Front End Coding Interview: Feature flags

Our company wants to have the ability to show different features to different users (this is also sometimes called A/B testing). There is already an existing backend system for this, it stores information about which features should be enabled for current user. For the purposes of task at hand this backend system is abstracted away like this:

```
// returns the state of *all* features for current user
function fetchAllFeatures() {
  // in reality, this would have been a `fetch` call:
  // `fetch("/api/features/all")`
  return new Promise(resolve => {
    const sampleFeatures = {
      "extended-summary": true,
      "feedback-dialog": false
    };

    setTimeout(resolve, 100, sampleFeatures);
  });
}
```

You are tasked with implementing a *robust, production-ready* utility function `getFeatureState(featureName)`. The entire application will make use of this function, and we intend this function to be used as a central place for feature management in our codebase. Here are some examples of intended usages:

```
// src/feature-x/summary.js
getFeatureState("extended-summary")
  .then(function(isEnabled) {
    if (isEnabled) {
      showExtendedSummary();
    } else {
      showBriefSummary();
    }
  });

// src/feature-y/feedback-dialog.js
getFeatureState("feedback-dialog")
  .then(function(isEnabled) {
    if (isEnabled) {
      makeFeedbackButtonVisible();
    }
  });
```