

be() - a1							1-2	3-4	5-6	7-8	9-10	feedback
Application	Node, Npm, Express, templating are used; the web server runs without errors						Data is read from, removed from, and created on the server; users can input and retrieve data	Data stored in MongoDB is changed (find and update) in the database through the web app	Special care was taken to create a performant or secure web app and database; the web app is deployed	The way the student applies npm, databases, and Node is more advanced than what they were taught in class		
	There is substantial own code; the student can explain the code that exists						The student can explain some parts of their code, how some parts works together, and tech stack	The student can explain every part of their code, how everything works together	The project is complex but still understandable; the student carefully chose every part of their stack	A nerdy conversation can be held; the student can make live changes, explain why software is used instead of alternatives		
	The project is handed in documented, on time, working without technical problems, and on GitHub						The code is readable, consistent and the code, project, and process are partially documented	Code adheres to standards; docs cover the process and what the project is and does	Code quality is good and enforced; docs are more than useful and professional	Code and docs both read like great books and the project is structured logically		
You'll need a > 5.5 for each row to pass: you can't compensate between rows. Each of this rubric's rows is cumulative: for example, to get a 5-6 on application, you also need to have a 1-2 and 3-4.												
student name		student number		lecturer		date (first chance)		grade				