

AutoAdvisor

David Wemhoener, Ki Yung Chan, Hui Chen, Dan Stubbs, Nicholas Conlon, Peng Gao, Wei Lin, Duane Bailey, Kyle Hughes

AutoAdvisor

David Wemhoener, Ki Yung Chan, Hui Chen, Dan Stubbs, Nicholas
Conlon, Peng Gao, Wei Lin, Duane Bailey, Kyle Hughes

AutoAdvisor 1.0 Release Notes

1 May 2013

CS Rocks Inc.

Production URL

autoadvisor.elastickbeanstalk.com

Source code repository

The AutoAdvisor source repository is located at the AutoAdvisor github site:

<https://github.com/Auto-Advisor/autoadvisor>

Assumptions

Use of Linux-style operating system. Also familiarity with the command line, databases, gitHub, and Amazon AWS.

Requirements

git
ruby 1.9
nodejs
mysql 5.5

Build instructions

One-line build instructions for Ubuntu 12.10:

```
wget https://raw.githubusercontent.com/Auto-Advisor/autoadvisor/master/misc/ubuntu_boot.sh -O  
./ubuntu_boot.sh && bash ./ubuntu_boot.sh
```

This line will install all the requirements for a full development installation of the application, including mysql, git, and elastic beanstalk tools

Repository clone/checkout:

AutoAdvisor is hosted as a public repository on Github, anyone can check out the code with:

```
git clone https://github.com/Auto-Advisor/autoadvisor.git
```

Ensure you have Ruby on Rails installed:

```
gem install rails
```

Inside the root of the newly checked out source code, install the remaining ruby requirements for the project. This requires a c and c++ compiler. Make sure that you have installed the dev packages for mysqlclient and sqlite3:

```
bundle install
```

MySQL configuration:

Edit the MySQL configuration via `/etc/my.cnf`, but you shouldn't *need* to on a default mysql install. Here are the necessary bits to get mysql working locally:

user: root

password: password

host: localhost

port: 3306

If you are running a default install of mysql, this sets the password for root:

```
bash$ mysql -uroot # -pPASSWORD if password is set, no space after the '-p'
```

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('password');
```

```
Query OK, 0 rows affected (0.04 sec)
```

First use:

When first setting up the app, or when recreating the database, use:

```
rake db:setup
```

This will create the database, run any migrations, and load it with seed data.

Database maintenance:

Whenever checking code in, or after creating any database migration, please use:

```
rake db:migrate
```

This will keep the database schema consistent with the schema in the source.

Working features

Registration/Sign Up

Account Log in

Personal profile

Upload and parse unofficial transcript

Schedule recommendations

Create a Schedule

Known issues

For a full list of known issues, see the AutoAdvisor issue tracker at:

<https://github.com/Auto-Advisor/autoadvisor/issues>

Change logs

Change logs for the project can be found in the AutoAdvisor github commit history at:

<https://github.com/Auto-Advisor/autoadvisor/commits/master>

Documentation

The 1.0 release come with four documents (software requirements specification, software design specification, user manual, and admin manual). All documents can also be found on the AutoAdvisor github wiki page under 'Final 'release': <https://github.com/Auto-Advisor/autoadvisor/wiki>

Issue tracking

Issue tracking is through the gitHub issue tracker on the AutoAdvisor github page. A gitHub account is required for adding issues to the tracker. The plan of use for GitHub's issue tracking system is as follows. Any developer who finds a UI problem, bug, or any other issue must document it as soon as possible on the issue tracker. The tracker requires the following:

- A short descriptive title.
- A developer assigned (if necessary).
- A milestone for resolution (alpha, beta, ect).
- A priority (bug, feature request, security vulnerability, ect).
- A more in-depth description of the issue.

All developers working on the issue should leave comments in the appropriate section of the specific issue to help with later documentation. The issue tracker will be gone over as a team during meeting to make sure important issues are attended to. For more information see:

<https://github.com/Auto-Advisor/autoadvisor/wiki>

Testing information

The project takes advantage of the automated testing mechanism built into Ruby on Rails. Blank test files are automatically generated during development and are organized in the /test/ folder. Unit tests cover models, functional tests cover controllers, integration and performance tests cover the combination of models, views, and controllers. Tests are automated and all tests can be run from the command line with the command 'rake test'. For more information see the test package information: <https://github.com/Auto-Advisor/autoadvisor/wiki/Test-package-information>