# Node.js – Exam – October 2014

## Telerik Academy Events

You are assigned to design and implement an event application for Telerik Academy students. The students should be able to register into the application and create events.

Students have user profiles containing information about them – **first** and **last name**, **phone number**, **email** **address**, **Telerik Academy initiatives** and **seasons**, a **profile image** and **links to their Facebook/Twitter/LinkedIn/Google+** profiles. The phone number and the links to other profiles are optional. If a profile image is not provided, assign a default one (use the ninja ☺ ).

Events can be created only by registered users. When an event is created, its creator provides a **title**, **a brief description**, **location**, **category** and the creator's name and phone number are filled automatically (if the creator has not provided a phone number, s/he cannot create an event). Categories can be "Academy initiative", "Free time", etc… The location is defined using coordinates in the format (Latitude, Longitude).

Events can be either **public**, **initiative-based** or **initiative-and-season-based**:

* Public events can be joined by any registered user
* Initiative-based events can be joined only by those that are/were in the initiative (no matter the season)
  + For example an event marked as initiative-based "Software Academy” can be joined only by students that are/were in the **Software Academy initiative**, yet in any of the seasons
  + Student can create such an event, only if s/he also fulfills the criteria
* Initiative-and-season-based events can be joined only by students from the concrete initiative and season can join the event
  + For example an event for the initiative "Software Academy" and season "started 2013" can be joined only by students are/were in the **initiative Software Academy** and **season 2013**
  + Student can create such an event, only if s/he also fulfills the criteria

When the event is created, it is visible to the corresponding group and each student, that fits the criteria, can join the event.

An event is active only if one of the following conditions are met:

* The event is not deleted
* The current date is before the date of the event

When the event is past (i.e. its date has expired) the joined in the event students can evaluate it based on the criteria:

* Organization (up to 5 points)
* Venue (up to 5 points)

The creator of the event immediately receives points, based on the evaluation.

The system should be implemented as a server-side web application in Node.js using Jade view engine.

### Data Layer (21 points)

* Use **Mongoose** as ODM engine and **MongoDB** as database storage engine.

2 points

* **Data layer abstraction** – the data layer should be implemented as an abstract module.

7 points

Design a simple data layer to hold **users** and **events**.

* Each **user** has **username**, **password**, **event points** (Organization and Venue points), **first** and **last** **names**, **phone** **number**, **email** **address**, **Telerik Academy initiatives** and **seasons**, **a profile image** and links to their Facebook/Twitter/LinkedIn/Google+ profiles.
  + The password should be stored in the DB encrypted (not as plain text)
  + Facebook/Twitter/LinkedIn/Google+ profiles are optional
  + The phone number is optional
  + A student can be part of many initiatives and in different season
    - i.e. Peter Petrov can be part of School Academy Season 2011m School Academy Season 2012 and Software Academy Season 2013

3 point

* + Use the **Passport** system to keep the users and their encrypted passwords.

2 point

* Each **event** has a **title**, a **brief description**, **location**, **category**, **type**, its **creator's name** and **phone number** and a list of comments by users
  + Location is optional
  + Categories must be predefined and not editable by users (choose them wisely)
  + A type is constructed by two parts – initiative and season
    - The initiatives are "Software Academy", "Algo Academy", "School Academy" and "Kids Academy". They are constant and should be used as is
    - The seasons are as follows – "Started 2010", "Started 2011", "Started 2012" and "Started 2013"
  + Comments can be left only by joined-in-the-event-users

4 points

* Fill some **sample data** in the DB to simplify any further testing.

2 points

* Your **project should run after "copy/paste" deployment**, without changing connection strings or other settings

1 point

### Node.js Application – Common Features (21 points)

* Use proper application architecture – separation of concerns and high quality code should be used

5 point

* **Layout** – design a layout page to reuse the common page elements like headers and footers and navigation in all other pages in the project.

2 point

* In the **navigation** when user is not authorized add link only to the home page, register and login pages.

For registered users add link to **active** (upcoming) events, **past** events, **creating events**.

3 points

* **Configure the Passport module** to enable user management functionality (login / logout).

4 points

* + The username should be between 6 and 20 characters long and can contain **Latin letters**, **digits** and the symbols **'\_'** (underscore), **' '** (space) and **'.'** (dot)

2 points

* **Error handling** – in case of errors (e.g. DB connection lost, incorrect request, etc.), an appropriate error message should be displayed. You are free to decide how exactly.

3 points

* **User interface (UI)** – the user interface should be usable enough. It is not needed to be beautiful. Use Bootstrap if you want.

2 points

### Node.js Application – Public Area (18 points)

* **Home page** – at the application start page display statistics – passed events with their evaluation

3 points

* + **Cache** the statistics for 10 minutes

5 points

* **Register** – registering new user

5 points

* **Login** – login a registered user

5 points

### User Area (40 points)

* **Registered users** should be able to **create, join or leave events**
  + **List** activeevents for the concrete user by categories

2 points

* + - Sort the events by date (the closest date at the top)

1 points

* + - Use server side paging (page size is 10)

2 points

* + **Create** an event

2 points

* + - Validate the input

2 points

* + **View** event details. Show the title, location, comments, the creator of the event and their stats (result of the previous event evaluations), type, date, etc…

4 points

* + **Join** a non-joined-by-me active event. The user should fulfill the criteria to join the event

4 points

* + **Leave** a joined-by-me event

4 points

* + **Leave a comment for a** joined event

5 points

* **Profile page** – registered users (after login) should be able to see their profile and change the information
  + Change profile image

2 points

* + Add phone number

2 points

* + Add linked profile (Google+, Facebook, etc…)

2 points

* + List the created by the user events

2 points

* + List the joined by the user events

2 points

* + List passed events of the user

2 points

* + Use server side paging (page size 10).

2 points

## Evaluation Criteria

The evaluation criteria include: correct and complete fulfillment of the requirements; good technical design and appropriate use of technologies; high-quality code (correctness, readability, maintainability).

## Other Terms

During the exam you are allowed to use any teaching materials, lectures, books, existing source code, and other paper or Internet resources. Direct or indirect communication with anybody in class or outside is forbidden. This includes but does not limit to technical conversations with other students, using mobile phones, chat software (Skype, ICQ, etc.), email, forum posts, USB flash drives etc.

## Exam Duration

Students are allowed to work up to **8 hours**.