WEB APPLICATION DEVELOPMENT

Spotify API

MATTHEW BYRNE
CIAN LARKIN
CARL FLYNN

National College of Ireland

codeanywhere

AIM / WHAT WE WANTED TO DO

LAYOUT OF APPLICATION

FALLBACKS / DEVELOPMENT

TOOLS USED / FEATURES

NODE / REACT EXPLAINED

CONCLUSION / SUMMMARY

CONTENTS

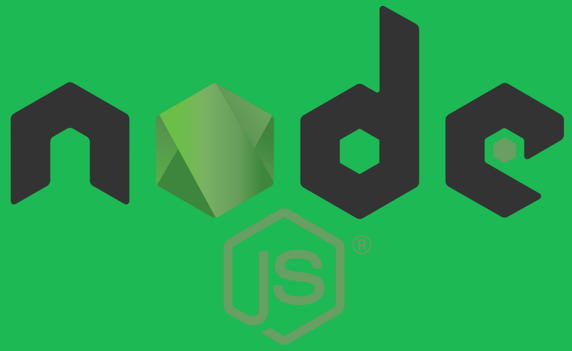
AIMS / WHAT WE WANTED TO DO

AS A TEAM WE BOTH DECIDED THAT WE HAD TO TACKLE THE SPOTIFY API SYSTEM. TO BREAK DOWN THE ELEMENTS WHICH ALLOW THE APPLICATION TO RUN. WE WANTED TO TAKE CERTAIN ASPECTS SUCH AS PLAYING SONGS OR SHOWING ARTIST INFORMATION, TO BE SHOWN ON A SCREEN THAT CAN BE FETCHED USING A ONLINE DATABASE LIBRARY. IN THE NEXT FEW SLIDES, IT DETAILED HOW WE STARTED AND HOW WE ENDED UP WITH THE APPLICATION THAT WAS SUBMITTED.

TOOLS / SYSTEMS USED

React

Spotify API



{json}

JavaScript

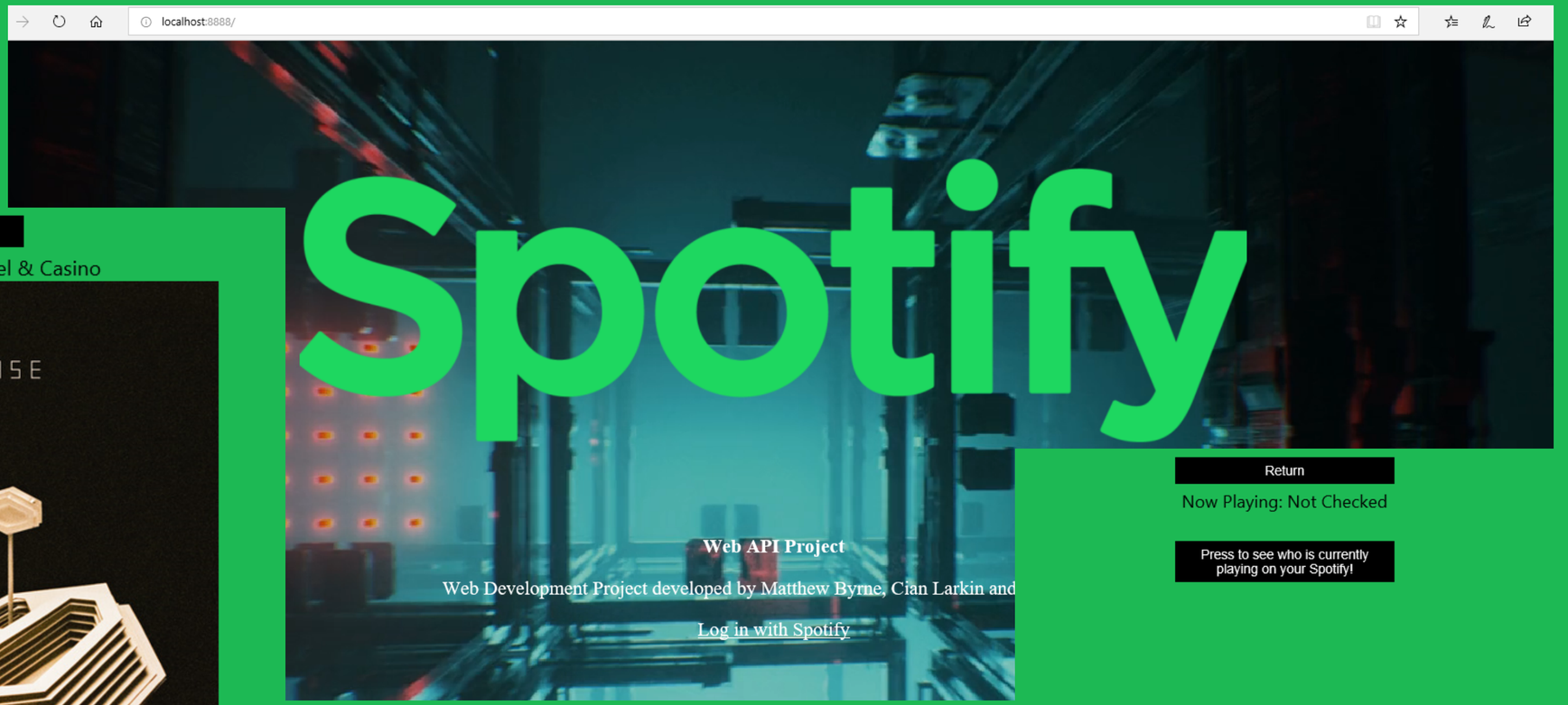
CSS

LAYOUT

Return

Now Playing: Tranquility Base Hotel & Casino

Press to see who is currently playing on your Spotify!



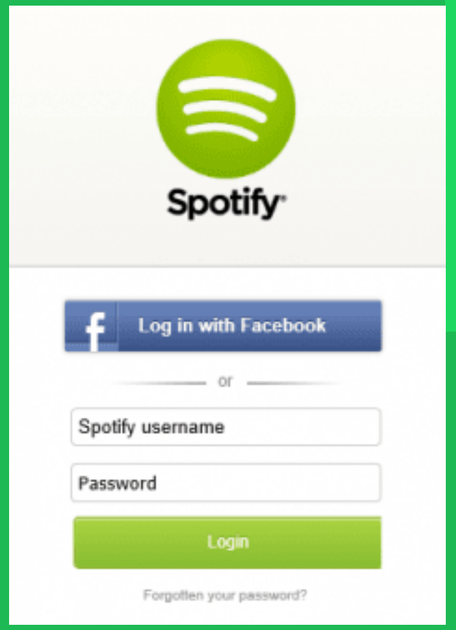
Return

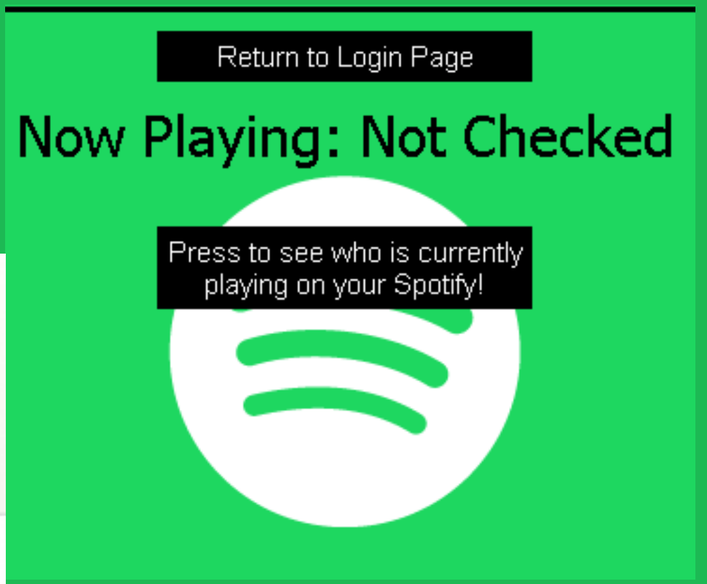
Now Playing: Not Checked

Press to see who is currently playing on your Spotify!

FEATURES

- Authorization of Spotify API account
- Node.js modules installed.
- Main application built with React
- Retrieval of Data and Artwork using refresh token system.
- Json Data included biography and details that link with Javascript.
- Built on two local hosts / containers (3000 and 8888)
- Redirect URI on CodeAnywhere for portable use.

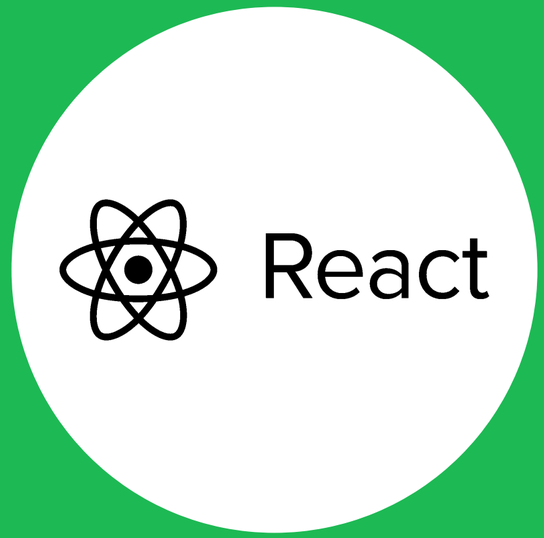




Spotify has been developed since 2006 by a team at Spotify AB, Stockholm, Sweden. The company was founded there by Daniel Ek, former CTO of Stardoll, and Martin Lorentzon, co-founder of TradeDoubler. The headquarters are now located in London, while research and development remains based in Stockholm.

It's used in 61 countries, has 159 million active users and a library of 35 million songs.

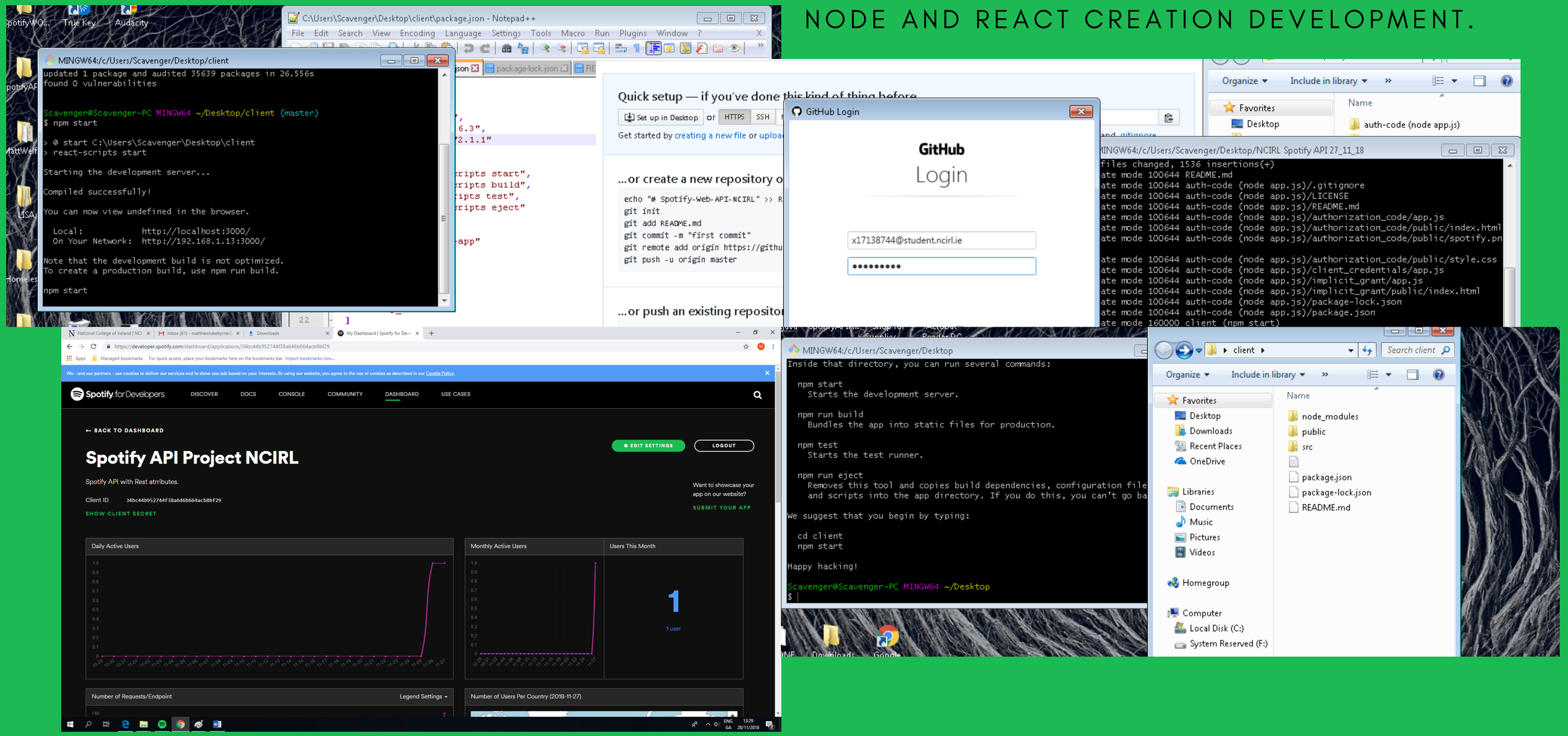
NODE.JS IS A CROSS PLATFORM JAVASCRIPT ENVIROMENT THAT EXCECUTES DATA OUTSIDE OF A BROWSER.
IN TURN IT CAN GENERATE DYNAMIC CONTENT, AND GIVE MORE FLEXIBILITY TO THE USER. NODE IN TURN ALSO WORKS WITH SQL TO MODIFY, ADD OR DELETE DATA AT EASE.



REACT IS A JAVASCRIPT LIBRARY CREATED BY FACEBOOK. THIS SYSTEM IS CONSTANTLY USED TO UPDATE INFORMATION ON THE SERVERS IN REAL TIME. REACT IN TURN IS ALSO A GREAT TOOL FOR BUILDING UI COMPONENTS AS IT WORKS HAND IN HAND WITH CSS. THIS WAS THE MAIN APPLICATION WHERE THE PROJECT WAS BASED.

SCREENSHOTS

NODE AND REACT CREATION DEVELOPMENT.



FALLBACKS

THE IDEA OF PLAYING SPOTIFY MUSIC ON A OUTSIDE APPLICATION WAS A LOT TOUGHER THAN WE THOUGHT. WE DECIDED AT A STAGE IN THE PROJECT TO KEEP THINGS SIMPLE AND DISPLAY SIMPLE INFORMATION TO THE USER THAT WOULD BE OF BENEFIT.

USING A IFRAME SPOTIFY WIDGET TO US WAS QUITE EASY AND THAT ANYBODY COULD DO, SO WE OPTED NOT TO BRING IT IN. ANY SPOTIFY WIDGET HAS NO LOG IN AUTHORIZATION, AND SO DOES NOT IN REAL TIME CHANGE INFORMATION WITH OUR ARTWORK BESIDE IT.

WE STARTED THINKING XML WOULD BE OF POTENTIAL BUT THEN REALISED JSON WAS A MUCH MORE SUITED FORMAT FOR SPOTIFY.

WE ALSO WOULD HAVE LIKED TO ADD A THIRD INDEX PAGE FOR ADDITONAL LISTINGS ENTIRELY USING JSON INFO BUT FOUND THE JSON DATA COULD ONLY BE EVER SHOWN ON JAVASCRIPT DUE TO RUNNING IN REACT.

CONCLUSION

AFTER GETTING CLOSER AND CLOSER TO THE DEADLINE OF THE PROJECT, WE REALISED WHAT WERE DOING WAS VERY AMBITIOUS FOR THE TIME FRAME WE WERE GIVEN.
(BUT STILL LOVED GOING IN THE DEEP END!)

WE REALISED THE POWER OF USING REACT WAS FANTASTIC, BUT TO HAVE EVERY ATTRIBUTE THAT SPOTIFY HAS INCLUDED WOULD BE A MASSIVE TASK OF CODING INVOLVED.

WE WERE HAPPY THAT WE MANAGED TO PRODUCE ARTWORK WITH THE SIMPLE CLICK OF A BUTTON ON ANY SPOTIFY ACCOUNT THROUGH JAVASCRIPT PLUS HAVE DATA RENDERED IN JSON FORMAT.

THE PROJECT WAS OF MASSIVE INTEREST FOR THE TEAM INVOLVED AND HOPE TO SEE MORE OF SIDE SERVER TECHNIQUES IN THE FUTURE FOR BUILDING APPLICATIONS