collaboration.
- **Knowledge of Agile Practices:**
  - Understand Agile methodologies and how DevOps aligns with Agile principles.

Hot development fields

- **Artificial Intelligence (AI):**
  - Understanding AI.
  - What is a model.
  - What is model training.
  - When to use AI.
  - Fine tuning ?

Problem : You have a business requirement to analyze the images of each of your clients to determine if the image includes a particular product or more. Each client has a different set of products.

Hot development fields

- **Object Detection models**
- **AWS Recognition**
- **Standard model / Custom labels**

- **How to use AI :**
  - Own developed models.
  - Managed pretrained models.
  - Packaged pretrained models.
  - Deploy public models.

Hot development fields

- **Block Chain:**
  - **What is Web3? The Decentralized Internet.**
  - **Smart contracts.**
  - **What are Tokens.**
  - **Am I a blockChain developer?**

Who are my colleagues?

- **UI / UX Designer**
- **Quality Assurance (QA)**
- **Quality Control (QC)**
- **Product Manager**

What are my challenges?

Who is my boss?

Learn, document, respect,appreciate,adapt, ask, ask again, request feedback, be patient.

University habit of proof of knowledge versus excellence delivery.

Production challenges.

What might go wrong will go wrong?

There can be more.design to scale

Availability (High Availability)

An Idea of an application

- Core Knowledge with experience
- Verified by others /stakeholders
- Comprehension of supportive tasks.
- Goal /What gap am I filling.
- Business model with financial analysis.
- Time plan with milestones that have measurable goals.
- continuous Evaluation.
- Be aware of operations.

References

- https://roadmap.sh
- https://remix.ethereum.org
- https://yolov8.com/
- https://us-east-1.console.aws.amazon.com/rekognition
- https://explorer.helium.com/
- https://www.blockchain-council.org/ethereum/erc20-vs-erc721/

Thank You . . .