

GAME STORE

FRONT END

public part (accessible without authentication)

private part (available for registered users)

Use any framework of your choice for the front-end

Optional, not required

KendoUI, AngularJS, Angular 2, Knockout, Bootstrap, etc...

Implement responsive design

It may be based on Bootstrap, Materialize or any other UI framework

Use at least one AJAX form and/or WebSockets communication

Apply error handling and data validation to avoid crashes when invalid data is entered

Use loaders, modals and notifications when applicable

Prevent yourself from security holes (XSS, XSRF, Parameter Tampering, etc.)

Handle correctly the special HTML characters and tags like `<script>`, `
`, etc.

Create usable UI

No need to be pretty, but usable

BACK END

DataBase

NodeJS + Express + Pug -> DONE

At least 5 different public dynamic web pages

Using Pug

At least 3 different private (authenticated) dynamic web pages

Using Pug

At least 5 different public RESTful routes for AJAX

At least 1 private (authenticated) route for AJAX

MongoDB

Create a data/service layer for accessing the database

Use Passport - for managing users

Implement WebSockets

- Using Socket.io or anything else