

---

# Fullstack javascript nanodegree

Session 10

# Agenda

- Project Discussion
- AMA (Ask Me Anything)





# Project Discussion



AMA

# Questions

What's the (roadmap I should follow / Skills I should gain) that will help me landing my first job?

## Technical

- ❑ Problem solving (in whatever programming languages you prefer)
- ❑ Frontend framework (hands-on / small project) [React.js, Angular, Vue.js]
- ❑ Docker
- ❑ ORM
- ❑ Databases (Design/ERD, NoSQL/Mongo, Subqueries, Joins, Indexing)
- ❑ OOP
- ❑ SOLID Principles
- ❑ Design Patterns (Just a small background)
- ❑ Algorithms & Data Structure (Just a small background)



# Questions

What's the (roadmap I should follow / Skills I should gain) that will help me landing my first job?

## Non-Technical

- English
- Agile
- Jira
- Communication and Presentation skills.
- Take care of your (CV, Cover Letter, GitHub, LinkedIn)
- Follow the tech industry latest news, which you can find through field related (YouTube channels, twitter accounts, articles, LinkedIn hashtags)
- Attend tech events whenever you can. building connections and increasing your network can be your most powerful weapon.



# Questions

What's the (roadmap I should follow / Skills I should gain) that will help me landing my first job?

## Operational

- Start applying for jobs.
- Write about the wonderful things you did, whether alone or with your team.
- Enter as many interviews as you can, even if you're not interested in the job.
- It will take you some time until you figure out the perfect combination to put in your CV. (people will start responding to your applications).
- Don't worry, the more time you spend applying/interviewing the more knowledge and experience you get.  
and this will increase your chances in the next application.
- Don't be embarrassed to ask someone in your connection for openings in his/her company, or to help you with a recommendation (This is more than normal in the market).



# Questions

What Programming languages should I learn?

- It doesn't matter, It's just a technology.  
Learning the concepts and the basics will increase your flexibility to jump from one language to another.
- Think of it as a tool that fits this need. For any other need you might have to use something else.
- The most populate in Egypt market right now are (PHP/Laravel, C#/.NET) and then (JS/Node.js, Java/Spring).





# Questions

GitHub collaboration guidelines? Branch-per-person or branch-per-feature?

□ Branch Per Feature



# Questions

How do I know if the code needs refactoring? And How do I know If I refactored it enough?

- ❑ Go with the refactoring **only** if the development hassle with the current code on the long run, is bigger than the refactoring hassle.  
Or if there's a huge security concerns with the current codebase.
- ❑ Normally you won't have to take this decisions, as the process must involve someone expert to consult.
- ❑ There's always a room for improvement, So that alone should not be a reason to refactor your code.
- ❑ Following best practices will save you a lot of time regarding this.



**Your Feedback is Appreciated**



Remember that we are here to help you  
All you have to do is **ASK!**



---

Thanks for attending



