Lawn Mowing website fullstack

Description of your project. If your project was completely finished, how would you imagine that people would use it?

The project is about creating a simple full-stack website for a lawn mowing service, which consists of three parts: a landing page, a user booking page, and a manager dashboard.

Landing page is for lawn mowing business owners to marketing their service to wide variety of people need those serves, and good for SEO, lawn mowing owner can frequently browse it and send question or quotes to website owner or sells consultant. It should be safe and fast loading and beautiful to attract more traffic from internet to get more people click it. Then I should show viewer the service it can provided the work case they finished. Also site manager could convenient to add or remove service when needed.

Booking page is for client who already know the cost of their service and want to make an appointment in future, this page allow them to self-service, register and login to add their bookings, they also will see the booking statue after they posted it and after completed by worker, they will be able to give rating to the service and and worker. The booking with totally have the statue:

- Approved
- Declined
- Completed
- Pending

Client also can cancel the booking when it is in “pending” state.

Manager dashboard is for site managers to manage the booking, quote statues and view the data visualization for user’s activates.

Functionality Completed

- Booking : add, update, delete database from frontend to backend, email autocomplete.
- Serverless post/contact form: validation check, send email after the form sent to AWS cloud.
- Make use of basic and customized hooks like `useContext`, `useState`, `useEffect`,`useRef`, `useForm` to efficiently manage react’s life cycle.
- Used redux library to manage state globally.(Hooks include `useSelect`, `useDispatch`, `useReducer`)
- Test driving development for API, in the end everything works fine.
- Implement firebase Auth, Google autocomplete
- Manipulate data structure to retrieve useful information from data.

Planned to implement but did not finish

- Fully serverless site, everything based on apis(currently user booking remaining)
- GraphQL APIs
- Use Typescript to make the code strong typed use React Native to make it have mobile version both IOS and Android
- Automatically get user's location and calculate his yard square meter
- Use containers and CI/CD when developing this project
- Recaptcha v3 to protect bots

Include, at some point, a sequence of steps that a user may undergo when using your application.

For landing page, "*get quote*", "*contact me*", "*Login*" could be some link for viewer to have interaction with the backend.

When viewer click *"contact me"* they will be directed to a page submit message or question to my email and message data also will be save on AWS dynamoDB for backup to be analysis or used for future training propose.

When *"login"* clicked browser will open anther website which is a webapp run on the heroku and allow user to manage their booking.

Include, at some point, a discussion of the technologies that you have used.

The main thing the project to me is to explore and make use of latest development technology as many as possible. So I may use multiple technologies for just same functionality. Used eslint to better lint the code,

Front-end

Tech	Description
React.js	Core framework for this website
MaterialUI	Design a responsive UI, used template for fast development, import as component when using
Next.js	provide better page oriented router and generate static site, JAMStack builder

Tech	Description
Vercel	Website Deployment platform which provide CDN and automatic host for static site(like gitPage)
Tailwind	CSS framework that you can edit all the style in `className` of a element(used for landing page since which support agile dev)
Google Firebase	Combine with `useContext` hook provide user authentication for whole react & send token to backend

Back-end

Tech	Description
Node.js	Manage the APIs for booking, quotes and client users
AWS Lambda service & Gateway	Provide serverless API
MongoDB & mongoose	NoSQL database to store website data, with mongoose to better control schema
Jest + supertest	test APIs
DynamoDB + AWS SNS	Push email to me after viewer send form contact me
Heroku	Deploy backend on server and provide a domain
Firebase admin	Used for decode and refresh the public key for auth(`uerIdToken`)
~~Bcrypt~~	~~encrypt pass work to protect user privacy in database abandoned after use firebase~~
~~JWT token~~(replaced by firebaseAdmin)	~~Decode token send by firebase and extract user information from it(User Identification)~~

those strikes line are tech replaced by other teches

Challenges

Time schedule is my biggest problem, I split the huge project into many small tasks and making todo list for each tasks, to control the time and schedule each one, but sometime one small bug could take a long time. So in future's project I should learn some useful testing tools to test driving development more professional.

No much function in admin management system, I was planning to add in app email and invoice sending application but got no time lift.

The project is more like a macro one includes everything but not perfection in very details, so I think I should focus on more micro function more in future.

Should use and retrieve more data from user and make more data orientated for admin manage page, since the data are valuable nowadays, business need that data to analysis their client's behaviors to be further give the service they need.

During this project I found for front-end side we should focus on build small and functional component/library that could be reused by others to save our time. People could build their project on the top of your component and concentrate on more important things rather than waste many time in adjust detailed styles.