CREATIVE
TECHNOLOGIES

Interactive
Development

MEAN STACK
EXPRESS JS
203

T3 | Brief

9 July 2020

Lecturer
Ruan de Necker
ruand@openwindow.co.za

OPEN WINDOW

LET'S GET STARTED...

INTRODUCTION

The MEAN stack is a collection of four different technologies that work together to create powerful web applications using the JavaScript programming language on both the server and client side. The E in MEAN stands for Express.

Express is a framework that sits on top of node. Many common features of webapps, such as routing and static file serving, need to be implemented from scratch when using vanilla node. Express simplifies this process by providing a single, simple way to use these features without having to implement them manually.

WHAT YOU'LL LEARN:

DEVELOP USING EXPRESS JS

DEVELOP APIS

UNDERSTAND EXPRESS JS ROUTING AND MIDDLEWARE

MANAGE THE SCOPE OF A PROJECT

TERM 3 OUTLINE...

THE BRIEF

designed by freepik

PROJECT...

Create an Express server for an online high school scheduling system.

BACKGROUND AND CONTEXT...

You have been hired to create a scheduling system for a high school. Teachers and learners should be able to log in to see the class schedule and read class details.

TERM 3 OUTLINE...

PROJECT BREAKDOWN

WHAT YOU WILL DO...

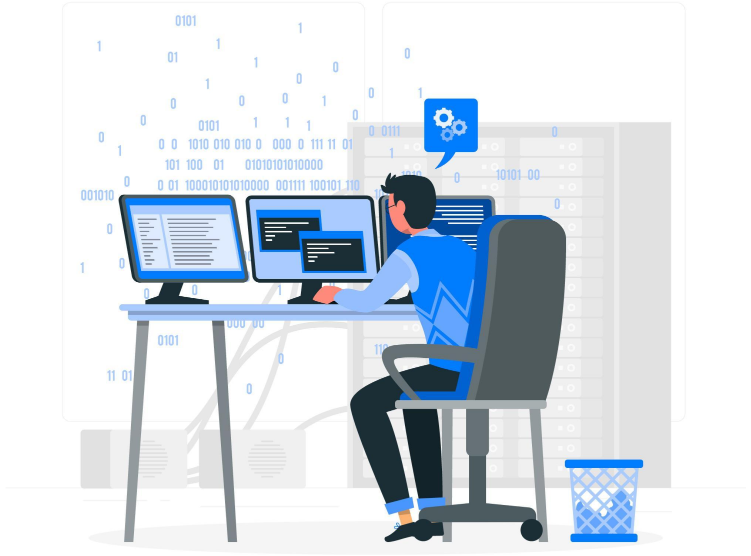
You should create an express server with an authentication system for users. Depending on whether the user is a student or a teacher, they will see different information. Requests to the server should be logged with middleware.

The tasks highlighted in **blue** below must be completed for the **midterm submission milestone** (week 4). The tasks highlighted in **green** below must be completed for the **final submission milestone** (week 7).

REQUIREMENTS...

The server should expose a JSON API, which returns the following information:

1. A list of all the classes
2. Details of a particular class, including:
    a. The teacher of the class
    b. The students in the class
    c. The time of the class (Day and period)
    d. The classroom number
3. School of Interaction Arts Project Brief
4. The subject
5. A list of classes taught by a particular teacher
6. A list of classes taken by a particular learner
7. A user id when a valid email and password are supplied



PROJECT BREAKDOWN CONTINUED...

REQUIREMENTS...

The server should return a web page from the root route that consumes the API with Angular to display the timetable information. The front end should include:

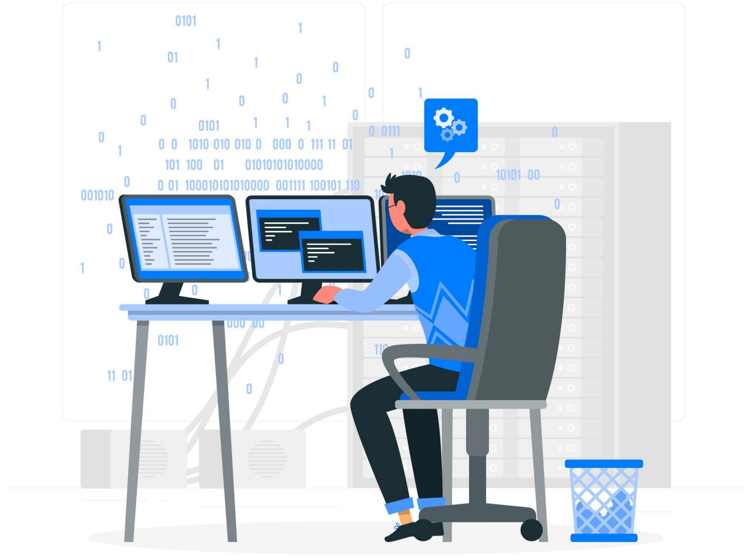
1. A sign in form for the user
2. A table layout of the classes of the logged in user
3. A details page, which displays when a class is selected
4. A form for teachers to edit class details

Additionally, the application should be refactored to:

1. Use route files
2. Use route instances

What you will be given:

A data.js file will be provided with data.



TERM 3 OUTLINE...

PROCESS BREAKDOWN

You will be assessed on your ability to plan and adjust the scope of your project based on the time available, and a realistic estimation of your own pace with weekly monitoring..

PHASE 1: PLANNING

In your module, you will be assessed on your ability to plan and adjust the scope of your project based on the time available, and a realistic estimation of your own pace. In order to plan the scope of your project, write down all the features you'd like to include, and separate into four groups:

- *Essential* (features without which the most basic version of the project cannot exist)
- *Polish* (making it look great, with excellent user experience)
- *Nice-to-have* (the first features to add if there's time after completing the first two groups)
- *Future* (features that may be too big for the time available, or aren't as high a priority as the previous groups)

Remember to **document** your research and **motivation** for your different features.

Use your lists of features to create a schedule for what you will work on each week. You will also be required to complete a series of research deliverables - see the weekly outline for more specifics..

PHASE 2: IMPLEMENTATION PHASE

PROGRESS:

- Keep the implementation of your project up to date in a well-documented repository on GitHub. You will be assessed on having frequent commits throughout the term. *This is for your safety and industry-readiness.*
- Provide a step-by-step user guide in your documentation, with screenshots.

PROJECT MANAGEMENT

- Follow an agile approach to managing your project, iterating with short sprints towards milestones. Align your milestones with your class times to maximise the benefit from your lecturer's feedback. You will be assessed throughout, on your execution of an agile work ethic.

TERM 3 OUTLINE...

PROCESS BREAKDOWN

PHASE 3: TESTING PHASE

An official user-testing session will be held during class time, in which fellow students will test your app and provide feedback on a semi-standardised feedback form that you will make available.

- Consider feedback received within the context of your overall timeline and development priorities, and act accordingly.
- Record insights received from feedback and your resulting decisions in the rationale.

PHASE 4: FINALISATION PHASE

WEEK 7:

1. Compile your project into the submission requirements.
2. Polish your application.
3. Update your documentation.

WEEK 8:

Final Project submit all required deliverables.

Don't forget to save often and create different versions, just in case! Back up your work! Lost work needs to be re-done!

A journal/workbook for all your research, process and planning will be required so that you can keep all your ideas in ONE place and take notes in class. Do not forget to hand in your plagiarism and image reference lists. Your assignments cannot be marked without this documentation.

TERM 3 OUTLINE...

WHAT YOU WILL LEARN

BY COMPLETING THIS PROJECT, LEARNERS WILL HAVE LEARNT TO...

- Create an HTTP web socket server with node
- Accept HTTP requests
- Return HTTP responses
- Include node modules
- Consume a web API.
- Manage their time.

TERM 3 OUTLINE...

SUBMISSION

The submission requirements for this project are:

1. Submit a link to your project's well-documented repo on GitHub
2. Submit a **video** to Google Classroom in which you demonstrate and verbally describe each feature of your app.

Make sure of the following:

Your **repo** must contain your **documentation as a readme**, and your **plagiarism form**; Your Google Classroom submission must be a **LINK** to your repo, a **VIDEO** demonstrating your features (use your voice), and the required **PDF's**.

TERM 2 OUTLINE...

CRITERIA FOR ASSESSMENT

- Scope estimation
- Completeness
- Concept
- Aesthetic
- User experience
- Technical proficiency
- Agile workflow
- Documentation
- Version control (frequency + descriptions)

TERM 3 OUTLINE...

RESOURCES

YOU WILL BE REQUIRED TO READ + RESEARCH THE FOLLOWING:

NODE JS

https://nodejs.org/en/

HOW TO CREATE USEFUL SOFTWARE PROCESS DOCUMENTATION:

http://www.westfallteam.com/Papers/Useful_Software_Process_Documentation.pdf

Documentation:

https://www.atlassian.com/software/confluence/documentation

TERM 2 OUTLINE...

TERM SCHEDULE

WEEK 1:	
Project Briefing	

WEEK 2:	
SKILLS: - Javascript - ExpressJS - Middleware	Introduction to Express Middleware

WEEK 3:	
User Parameters	Implementation

WEEK 4:	
Mid-Term Evaluation	Implementation and homework

WEEK 5:	
POST and DELETE	Implementation and homework

WEEK 6:	
Refactoring	Testing Application

WEEK 7:	
Submission, video and assessment	Final Phase Final Submission Final Test

WEEK 8:
Final Feedback

PROJECT RUBRIC

PROGRESS, TASKS + SPRINT MEETINGS	**20%**
PRESENTATION	**20%**
● **Presentation (Week 9)**	**80%**
○ General Impression	
○ Clarity	
○ Knowledge demonstrated	
● **Style + Ability**	**20%**
TERM PROJECT	**60%**
● Scope estimation	10%
● Completeness	20%
● Concept + Aesthetic	20%
● User experience	10%
● Technical proficiency	25%
● Version control (frequency + descriptions)	5%
● Bonus Effort	10%

"Rule #1: Life is supposed to be fun!"

— John McGrail