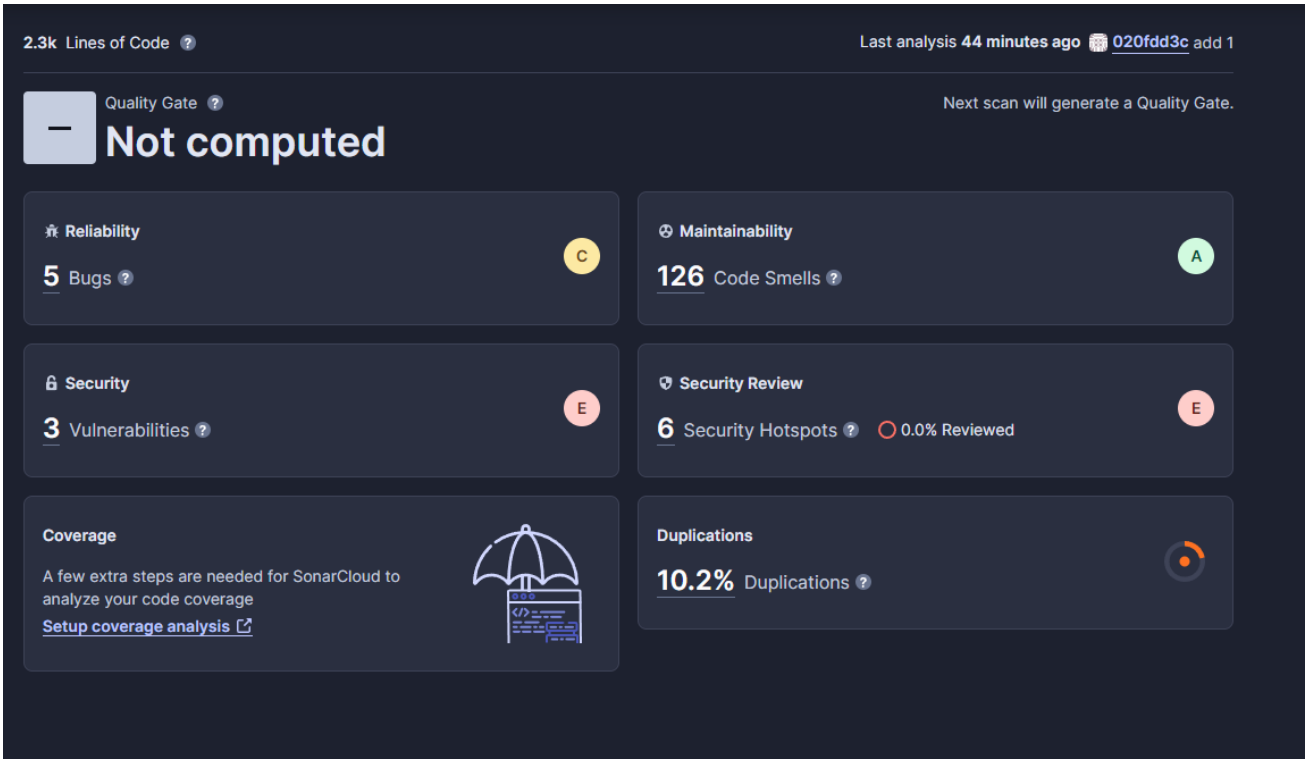
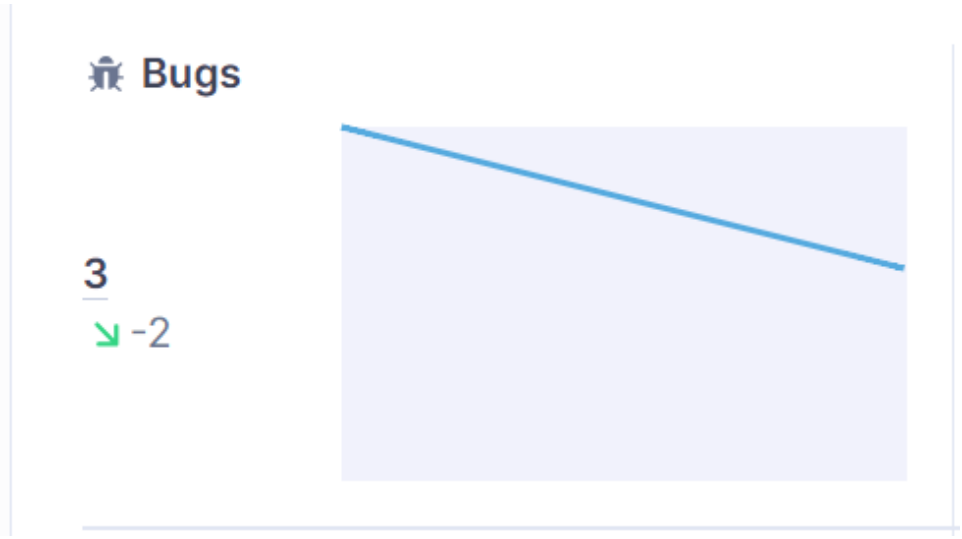
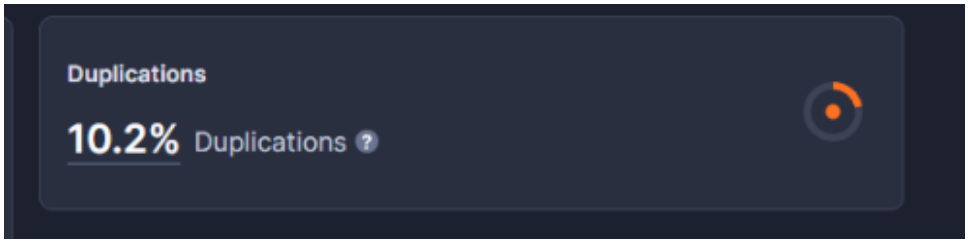


-> delete unused files

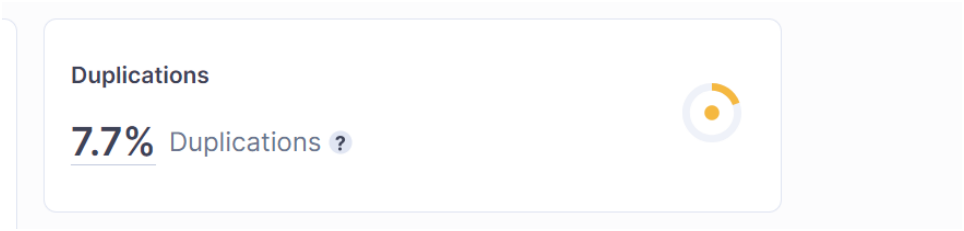


-> fix lines in css/login.css and mortgageCalculator.ejs (duplicates and legend for fieldset)

Started with duplication

Original: 

Removing some double files we had in the css and js (Clean up)

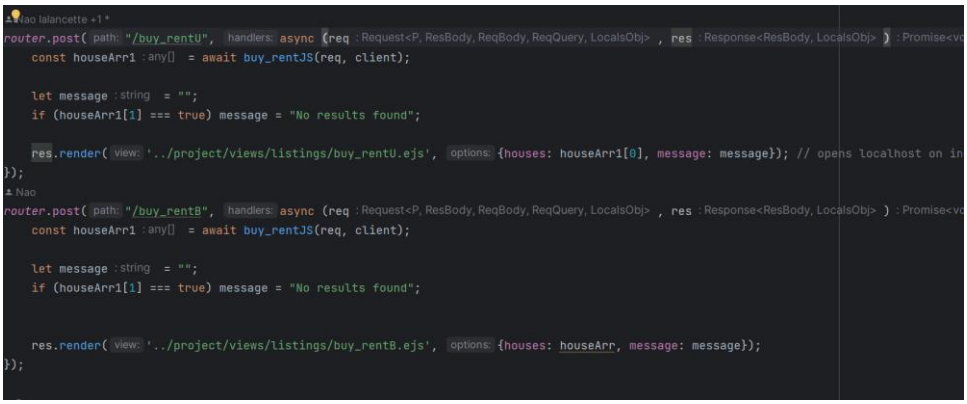


Instead of having a buy_rentB and buy_rentU .post method I created a function re regroup them for less duplication (Buy_rentJS in controller/serverListing)

Before

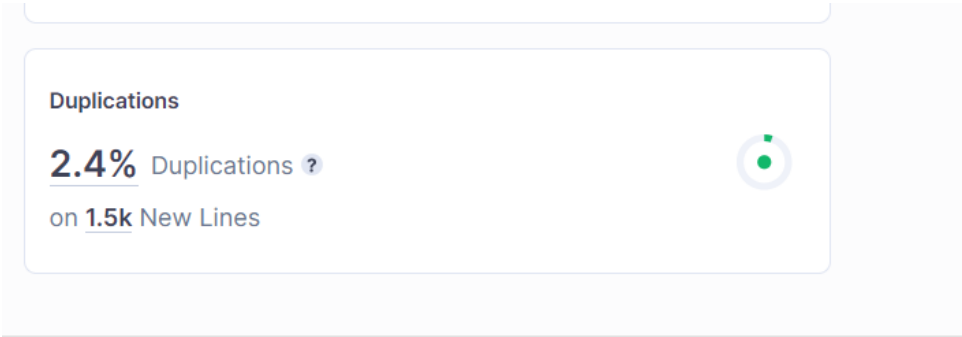
```
router.post( path: "/buy_rentB",  handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<void> =
    let location : string  = req.body.location.toLowerCase();
    let minPrice = req.body.minPrice;
    let maxPrice = req.body.maxPrice;
    let bath = req.body.bath;
    let beds = req.body.beds;
    let yearBuild = req.body.yearBuild;
    let floors = req.body.floors;
    let garage = req.body.garage;
    let prop : ...  = req.body.prop;
    let furnished = req.body.furnished;
    let extra = req.body.extra;
    let propsize = req.body.propsize;
    let listingType = req.body.listing;
    let time = req.body.time;

    const houseArr1 : any[]  = await buy_rentJS(location, minPrice, maxPrice, bath, beds, yearBuild, floors, garage, prop, furnished, extra, props

    let message : string  = "";
    if (houseArr1[1] === true) message = "No results found";


    res.render( view: '../project/views/listings/buy_rentB.ejs',  options: {houses: houseArr, message: message});
```

After

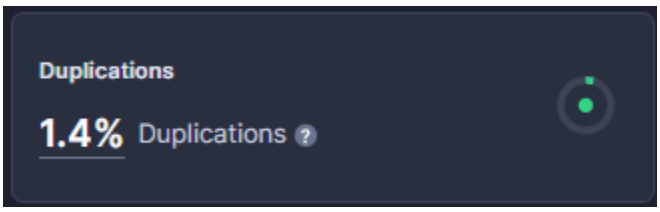


```
Nao lalancette +1 *
router.post( path: "/buy_rentU",  handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<vo
    const houseArr1 : any[]  = await buy_rentJS(req, client);

    let message : string  = "";
    if (houseArr1[1] === true) message = "No results found";

    res.render( view: '../project/views/listings/buy_rentU.ejs',  options: {houses: houseArr1[0], message: message}); // opens localhost on in
});
Nao
router.post( path: "/buy_rentB",  handlers: async (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj> ) : Promise<vo
    const houseArr1 : any[]  = await buy_rentJS(req, client);

    let message : string  = "";
    if (houseArr1[1] === true) message = "No results found";


    res.render( view: '../project/views/listings/buy_rentB.ejs',  options: {houses: houseArr, message: message});
});
```

Cleaned up the serverListing previously made



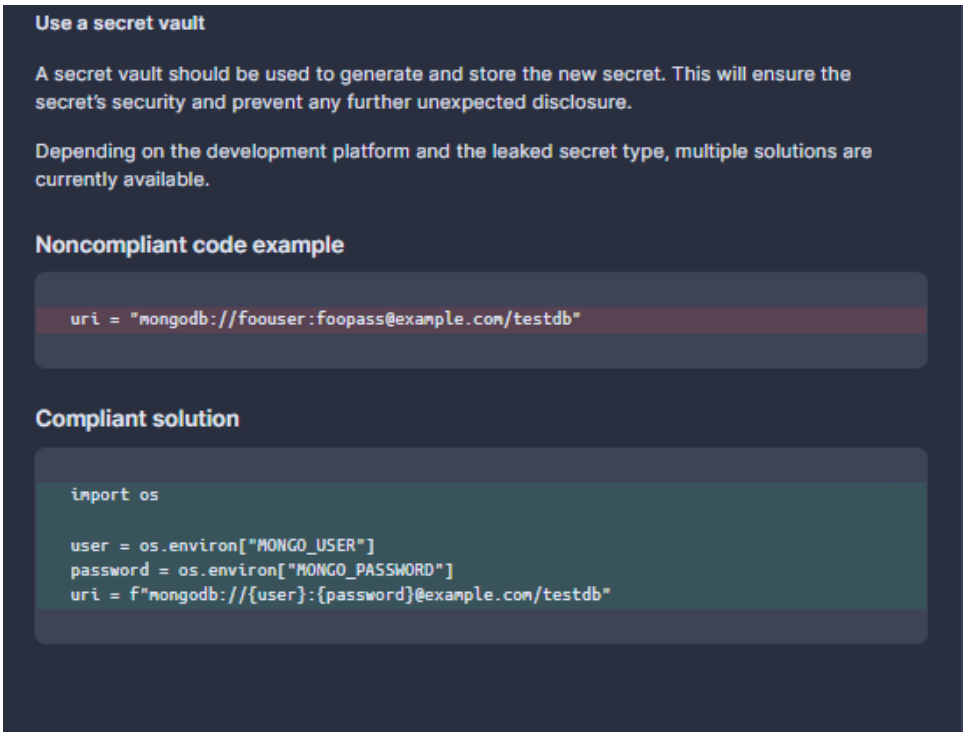
Only half of the commits when through with the other half

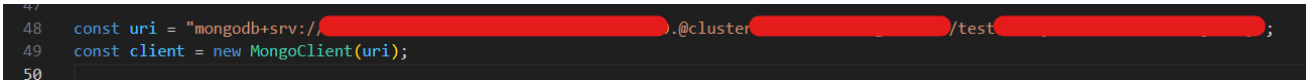


There are two security categories to verify: security vulnerabilities and security reviews.

SECURITY VULNERABILITIES:

Before : According to the CWE, the use of hardcoded passwords and credentials is frowned upon.



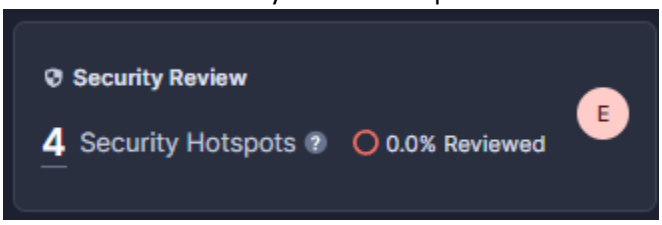


After: After reviewing the code for this vulnerability, it was decided that creating hashes for a connection to the database was ultimately not necessary. The information was hardcoded because our client will never use the website outside of the local host environment. Therefore, I marked the three issues as “Resolved: won’t fix”.

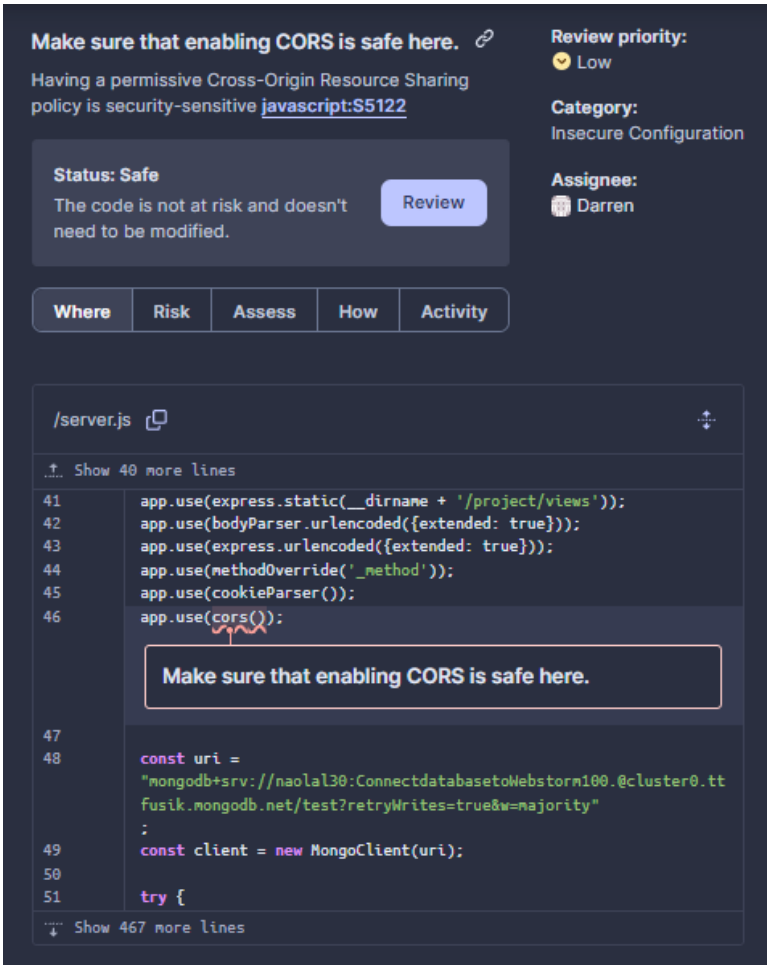
SECURITY REVIEW:

Before: there was no limit size to diskStorage for uploads to the database using MongoDB. This made the program susceptible to DDOS attacks.

This resolved 2 Security Review hotspots:



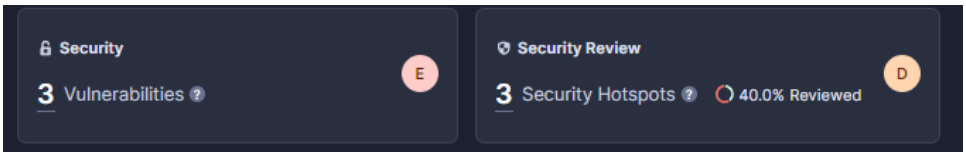
Before:
The use of CORS was deemed unsafe for the server.js file.



After research, CORS was safe to use so long as the website was never hosted outside of the localhost. Since this project will only ever be run locally in a secure Origin network, there is no major security risk linked to the code.

The code was thus deemed `Safe`

After:



Final quality Assurance: