## Selected Requirements

|  |  |  |
| --- | --- | --- |
| Requirement | Points | Adaption |
| **Using React.js front-end** | **15** | **Using Angular2 front-end** |
| Using redux and an advanced React architecture | 5 | Using of global services as data management |
| **Running your application in a Docker container (or having the application Docker compatible)** | **5** |  |
| Running your application in Rahti | 5 |  |
| Having more than one container (such as a separate database server or a load balancer) | 3 | Additional Database container |
| **Using a database, such as Mongo, Redis, or any SQL-compatible** | **5** | **Using MongoDB** |
| Use an ORM and models in backend, such as the Mongoose (MongoDB) or Sequelize (SQL) | 3 | Using Mongoose |
| **Supporting pictures storage and display** | **5** |  |
| Resize and validate the images after download - create a set of thumbnails on loading to be shown in different places | 3 |  |
| **Using XHR (also know as AJAX/AJAJ) for data transfer and frontend syncronization** | **4** | **Using Angular2 Http-Client** |
| **User registration, authentication, and password storage** | **5** | **User management + Authentication via JWT** |
| Add access permissions: blog is open to everyone, to registered users, to specific users, to owner only | 2 |  |
| **Provide data from your application through an API and document it with a suitable tool, e.g. with apiDoc or API Blueprint.** | **3** |  |
|  |  |  |
|  | 63 pts |  |

## Libraries

|  |  |
| --- | --- |
| Library | Use |
| cors | Allow request to other Servers |
| express | Webserver Framework |
| express-validator | Parameter validation |
| bcryptjs | Hashing of Passwords |
| jsonwebtoken | Generation of JWT |
| mongoose | Database ORM |
| passport | User authorization |
| passport-jwt | User authorization by JWT |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Development Library | Use |
| dotenv | Emulate Environment Variables |
| nodemon | Auto reload of Server after changes |
|  |  |