**ADS Listing System v 0.1.0**

The **Ads Listings Site** (ALS) provides users the ability to view and create a listing by category and location. Users can register and login to view advertisements or create their own.

The system will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + hapi*** as backend technologies and MongoDB as database.

**Project Configuration**

Please choose the MongoDB data folder and specify it in *package.json* in scripts section, *'mongo'* script. By default it uses Z:/mongo-data folder.

**Runnung The Demo**

In order to start the project instal latest *Node.js* and from console run:

npm install

npm start

Alternatively you can selectively start the individual application components:

* MongoDB - npm run mongo
* node.js + hapi.js serrvices - npm run services (if you don't need automatic server restarting on change - use node instead of nodemon in the script)
* webpack-dev-server with HMR (Hot Module Replacement) - npm run webpack

**Project Decription and Main Components**

Application provides following functionality:

* *Anonymous users* can only view all advertisement without registration.
* *Registered User (*extends *Anonymous User)* – can login/edit profile and be able to create/update/delete his own advertisement.
* *Instructors* can create tests and see the students' test results.
* *Administrator (*extends *Registered User) –* can manage (create, edit and delete user data) all *Registered Users*, as well as manage category.

JavaScript (ECMAScript 6) client part is available in /app folder. It uses *Webpack* and *webpack-dev-server* with HMR (Hot Module Replacement). Configuration is specified in webpack.config.js.

Client side application is developed as *Singe Page App (SPA)*. The app entry point is index.js, which imports *react-router* configuration specified in main-router.jsx. The top-level application component is ads-listing-system.jsx (React JSX + ES6).

Server side is implemented using *hapi.js (*[*http://hapijs.com/*](http://hapijs.com/)*)* *Node* framework and resides in /ads\_api forlder. The ads API is proxied by webpack-dev-server configuration to port 9000 (configurable). Main server class is server.js.

Two main features are implemented both on client and server side in this version of the project:

* User management
* Advertisement management

The server side implementation for two features are in user.routes.js and handlersUser.js (for User management), ads.routes.js and handlersAds (for Advertisement management) and category.routes.js and handlersCategory.js (for Category management).

Client side implementation uses React ES6 components and resides id /app/componets/views/users, and /app/componets/views/ads accordingly.