

BECOMING A .NET DEVELOPER USING NET CORE X.X WHILE YOU BUILD A SOCIAL NETWORK.

STARLING GERMOSEN

SGERMOSEN.COM @SGERMOSEN24

[HTTP://GITHUB.COM/SGERMOSEN](http://github.com/sgermosen)



INTRODUCTION

WHAT WE WILL SEE

- BUILDING USER-INTERFACES
- BUILDING BACK-END APIs
- OBJECT-ORIENTED PROGRAMING
- CLEAN ARQUITECTURE
- EF CORE
- LINQ
- AZURE SERVICES



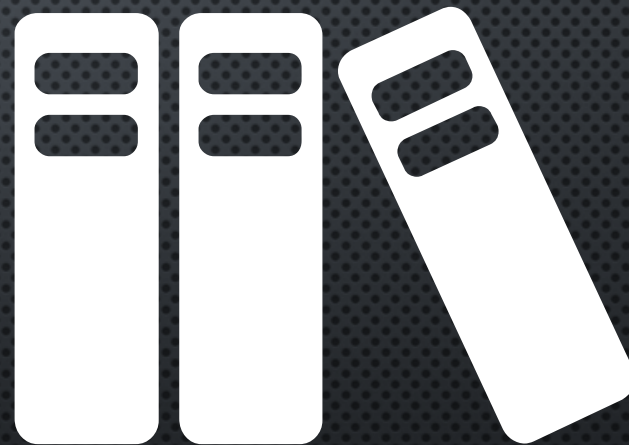
PREREQUISITES

- C#
- POO
- DATABASE MODELING
- VS COMMUNITY >= 2017
- .NETCORE >= 2.2



WHAT WE'RE GONNA LEARN

- BOOTSTRAP
- CSS
- USABILITY
- JAVASCRIPT
- EF CODE FIRST
- RESTFUL APIs
- SECURITY
- POCO
- OOD



STRUCTURE

- FUNDAMENTALS
- ADVANCE TOPICS
- ARCHITECTURE



FUNDAMENTALS

- WHERE THE F*CK SHOULD START
- WHAT'S NEXT
- REQUIREMENTS DOCUMENT
- SOFTWARE ENGINEERING MINDSET
- BACKEND
- FRONTEND
- DATA VALIDATIONS
- USABILITY
- ARTISTIC ASPECT OF UI



ADVANCE TOPICS

- NOTIFICATION SERVICE
- OBJECT-ORIENTED DESIGN
- BUILDING APIs
- BOOTSTRAP



ARCHITECTURE



LET'S START



WHAT THE H*LL IS DOT NET CORE



- MULTI PLATFORM
- RETRO COMPATIBLE
- SIZE
- PERFORMANCE
- NUGET
- SECURITY
- CLOUD ORIENTED
- DEPENDENCY INJECTION
- IIS OR APACHE
- OPEN SOURCE AND COMMUNITY ORIENTED

FILE NEW PROJECT

- RESHARPER
- WEB ESSENTIALS
- PRODUCTIVITY POWER TOOL



SUMMARY

- WE MET THE TEACHER.
- WE SAW THE KNOWLEDGE AND THE WAY OF WORK THAT WE ARE GOING TO IMPLEMENT.
- WE KNEW ABOUT .NET CORE, DIFFERENCES BETWEEN THEIR PREDECESOR, AND THE TEMPLATES.