Significance Of Software Architecture

In this lesson, we'll cover the significance of web application & software architecture & the reasoning behind learning it.

We'll cover the following

Importance Of Getting The Application Architecture Right

The key element in successfully creating anything is getting the base right. Now whether it is constructing a building or making a pizza. If we don't get the base right, we have to start over. Yeah!!.. I know, but there is no other way around.

Building a web application is no different. The architecture is it's base & has to be carefully thought, to avoid any major design changes & code refactoring at a later point in time.

Speaking with experience, you don't want to delve into re-designing stuff. It eats up your time like a black hole. It has the potential to push your shipping date further down the calendar by months, if not longer. And I won't even bring up the wastage of engineering & financial resources which is caused due to this. No, I won't!!

It also depends on what stage of the development process we hit an impasse due to the hasty decisions taken during the initial design phases. So, before we even touch the code & get our hands dirty, we have to make the underlying architecture right.

A look at the architecture of our app should bring a smile to everyone's face.

Though software development is an iterative and evolutionary process, we don't always get things perfect at the first go. Still, this can't be an excuse for not doing our homework.

An Overview Of The Software Development Process

In the industry, architects, developers and product owners spend a lot of time studying & discussing business requirements. In software engineering jargon, this is known as the *Requirement Gathering & Analysis*.

Once we are done with the business requirements, we sit down & brainstorm the use cases which we have to implement. This involves figuring out the corner cases as early as possible & fitting the Lego blocks together.

If you're a fan of documentation, you might also want to write a *high-level design document*.

Now, we have an understanding of the business requirements, use cases, corner cases and all. It's time to start the research on picking the right technology stack to implement the use cases.

Proof Of Concept

After we pick the fitting tech stack, we start writing a *POC (Proof of Concept)*

Why a POC?

A *POC* helps us get a closer, more hands-on view of the technology & the basic use case implementation. We get an insight into the pros and cons of the tech, performance or other technical limitations if any.

It helps with the learning curve if we're working with completely new tech, also the non-technical people like the product owners, stakeholders have something concrete to play with & to base their further decisions on.

Now, this is only for an industry scale product. If you are a solo indie developer or a small group, you can always skip the *POC* part and start with the main code.

So, we showcase the *POC* to the stakeholders & if everyone is satisfied, we finally get down to creating the main repo & our first dev branch on *GitHub*, or any other similar code hosting service which the business prefers.

Phew!!

So, by now you would have realized how important it is to get the architecture right at the first time & the knowledge of web architecture to developers.

Course Design

Hmmm.... Alright, this being said. Now speaking of this course. It is divided into two parts. In the first, we will discuss the concepts & the architectural components involved in designing web applications.

We will get insights into different tiers of software applications, *monolithic* repos, *microservices*, *peer to peer* architecture & a lot more.

In the second part, we will go through some of the use cases of designing the architecture for applications which we use in our day to day lives & are well familiar with.

We will also understand how applications are designed from the bare bones. What is the thought process of picking the right technology stack for our use case & so forth?

So, without further ado. Let's get started.