**Name of Student**: Patrick Delaney

**Name of Project**: StarLife

**Project's Purpose or Goal**: S*imulate the life cycles of stars*

**List the absolute minimum features the project requires to meet this purpose or goal:**

* User manipulates variables of the star’s birth: stellar mass.
* Program will use these variables to calculate and predict the life cycle of stars manipulated by the user.
* Changes in stages in a star’s life cycle will be displayed for the user.

**What tools, frameworks, libraries, APIs, modules and/or other resources (whatever is specific to *your* track, and *your* language) will you use to create this MVP? List them all here. Be specific.**

* Front end: React, CSS, HTML
* Back end: Javascript, Node.js, Redux
* Simulations: Three.js
* A Concise Introduction to Astrophysics – Lecture Notes for FY2450 – M. Kachelrie <http://web.phys.ntnu.no/~mika/skript_astro.pdf>
* Berkeley, UoC: <http://w.astro.berkeley.edu/~echiang/Astro7A/astro7.pdf>
* (a couple other textbooks (will input later)

**If you finish developing the** [**minimum viable product (MVP)**](https://www.learnhowtoprogram.com/lessons/the-minimum-viable-product) **with time to spare, what will you work on next? Describe these features here: Be specific.**

* Display variables: Luminosity, mass, compositions, temperature, gravitational force, fusion force.
* Allow the user to manipulate additional variables: composition, external forces (planetary bodies, other stars), more specific birthplace composition adjustment.
* Program will create planetary bodies based on the mass and composition of the star’s birthplace. Variables adjusted by the user will influence these bodies to become planets and form a solar system.
* Have a “reverse-engineer” feature to allow the user to select one of the many deaths of a star and go backwards to visualize its lifecycle.
* Randomizer feature: Program selects random numbers to manipulate variables. Mystery outcome.
* (super stretch goal) Develop an ‘artificial intelligence’ to compete with the user to build a solar system that lasts the longest.

**What additional tools, frameworks, libraries, APIs, or other resources will these additional features require?**

* To be determined upon more research

**Is there anything else you'd like your instructor to know?**

* At the moment, I don’t feel incredibly comfortable with React. However, I would love to use it, as it’s structure/flow seems to be perfect for this type of application.
* For the display, I’d like to use three.js to have animations. Maybe add some video clips from around the web (I assume I’d have to look into copyrights?). I looked into React video components that seem pretty straight forward.