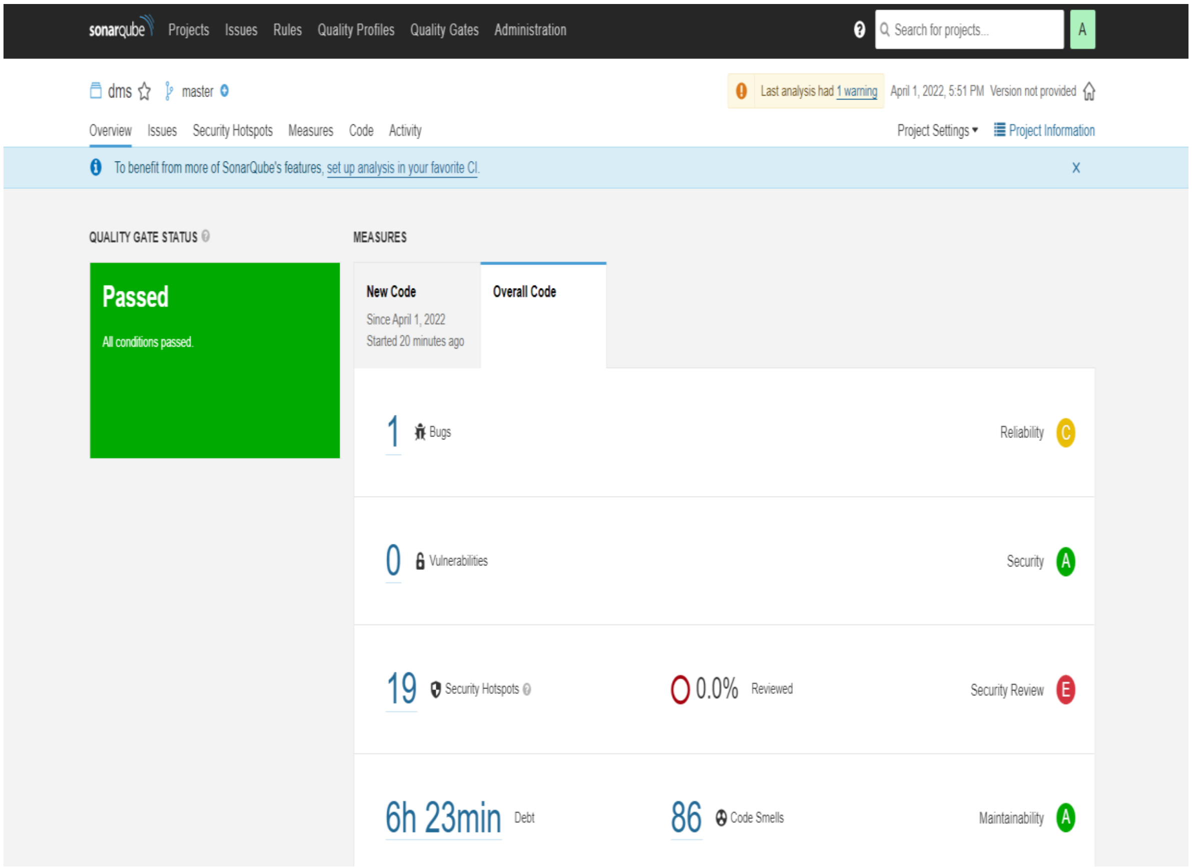
Since we are not using Java for building our application, we decided to go with SonarQube for detecting smells in our code. Initially, when we ran the sonar scanner against our frontend and backend application, we came across around 86 smells.



Last analysis had 1 warning April 1, 2022, 5:51 PM Version not provided

To benefit from more of SonarQube's features, set up analysis in your favorite CI.

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code
Since April 1, 2022
Started 20 minutes ago

Overall Code

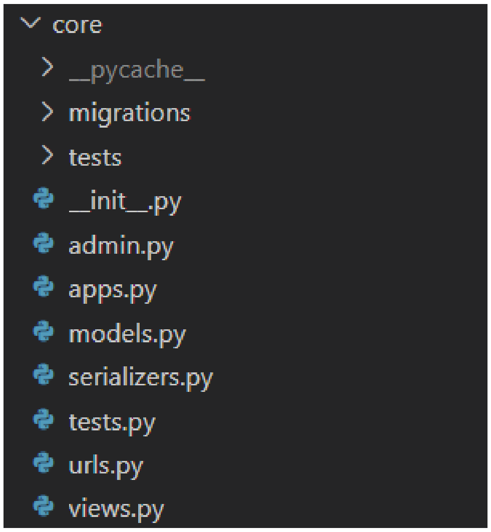
1 Bugs — Reliability C

0 Vulnerabilities — Security A

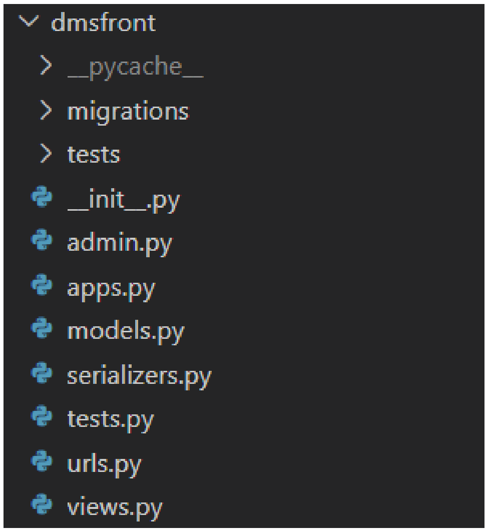
19 Security Hotspots — 0.0% Reviewed — Security Review E

6h 23min Debt — 86 Code Smells — Maintainability A

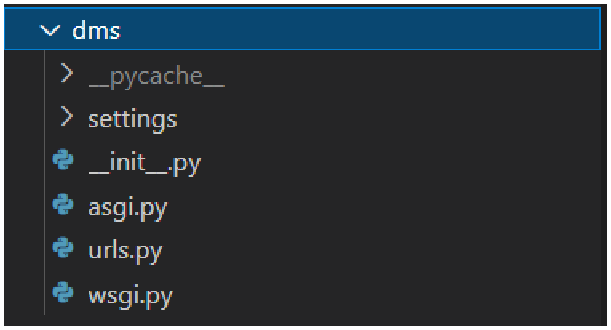
The SonarQube mostly scans implementation level smells. At the design and architecture levels, we have organized our code on both the frontend and backend to maintain proper modularity. In the Django code, we have created an app for authentication called core. We have kept all other implementations into a separate app called “dmsfront”.



```
core
  __pycache__
  migrations
  tests
  __init__.py
  admin.py
  apps.py
  models.py
  serializers.py
  tests.py
  urls.py
  views.py
```

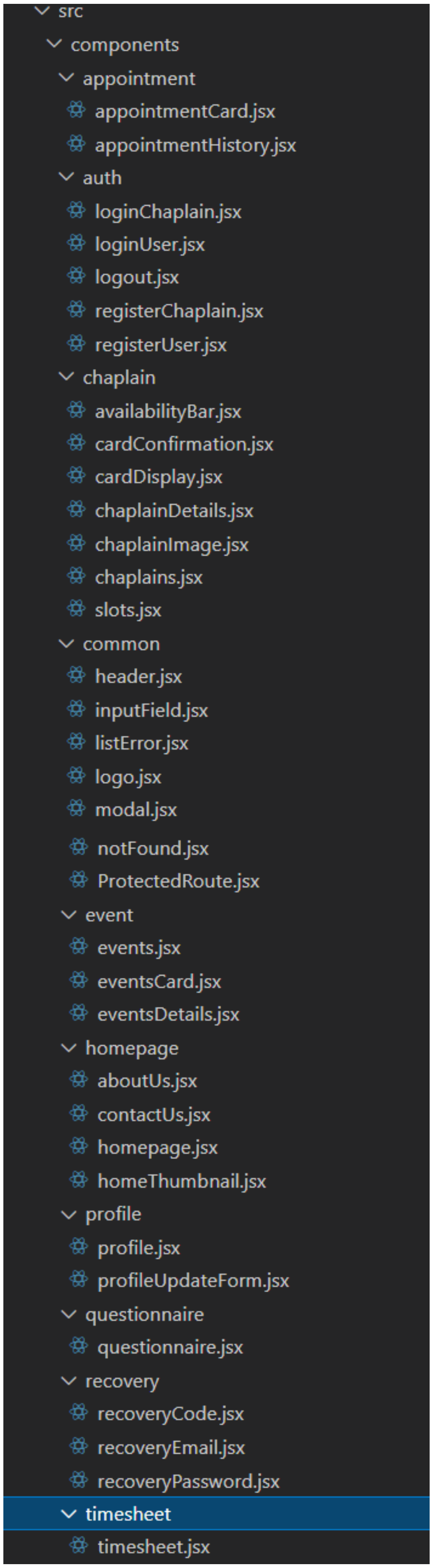


```
dmsfront
  __pycache__
  migrations
  tests
  __init__.py
  admin.py
  apps.py
  models.py
  serializers.py
  tests.py
  urls.py
  views.py
```



```
dms
  __pycache__
  settings
  __init__.py
  asgi.py
  urls.py
  wsgi.py
```

In the React code, we have organized all the components groups into separate directories to maintain modularity.



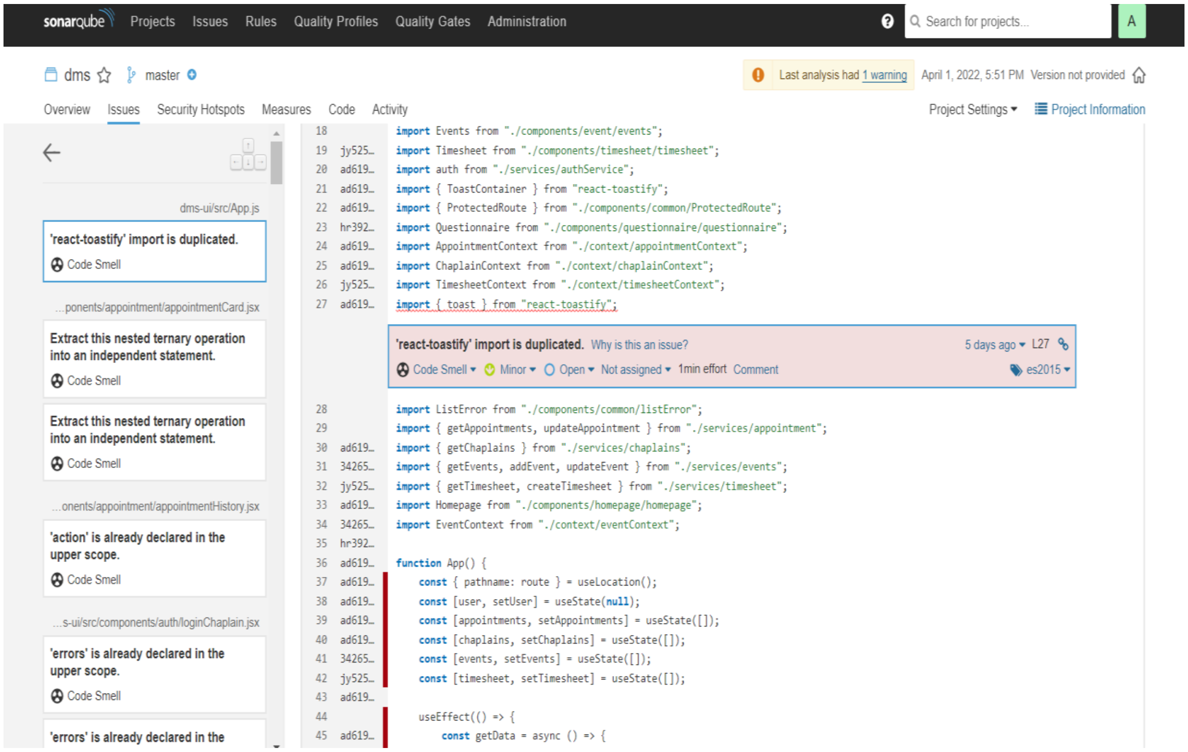
```
src
  components
    appointment
      appointmentCard.jsx
      appointmentHistory.jsx
    auth
      loginChaplain.jsx
      loginUser.jsx
      logout.jsx
      registerChaplain.jsx
      registerUser.jsx
    chaplain
      availabilityBar.jsx
      cardConfirmation.jsx
      cardDisplay.jsx
      chaplainDetails.jsx
      chaplainImage.jsx
      chaplains.jsx
      slots.jsx
    common
      header.jsx
      inputField.jsx
      listError.jsx
      logo.jsx
      modal.jsx
      notFound.jsx
      ProtectedRoute.jsx
    event
      events.jsx
      eventsCard.jsx
      eventsDetails.jsx
    homepage
      aboutUs.jsx
      contactUs.jsx
      homepage.jsx
      homeThumbnail.jsx
    profile
      profile.jsx
      profileUpdateForm.jsx
    questionnaire
      questionnaire.jsx
    recovery
      recoveryCode.jsx
      recoveryEmail.jsx
      recoveryPassword.jsx
    timesheet
      timesheet.jsx
```

Resolution of smells:

We are highlighting below a few of the smells that we encountered when we first ran the sonar scanner. Out of 86 smells, most of the smells are of the same type throughout the project. Hence, instead of mentioning all the 86 smells, we have included only one out of all the types that we saw.

Duplicated import:

The imports were duplicated in the component file. We moved the import to a single line to resolve this smell.



dms-ui/src/App.js

'react-toastify' import is duplicated.
Code Smell

...ponents/appointment/appointmentCard.jsx

Extract this nested ternary operation into an independent statement.
Code Smell

Extract this nested ternary operation into an independent statement.
Code Smell

...onents/appointment/appointmentHistory.jsx

'action' is already declared in the upper scope.
Code Smell

...s-ui/src/components/auth/loginChaplain.jsx

'errors' is already declared in the upper scope.
Code Smell

'errors' is already declared in the

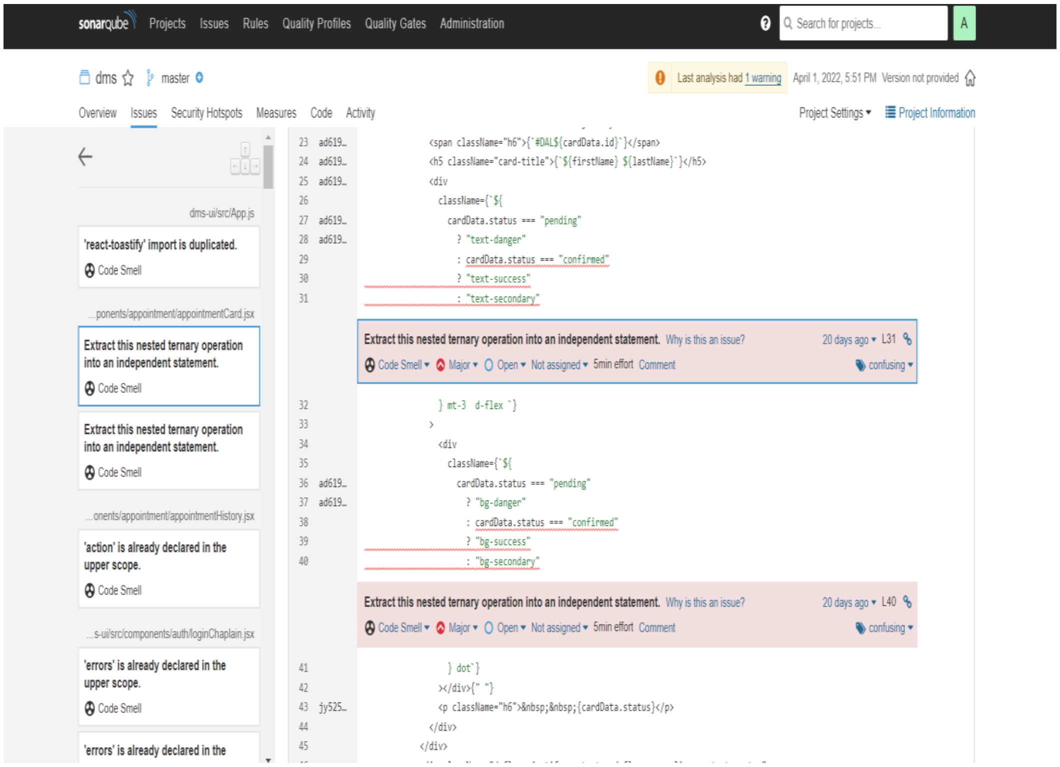
```
import Events from "./components/event/events";
import Timesheet from "./components/timesheet/timesheet";
import auth from "./services/authService";
import { ToastContainer } from "react-toastify";
import { ProtectedRoute } from "./components/common/ProtectedRoute";
import Questionnaire from "./components/questionnaire/questionnaire";
import AppointmentContext from "./context/appointmentContext";
import ChaplainContext from "./context/chaplainContext";
import TimesheetContext from "./context/timesheetContext";
import { toast } from "react-toastify";
```

'react-toastify' import is duplicated. Why is this an issue? 5 days ago L27
Code Smell Minor Open Not assigned 1min effort Comment es2015

```
import ListError from "./components/common/listError";
import { getAppointments, updateAppointment } from "./services/appointment";
import { getChaplains } from "./services/chaplains";
import { getEvents, addEvent, updateEvent } from "./services/events";
import { getTimesheet, createTimesheet } from "./services/timesheet";
import Homepage from "./components/homepage/homepage";
import EventContext from "./context/eventContext";

function App() {
    const { pathname: route } = useLocation();
    const [user, setUser] = useState(null);
    const [appointments, setAppointments] = useState([]);
    const [chaplains, setChaplains] = useState([]);
    const [events, setEvents] = useState([]);
    const [timesheet, setTimesheet] = useState([]);

    useEffect(() => {
        const getData = async () => {
```

Extract nested ternary operator:

We removed the nested ternary operator into a separate method to improve the code readability.



dms-ui/src/App.js

'react-toastify' import is duplicated.
Code Smell

...ponents/appointment/appointmentCard.jsx

Extract this nested ternary operation into an independent statement.
Code Smell

Extract this nested ternary operation into an independent statement.
Code Smell

...onents/appointment/appointmentHistory.jsx

'action' is already declared in the upper scope.
Code Smell

...s-ui/src/components/auth/loginChaplain.jsx

'errors' is already declared in the upper scope.
Code Smell

'errors' is already declared in the

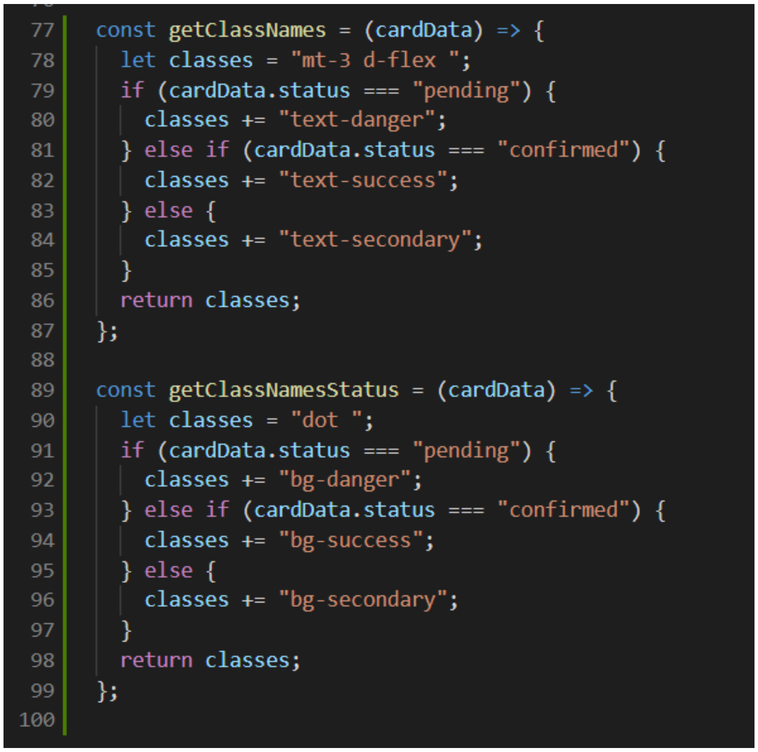
```
<span className="h6">{`#DAL${cardData.id}`}</span>
<h5 className="card-title">{`${firstName} ${lastName}`}</h5>
<div
  className={`${
    cardData.status === "pending"
      ? "text-danger"
      : cardData.status === "confirmed"
      ? "text-success"
      : "text-secondary"
```

Extract this nested ternary operation into an independent statement. Why is this an issue? 20 days ago L31
Code Smell Major Open Not assigned 5min effort Comment confusing

```
  } mt-3  d-flex `}
>
  <div
    className={`${
      cardData.status === "pending"
        ? "bg-danger"
        : cardData.status === "confirmed"
        ? "bg-success"
        : "bg-secondary"
```

Extract this nested ternary operation into an independent statement. Why is this an issue? 20 days ago L40
Code Smell Major Open Not assigned 5min effort Comment confusing

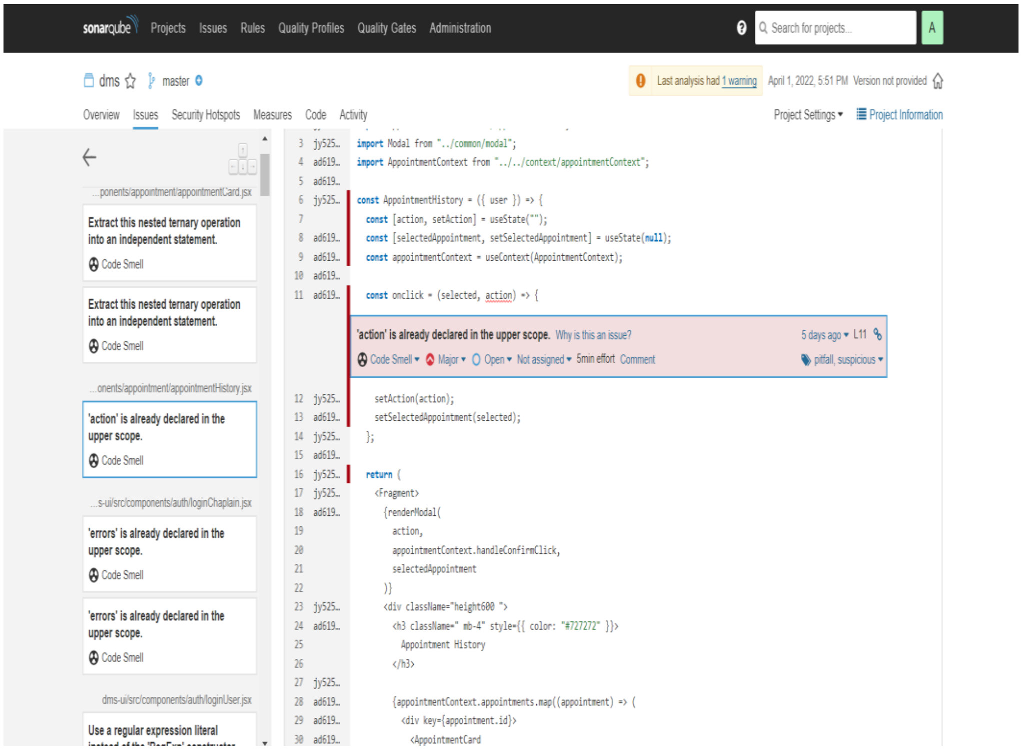
```
    } dot`}
  ></div>{" "}
  <p className="h6">&nbsp;&nbsp;{cardData.status}</p>
</div>
</div>
```



```
const getClassNames = (cardData) => {
  let classes = "mt-3 d-flex ";
  if (cardData.status === "pending") {
    classes += "text-danger";
  } else if (cardData.status === "confirmed") {
    classes += "text-success";
  } else {
    classes += "text-secondary";
  }
  return classes;
};

const getClassNamesStatus = (cardData) => {
  let classes = "dot ";
  if (cardData.status === "pending") {
    classes += "bg-danger";
  } else if (cardData.status === "confirmed") {
    classes += "bg-success";
  } else {
    classes += "bg-secondary";
  }
  return classes;
};
```

Declaration of the same variable in the upper scope:

We renamed the variable in the inner scope to resolve this smell. This improves the code readability and maintainability.



Last analysis had 1 warning April 1, 2022, 5:51 PM Version not provided

...ponents/appointment/appointmentCard.jsx

Extract this nested ternary operation into an independent statement.
Code Smell

Extract this nested ternary operation into an independent statement.
Code Smell

...onents/appointment/appointmentHistory.jsx

'action' is already declared in the upper scope.
Code Smell

...s-ui/src/components/auth/loginChaplain.jsx

'errors' is already declared in the upper scope.
Code Smell

'errors' is already declared in the upper scope.
Code Smell

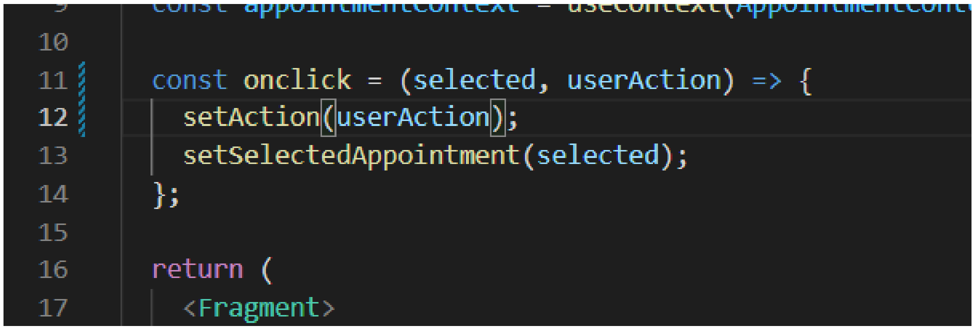
dms-ui/src/components/auth/loginUser.jsx

Use a regular expression literal

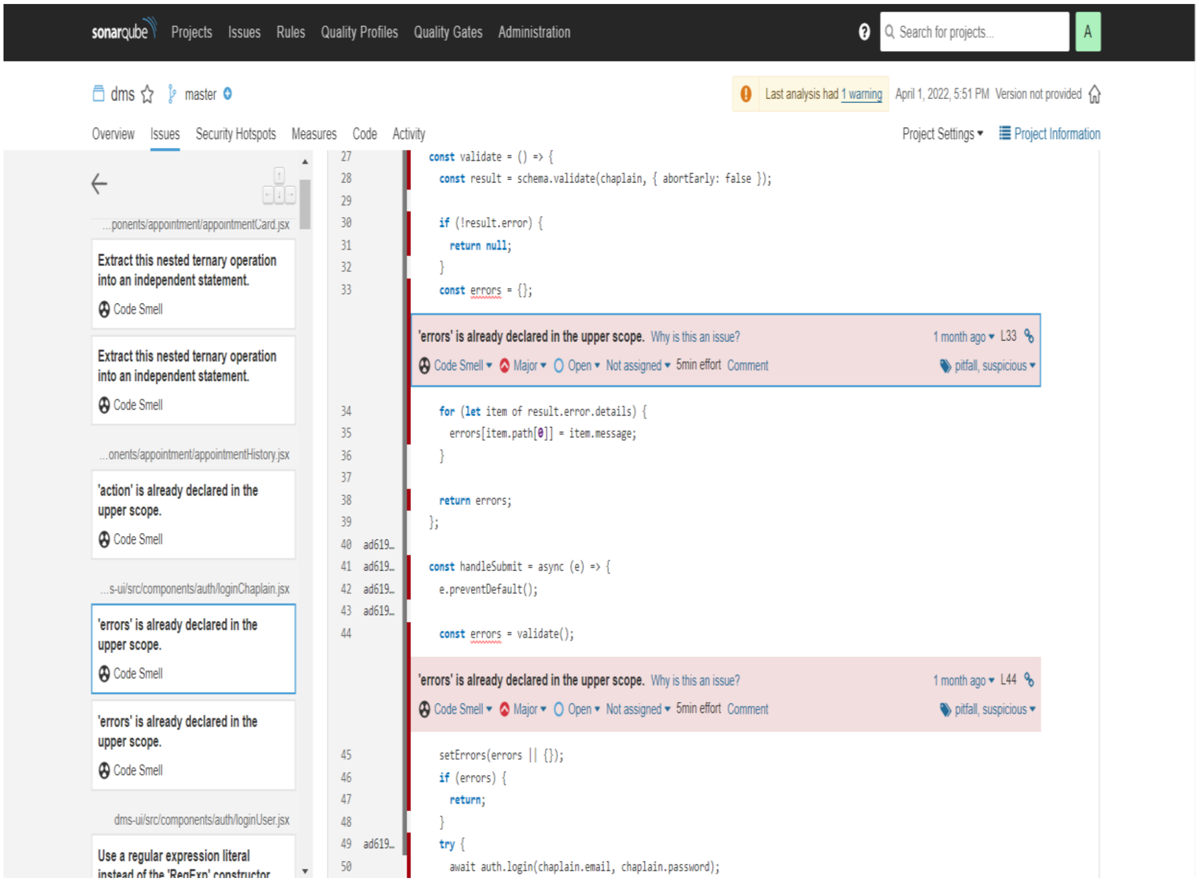
```
import Modal from "../common/modal";
import AppointmentContext from "../../context/appointmentContext";

const AppointmentHistory = ({ user }) => {
  const [action, setAction] = useState("");
  const [selectedAppointment, setSelectedAppointment] = useState(null);
  const appointmentContext = useContext(AppointmentContext);

  const onclick = (selected, action) => {
```

'action' is already declared in the upper scope. Why is this an issue? 5 days ago ▾ L11
Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment pitfall, suspicious ▾

```
    setAction(action);
    setSelectedAppointment(selected);
  };

  return (
    <Fragment>
      {renderModal(
        action,
        appointmentContext.handleConfirmClick,
        selectedAppointment
      )}
      <div className="height600 ">
        <h3 className=" mb-4" style={{ color: "#727272" }}>
          Appointment History
        </h3>

        {appointmentContext.appointments.map((appointment) => (
          <div key={appointment.id}>
            <AppointmentCard
```



```
  const appointmentContext = useContext(AppointmentConte

  const onclick = (selected, userAction) => {
    setAction(userAction);
    setSelectedAppointment(selected);
  };

  return (
    <Fragment>
```



...ponents/appointment/appointmentCard.jsx

Extract this nested ternary operation into an independent statement.
Code Smell

Extract this nested ternary operation into an independent statement.
Code Smell

...onents/appointment/appointmentHistory.jsx

'action' is already declared in the upper scope.
Code Smell

...s-ui/src/components/auth/loginChaplain.jsx

'errors' is already declared in the upper scope.
Code Smell

'errors' is already declared in the upper scope.
Code Smell

dms-ui/src/components/auth/loginUser.jsx

Use a regular expression literal instead of the 'RegExp' constructor

```
const validate = () => {
  const result = schema.validate(chaplain, { abortEarly: false });

  if (!result.error) {
    return null;
  }
  const errors = {};
```

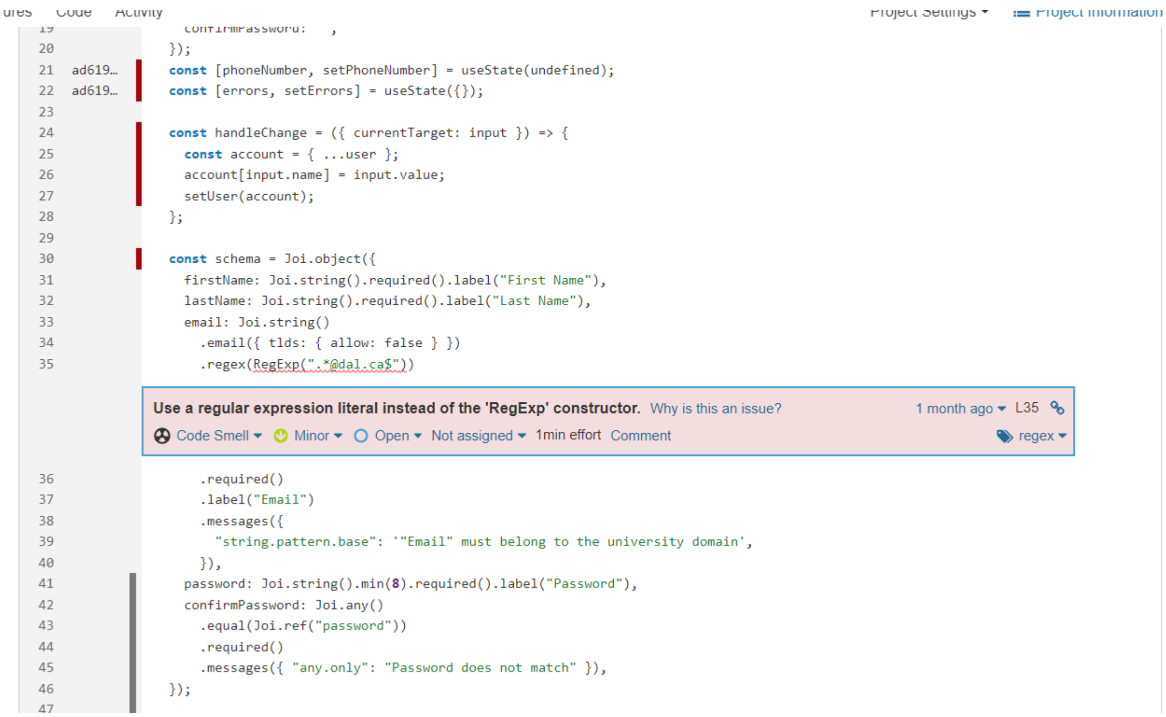
'errors' is already declared in the upper scope. Why is this an issue? 1 month ago ▾ L33
Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment pitfall, suspicious ▾

```
  for (let item of result.error.details) {
    errors[item.path[0]] = item.message;
  }

  return errors;
};

const handleSubmit = async (e) => {
  e.preventDefault();

  const errors = validate();
```

'errors' is already declared in the upper scope. Why is this an issue? 1 month ago ▾ L44
Code Smell ▾ Major ▾ Open ▾ Not assigned ▾ 5min effort Comment pitfall, suspicious ▾

```
  setErrors(errors || {});
  if (errors) {
    return;
  }
  try {
    await auth.login(chaplain.email, chaplain.password);
```

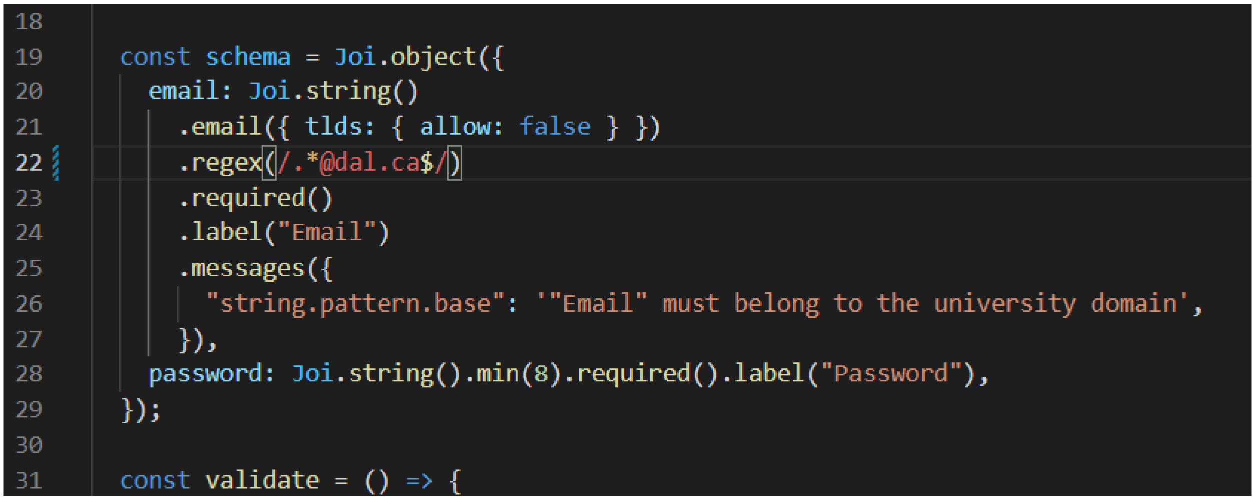
Replaced an REGEXP constructor with the regex literal:



```
  confirmPassword: "",
});
const [phoneNumber, setPhoneNumber] = useState(undefined);
const [errors, setErrors] = useState({});

const handleChange = ({ currentTarget: input }) => {
  const account = { ...user };
  account[input.name] = input.value;
  setUser(account);
};

const schema = Joi.object({
  firstName: Joi.string().required().label("First Name"),
  lastName: Joi.string().required().label("Last Name"),
  email: Joi.string()
    .email({ tlds: { allow: false } })
    .regex(RegExp(".*@dal.ca$"))
```

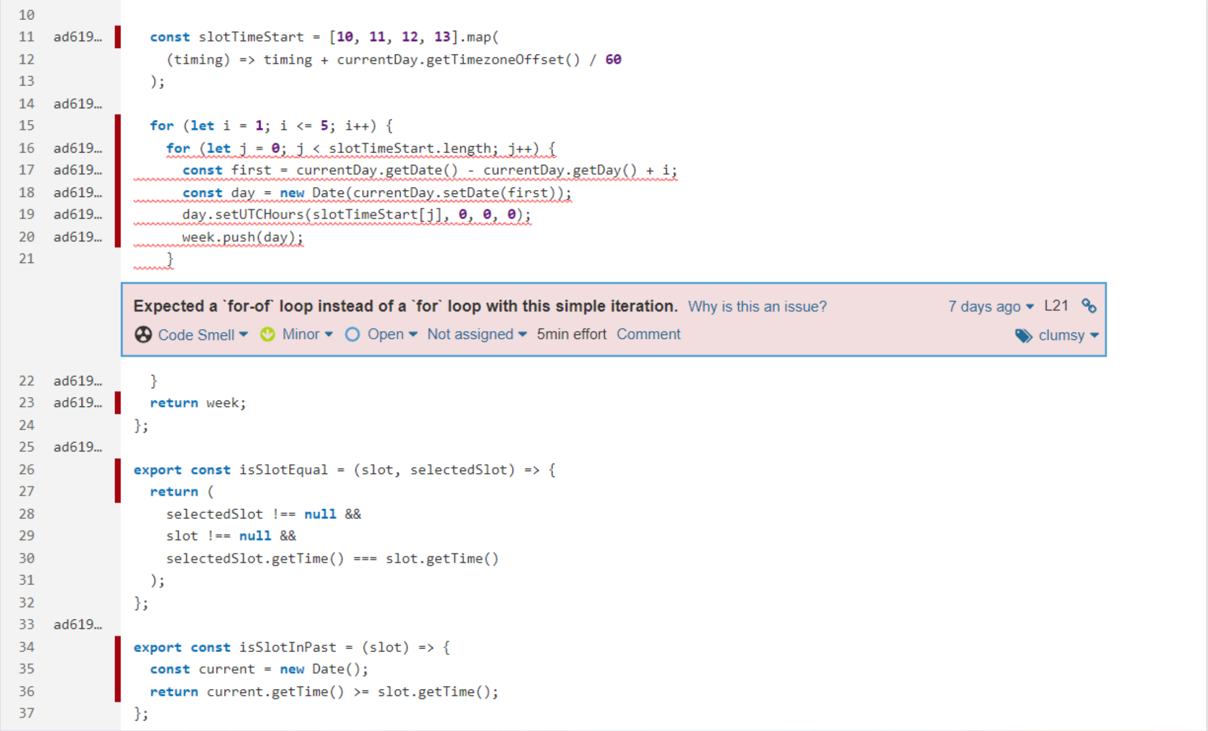
Use a regular expression literal instead of the 'RegExp' constructor. Why is this an issue? 1 month ago ▾ L35
Code Smell ▾ Minor ▾ Open ▾ Not assigned ▾ 1min effort Comment regex ▾

```
    .required()
    .label("Email")
    .messages({
      "string.pattern.base": '"Email" must belong to the university domain',
    }),
  password: Joi.string().min(8).required().label("Password"),
  confirmPassword: Joi.any()
    .equal(Joi.ref("password"))
    .required()
    .messages({ "any.only": "Password does not match" }),
});
```



```
const schema = Joi.object({
  email: Joi.string()
    .email({ tlds: { allow: false } })
    .regex(/.*@dal.ca$/)
    .required()
    .label("Email")
    .messages({
      "string.pattern.base": '"Email" must belong to the university domain',
    }),
  password: Joi.string().min(8).required().label("Password"),
});

const validate = () => {
```

Replaced a normal for loop with the for-of loop

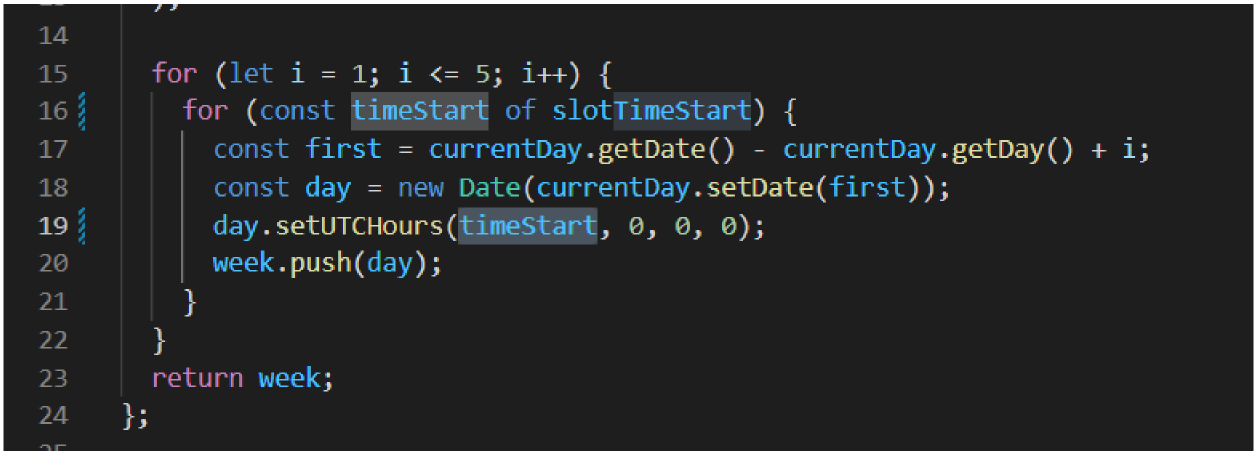


```
  const slotTimeStart = [10, 11, 12, 13].map(
    (timing) => timing + currentDay.getTimezoneOffset() / 60
  );

  for (let i = 1; i <= 5; i++) {
    for (let j = 0; j < slotTimeStart.length; j++) {
      const first = currentDay.getDate() - currentDay.getDay() + i;
      const day = new Date(currentDay.setDate(first));
      day.setUTCHours(slotTimeStart[j], 0, 0, 0);
      week.push(day);
    }
```

Expected a `for-of` loop instead of a `for` loop with this simple iteration. Why is this an issue? 7 days ago ▾ L21

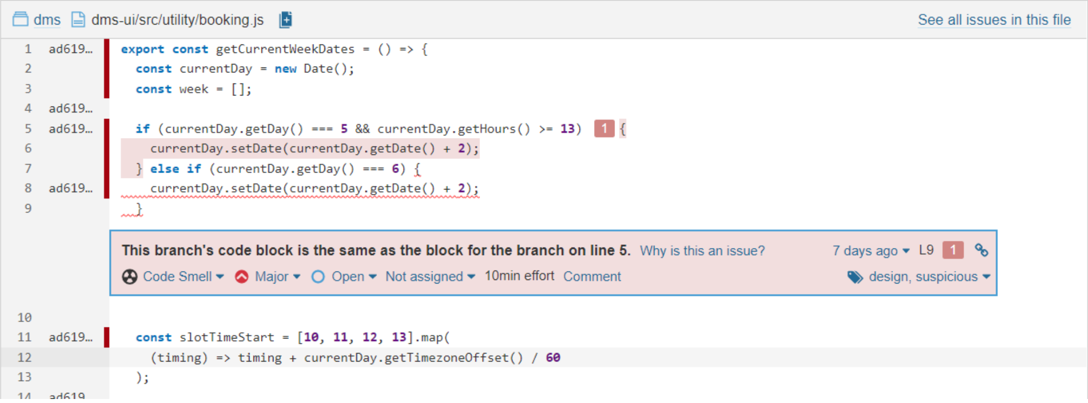
Code Smell ▾ Minor ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment clumsy ▾

```
  }
  return week;
};

export const isSlotEqual = (slot, selectedSlot) => {
  return (
    selectedSlot !== null &&
    slot !== null &&
    selectedSlot.getTime() === slot.getTime()
  );
};

export const isSlotInPast = (slot) => {
  const current = new Date();
  return current.getTime() >= slot.getTime();
};
```



```
  for (let i = 1; i <= 5; i++) {
    for (const timeStart of slotTimeStart) {
      const first = currentDay.getDate() - currentDay.getDay() + i;
      const day = new Date(currentDay.setDate(first));
      day.setUTCHours(timeStart, 0, 0, 0);
      week.push(day);
    }
  }
  return week;
};
```

Removed duplicated code:

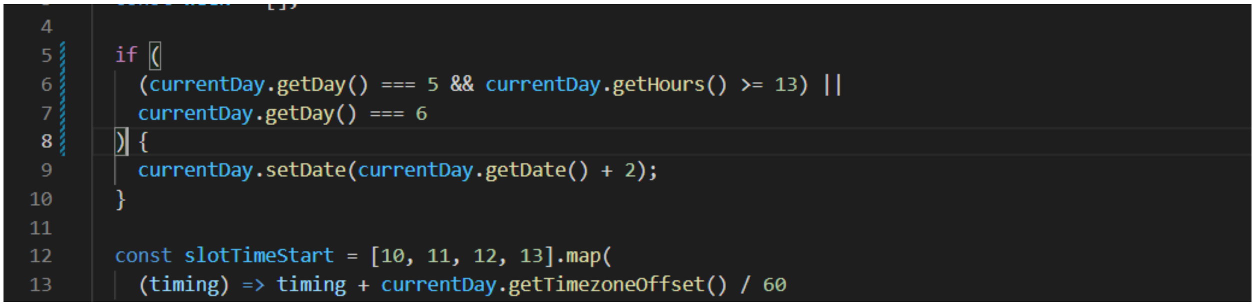


```
1   ad619…  export const getCurrentWeekDates = () => {
2             const currentDay = new Date();
3             const week = [];
4   ad619…
5   ad619…    if (currentDay.getDay() === 5 && currentDay.getHours() >= 13)  1 {
6               currentDay.setDate(currentDay.getDate() + 2);
7             } else if (currentDay.getDay() === 6) {
8   ad619…      currentDay.setDate(currentDay.getDate() + 2);
9             }
```

This branch's code block is the same as the block for the branch on line 5. Why is this an issue? 7 days ago ▾ L9 1

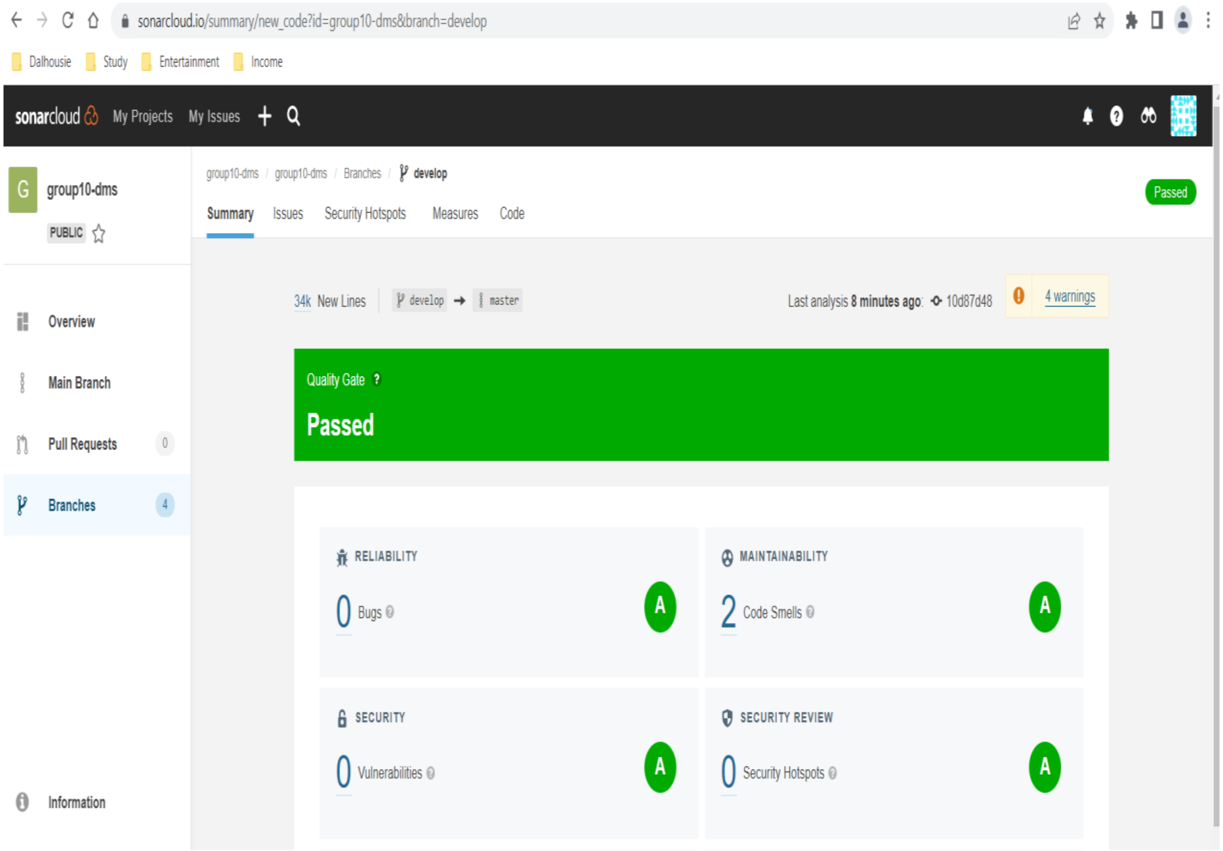
Code Smell ▾ Major ▾ ○ Open ▾ Not assigned ▾ 10min effort Comment design, suspicious ▾

```
10
11  ad619…    const slotTimeStart = [10, 11, 12, 13].map(
12              (timing) => timing + currentDay.getTimezoneOffset() / 60
13            );
14  ad619
```

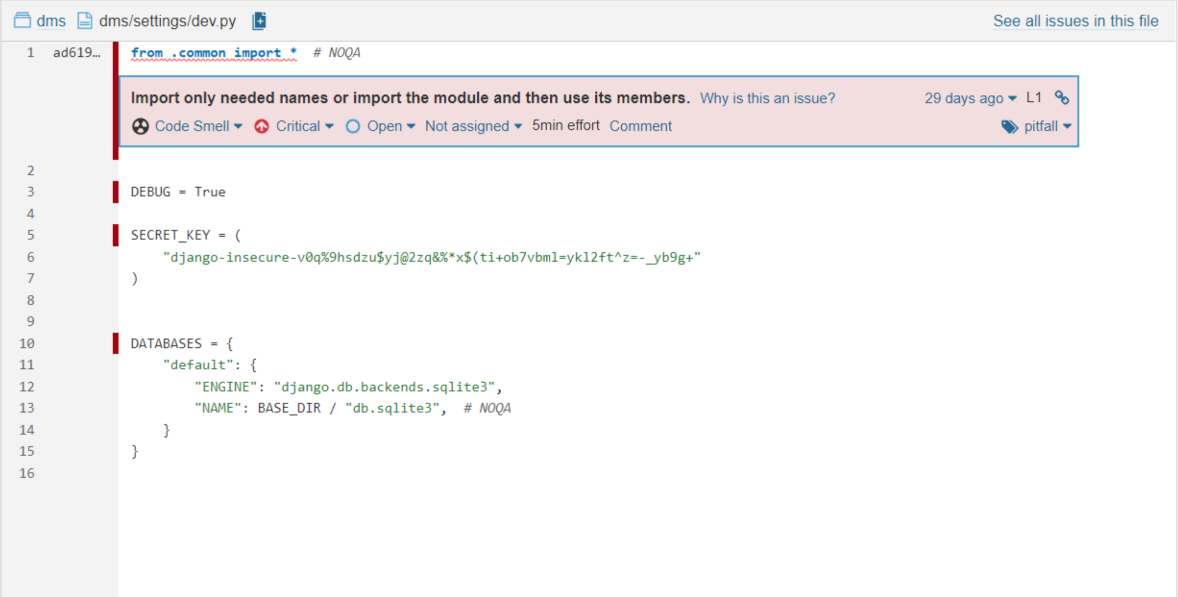


```
  if (
    (currentDay.getDay() === 5 && currentDay.getHours() >= 13) ||
    currentDay.getDay() === 6
  ) {
    currentDay.setDate(currentDay.getDate() + 2);
  }

  const slotTimeStart = [10, 11, 12, 13].map(
    (timing) => timing + currentDay.getTimezoneOffset() / 60
```

In the end, we were able to resolve 84 out of 86 smells.



The two code smells that we didn’t resolve were in the python code. This smell says that we need to import modules only what we need. We were able to resolve this smell locally. However, to run our code on the CICD pipeline we had to separate the production and the development files into a separate directory and import all the modules using “*”. This is a recommended approach for deploying a Django application on the Heroku platform. Hence, we didn't resolve the smell.

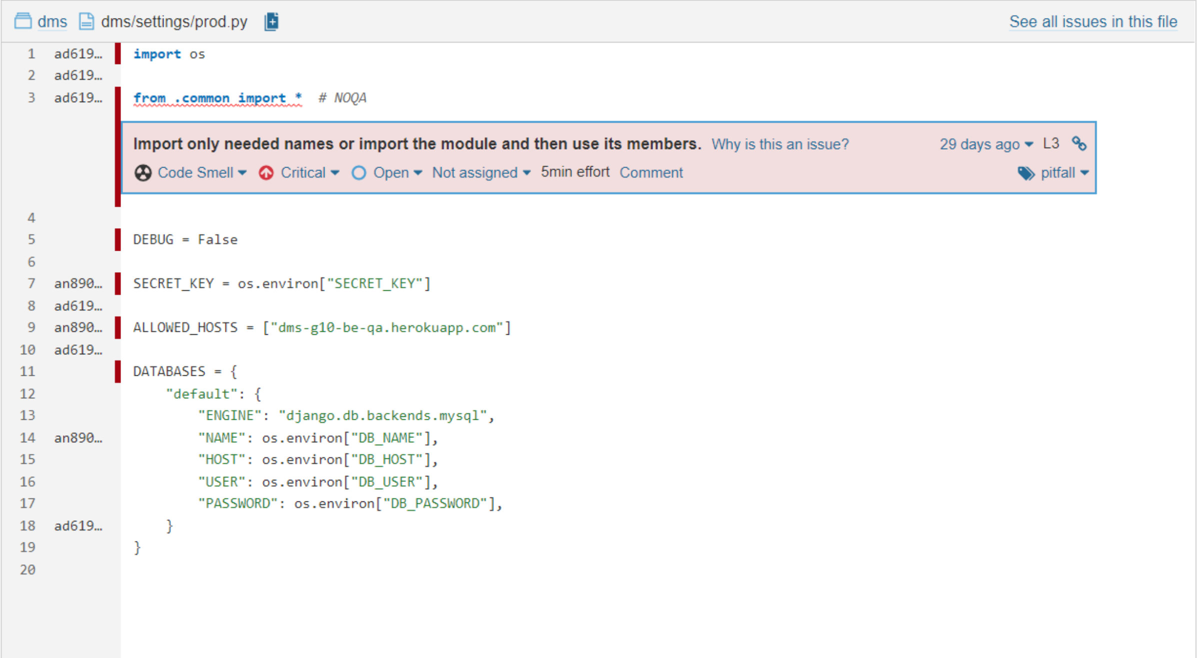


```
from .common import *  # NOQA
```

Import only needed names or import the module and then use its members. Why is this an issue? 29 days ago ▾ L1

Code Smell ▾ Critical ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment pitfall ▾

```

DEBUG = True

SECRET_KEY = (
    "django-insecure-v0q%9hsdzu$yj@2zq&%*x$(ti+ob7vbml=ykl2ft^z=-_yb9g+"
)


DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": BASE_DIR / "db.sqlite3",  # NOQA
    }
}
```



```
import os

from .common import *  # NOQA
```

Import only needed names or import the module and then use its members. Why is this an issue? 29 days ago ▾ L3

Code Smell ▾ Critical ▾ ○ Open ▾ Not assigned ▾ 5min effort Comment pitfall ▾

```

DEBUG = False

SECRET_KEY = os.environ["SECRET_KEY"]

ALLOWED_HOSTS = ["dms-g10-be-qa.herokuapp.com"]

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.mysql",
        "NAME": os.environ["DB_NAME"],
        "HOST": os.environ["DB_HOST"],
        "USER": os.environ["DB_USER"],
        "PASSWORD": os.environ["DB_PASSWORD"],
    }
}
```