



The “Large Project”

COP 4331

University of Central Florida

First...

- Congratulations on being done with the small project
 - Remember to turn in your PowerPoint
 - Your work has looked great
 - “Pretty good” has been the low bar
 - That's good
 - Especially if you're particularly proud of your work, keep it somewhere
 - I may ask for copies of some of your work to use as examples going forward
 - It's up to you whether you're okay with that!
- You can take down the website

Teams

- We'll be changing teams for the large project
- You'll get to choose your teams this time...
 - ...to an extent
 - Must have at least 6, may have up to 7
 - Team selection will open this weekend
- **Don't wait to get started**
- You need to do your part: Do What You Say You Will Do (DWYSWD)
 - Be respectful to team members
 - Even if you think you know everything, split the work

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - Remote API – Using Express
 - Web app – Using React
 - Runtime environment – Node.js on the server side
 - Mobile app – Cross-platform framework

Project Requirements and Recommendations

- In general, the project will have the following components:
 - **Remote database - MongoDB**
 - Remote API – Using Express
 - Web app – Using React
 - Runtime environment – Node.js on the server side
 - Mobile app – Cross-platform framework
- Stores data and information

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - **Remote API – Using Express**
 - Web app – Using React
 - Runtime environment – Node.js on the server side
 - Mobile app – Cross-platform framework
- Connects your user interfaces written via React to the underlying Node environment, and through Node to MongoDB

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - Remote API – Using Express
 - **Web app – Using React**
 - Runtime environment – Node.js on the server side
 - Mobile app – Cross-platform framework
- The framework for the user interface
- You'll probably also want a second-order style library like **Tailwind** or **Bootstrap**
- You **may not** use straight HTML/CSS
- I'll require you to use TypeScript where appropriate

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - Remote API – Using Express
 - Web app – Using React
 - **Runtime environment – Node.js on the server side**
 - Mobile app – Cross-platform framework
- The underlying interpreter that runs all the JavaScript code
- Includes a **very** large number of libraries
- A Node deployment can get **very** complex
- We will show you techniques to keep it manageable

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - Remote API – Using Express
 - Web app – Using React
 - Runtime environment – Node.js on the server side
 - **Mobile app – Cross-platform framework**
- Designed to run on phones natively
- Not just displaying the webpage on a phone's browser
- Generally, you will use the **Flutter** framework to make deploying to both iOS and Android (relatively) easy, and make it (relatively) easy to connect to the rest of the stack

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - MongoDB
 - Remote API – Using Express
 - Web app – Using React
 - Runtime environment – Node.js on the server side
 - Mobile app – Cross-platform framework

Project Requirements and Recommendations

- In general, the project will have the following components:
 - Remote database - **M**ongoDB
 - Remote API – Using **E**xpress
 - Web app – Using **R**ead
 - Runtime environment – **N**ode.js on the server side
 - Mobile app – Cross-platform framework

Documentation Requirements

- Gantt chart
- ERD
- Use Cases
- Sequence diagram or Activity diagram
- SwaggerHub, PostMan, or something like it for API
- Class diagram for the Mobile app
- Unit and Integration test results
- Prototypes / Wireframes – Suggest using Figma, Adobe XD or Sketch
 - Yes, these are required

Verification and Security

- When someone creates a new account, send an email to that person before they can login
- You'll also need password reset functionality
- Potential options: SendGrid, NodeMailer, service-native authentication
- We'll discuss best practices on security for the app proper as well
- (Hint: it's not going to involve open API calls like you've been doing)

Past Projects

- Menu programs
- Nutrition programs
- Exercise programs
- Teaching games
- Political programs
- Research-Oriented projects

Past Projects

- | | |
|------------------------------|--------|
| • Menu programs | No. |
| • Nutrition programs | Maybe. |
| • Exercise programs | Okay. |
| • Teaching games | Okay. |
| • Political programs | No. |
| • Research-Oriented projects | Okay. |