9.4 Conclusion

We've covered a lot of ground in the last three chapters, transforming our promising but unformed application into a site capable of the full suite of signup and login behaviors. All that is needed to complete the authentication functionality is to restrict access to pages based on login status and user identity. We'll accomplish this task en route to giving users the ability to edit their information, which is the main goal of Chapter 10.

Before moving on, merge your changes back into the master branch:

```
$ rails test
$ git add -A
$ git commit -m "Implement advanced login"
$ git checkout master
$ git merge advanced-login
$ git push
```

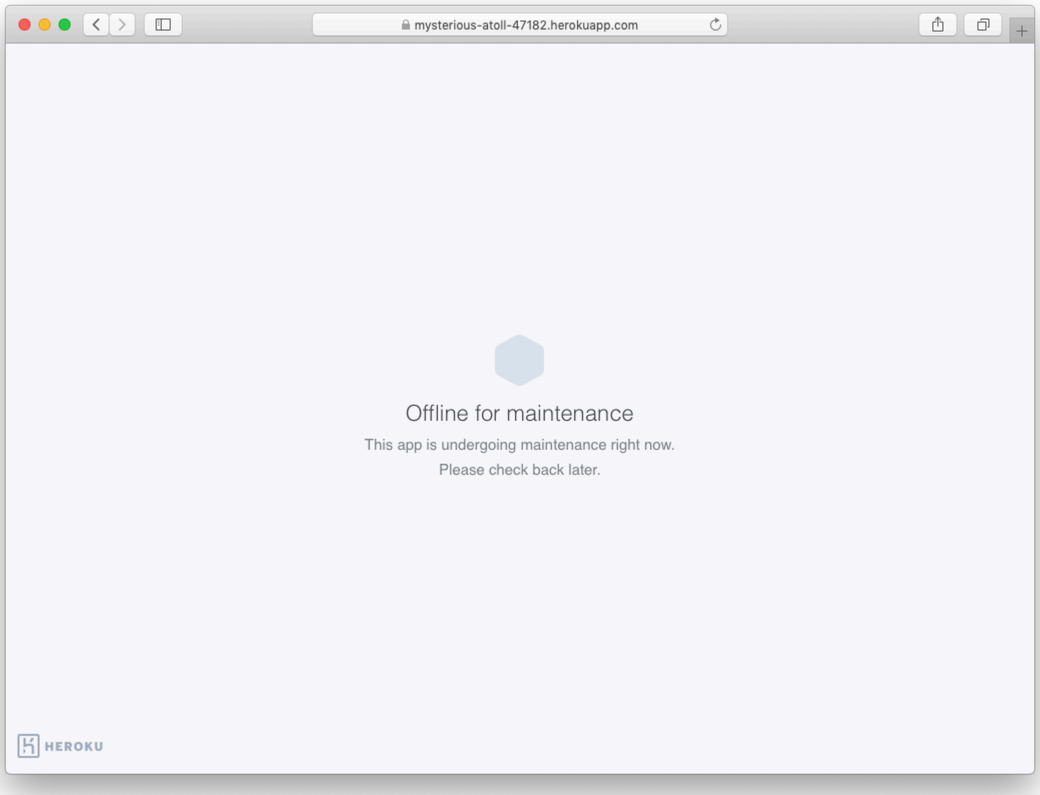
Before deploying to Heroku, it's worth noting that the application will briefly be in an invalid state after pushing but before the migration is finished. On a production site with significant traffic, it's a good idea to turn *maintenance mode* on before making the changes:

```
$ heroku maintenance:on
$ git push heroku
$ heroku run rails db:migrate
$ heroku maintenance:off
```

This arranges to show a standard error page during the deployment and migration (Figure 9.5). (We won't bother with this step again, but it's good to see it at least once.) For more information, see the Heroku documentation on error pages.

9.4.1 What we learned in this chapter

- Rails can maintain state from one page to the next using persistent cookies via the **`cookies`** method.



Offline for maintenance

This app is undergoing maintenance right now.
Please check back later.

Figure 9.5: The production app in maintenance mode.

- We associate to each user a remember token and a corresponding remember digest for use in persistent sessions.
- Using the `cookies` method, we create a persistent session by placing a permanent remember token cookie on the browser.
- Login status is determined by the presence of a current user based on the temporary session's user id or the permanent session's unique remember token.
- The application signs users out by deleting the session's user id and removing the permanent cookie from the browser.
- The ternary operator is a compact way to write simple if-then statements.