CS 350 Database Systems Term Project Part 3:
Design and Implementation

Semih Yılmaz
Ahmet Ensar Köprülü

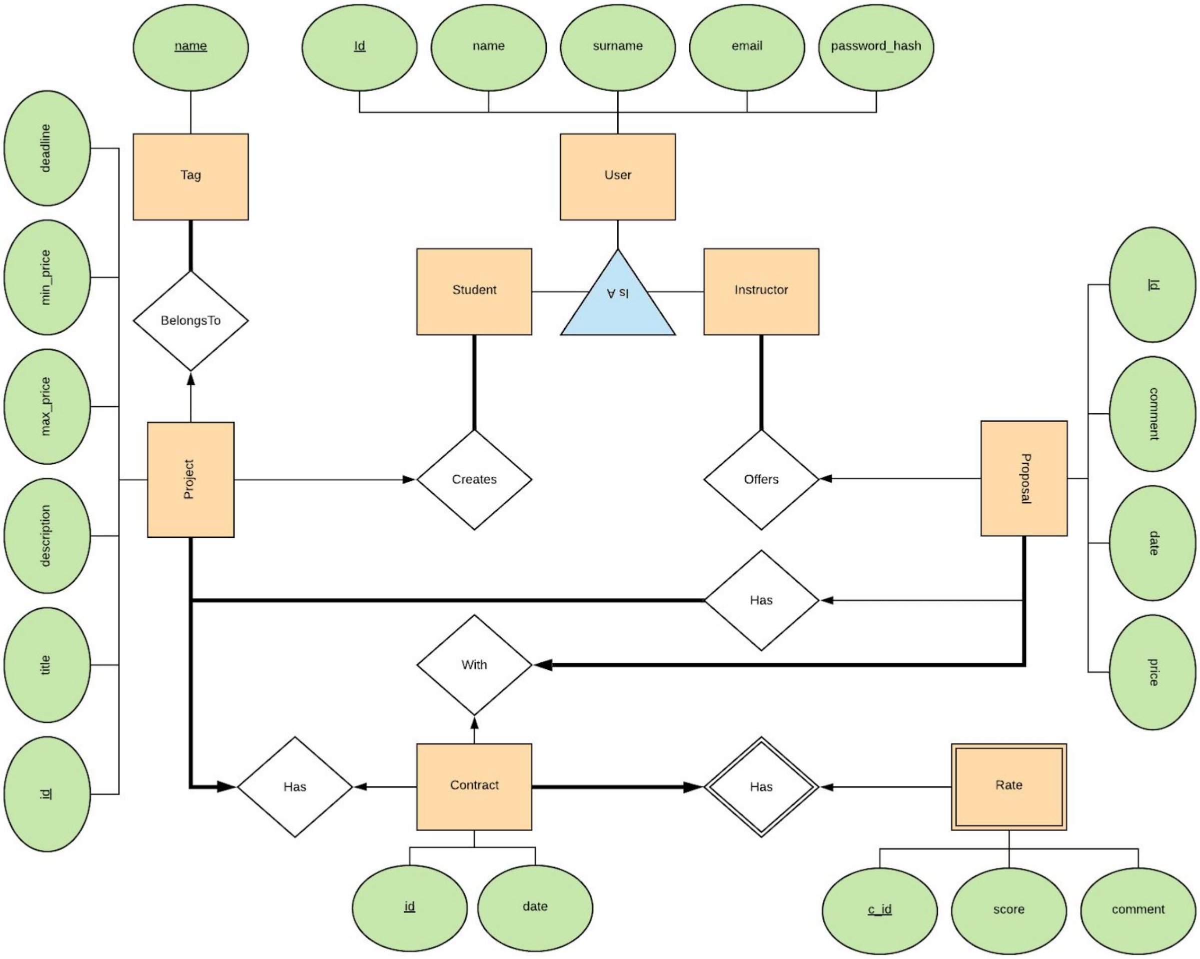
1. Motivation and Requirements:

Most of the teachers who is not employed by the school or any organization working “freelance” and researching market by themself. We believe exist applications that let people to search for works is not capable to serve for this purpose since, they are designed for remote works but teaching is a process and needs more interaction than a normal job. Moreover, students who want to find supervisor for a specific subject or instructor for a lecture, has no much online solution.

Features:

- Students will be able to create a course which may include a lecture or a homework in order to find a supervisor or a instructor.
- Instructors able to filter courses or homeworks by according to their interest.
- Instructors will be able to offer themself to that created opportunities.
- Students able to search about instructors that offered to their project, by looking their previous works and references messages.
- Students will be able to select best option and have contract with the selected instructor.
- After a contract fulfil students will write his thoughts about instructor to rate instructor for other students.

2. Conceptual Database Design:

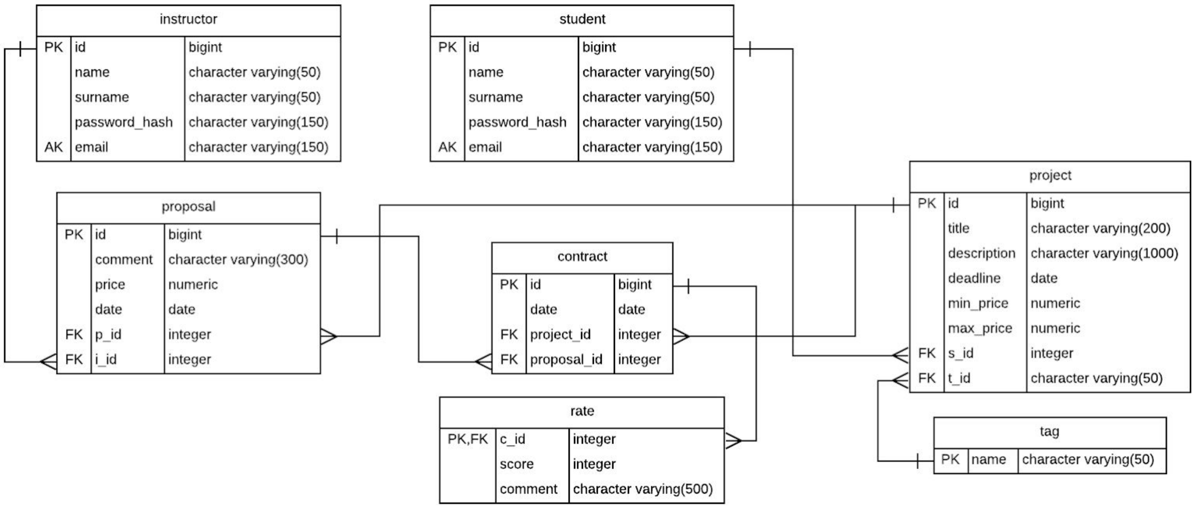


Data Requirements:

- Each student has id, name, surname, email, password_hash.
- Each instructor has id, name, surname, email, password_hash.
- Each project has id, title, description, max_price, min_price, deadline.
- Each proposal has id, comment, date, price.
- Each contract has id, date.
- Each rate has c_id, score, comment.
- Each tag has name.
- Students can create zero or many projects.
- Instructor can offer zero or many proposal..
- Instructor can offer one proposal to one project.
- Project can has zero or many proposal
- Project can has zero or one contract.

- Project has a tag.
- Tag can has zero or many project.
- Contract can has one proposal and one project.
- Contract can has zero or one rate..
- Rate cannot exist without contract.

3. Logical Database Design:



instructor
PK id bigint
name character varying(50)
surname character varying(50)
password_hash character varying(150)
AK email character varying(150)
student
PK id bigint
name character varying(50)
surname character varying(50)
password_hash character varying(150)
AK email character varying(150)
proposal
PK id bigint
comment character varying(300)
price numeric
date date
FK p_id integer
FK i_id integer
contract
PK id bigint
date date
FK project_id integer
FK proposal_id integer
project
PK id bigint
title character varying(200)
description character varying(1000)
deadline date
min_price numeric
max_price numeric
FK s_id integer
FK t_id character varying(50)
rate
PK,FK c_id integer
score integer
comment character varying(500)
tag
PK name character varying(50)

Functional Dependencies:

For student table:

id → {id, name, surname, email, password_hash}

email → {id, name, surname, email, password_hash}

For instructor table:

id → {id, name, surname, email, password_hash}

email → {id, name, surname, email, password_hash}

For project table:

id → {id, title, description, max_price, min_price, deadline, s_id, t_id}

For proposal table:

id → {id, comment, date, price, i_id, p_id}

{i_id, p_id} → {id, comment, date, price, i_id, p_id}

For contract table:

id → {id, date, project_id, proposal_id}

{project_id, proposal_id} → {id, date, project_id, proposal_id}

For rate table:

c_id → {c_id, score, comment}

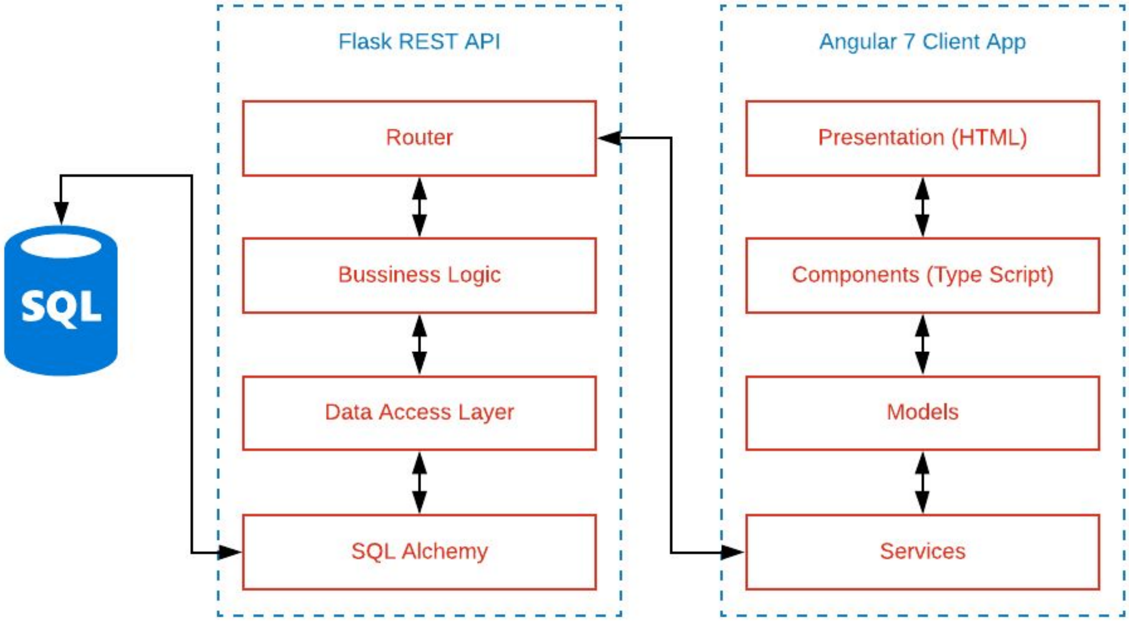
For tag table:

name → {name}

Normalization:

All the schemas are in Boyce-Codd Normal Form since: All the functional dependencies dependent to super or candidate key and they uniquely identifying the all attributes/ whole schema. Also, all the functional dependencies are non-trivial.

4. Application Design and Implementation:



The whole project consist of 3 tier; Database, Backend and Frontend. In database layer, PostgreSQL is used as DBMS. In backend, Flask REST API is running. SQLAlchemy is used for database connection and executing queries by api. All the queries are created in Data Access Layer. Business Layer is responsible for validations, data transformations and other high level processes. Router is responsible for incoming requests. In frontend, Angular 7 application is running. Services are bridge between frontend and backend. All requests are send by services and return json objects. Models are TypeScript classes that represent json objects returned from api. Components are logic behind presentation, all the html pages. Responsible for form datas, validations, management of data that returned from services. Presentation layer is html pages that users will interact with.

Tech-Stack: PostgreSQL, Python, SQLAlchemy, TypeScript, JavaScript, Html

Queries:

Semantic: Get instructor with the specified id.

```
SELECT *
FROM instructor
WHERE id = {id}
```

Semantic: Register instructor.

```
INSERT INTO instructor(name, surname, email, password_hash)
VALUES('{name}', '{surname}', '{email}', '{password_hash}')
```

Semantic: Get instructor with specified credentials for login.

```
SELECT *
FROM instructor
WHERE email = '{email}'
 AND password_hash = '{password_hash}'
```

Semantic: Update instructor informations with the specified id.

```
UPDATE instructor
SET name='{name}', surname='{surname}',
password_hash='{password_hash}', email='{email}'
WHERE id = {id}
```

Semantic: Get studentwith the specified id.

```
INSERT INTO student(name, surname, email, password_hash)
VALUES('{name}', '{surname}', '{email}', '{password_hash}')
```

Semantic: Get student with specified credentials for login.

```
SELECT *
FROM student
WHERE email = '{email}'
 AND password_hash = '{password_hash}'
```

Semantic: Update student informations with the specified id.

```
UPDATE student
SET name='{name}', surname='{surname}',
password_hash='{password_hash}', email='{email}'
WHERE id = {id}
```

Semantic: Create project.

```
INSERT INTO project (title, description, deadline, max_price,
min_price, s_id, t_id)
VALUES ('{title}', '{description}', '{deadline}', {max_price},
{min_price}, {s_id}, '{t_id}')
```

Semantic: List all projects.

```
SELECT *
FROM project
```

Semantic: List contracted projects of student with the specified id, its contract and its rate and instructor.

```
SELECT instructor.name, instructor.surname, project.id,
      title, deadline, t_id,
      proposal.id AS p_id, price,
      contract.id AS c_id,
      contract.date, rate.score, rate.comment
FROM project, proposal, instructor, contract
LEFT JOIN rate ON rate.c_id = contract.id
WHERE s_id = {s_id}
 AND p_id = project.id
 AND proposal.i_id = instructor.id
 AND contract.project_id = project.id
 AND contract.proposal_id = proposal.id
```

Semantic: List projects belongs to student with the specified id and not contracted yet.

```
SELECT *
FROM project
WHERE s_id = {s_id}
 AND id NOT IN (SELECT project_id
                FROM contract)
```

Semantic: List projects not contracted yet.

```
SELECT *
FROM project
WHERE id NOT IN (SELECT project_id
                 FROM contract)
```

Semantic: List contracted projects of instructor with the specified id., its contract and its rate and project owner student

```
SELECT student.name, student.surname, project.id,
       title, deadline, t_id,
       proposal.id AS p_id, price,
       contract.id AS c_id, contract.date, rate.score, rate.comment
FROM proposal,student, project, contract
   LEFT JOIN rate ON rate.c_id = contract.id
WHERE i_id = {i_id}
 AND p_id = project.id
 AND project.s_id = student.id
 AND contract.project_id = project.id
 AND contract.proposal_id = proposal.id
```

Semantic: Get project with the specified id

```
SELECT *
FROM project
WHERE id = '{p_id}'
```

Semantic : Update project with the specified id

```
UPDATE project
SET title='{title}', description='{description}',
deadline='{deadline}', min_price={min_price}, max_price={max_price}
WHERE id = {id}
```

Semantic: Delete project with the specified id.

```
DELETE FROM project
WHERE id = {id}
```

Semantic: Create proposal.

```
INSERT INTO proposal (comment, price, date, p_id, i_id)
VALUES('{comment}', {price}, '{date}', {p_id}, {i_id})
```

Semantic: Update proposal with the specified id.

```
UPDATE proposal
SET comment='{comment}', price={price}
WHERE id = {id}
DELETE FROM proposal
WHERE id = {id}
```

Semantic: List all proposal belongs to project with the specified id, and its instructors' contracts count, and average of rates.

```
SELECT proposal.id, name, surname, proposal.comment, proposal.price,
proposal.date, proposal.i_id, proposal.p_id,
      (SELECT count(*) AS count
      FROM proposal AS p, contract AS c, rate AS r
      WHERE p.i_id = proposal.i_id
        AND c.project_id = p.p_id
        AND c.proposal_id = p.id
        AND r.c_id = c.id
      GROUP BY i_id),
      (SELECT avg(score) AS avg
      FROM proposal AS p, contract AS c, rate AS r
      WHERE p.i_id = proposal.i_id
        AND c.project_id = p.p_id
        AND c.proposal_id = p.id
        AND r.c_id = c.id
      GROUP BY i_id)
FROM proposal, instructor AS i
WHERE p_id = {id}
AND proposal.i_id = i.id
```

Semantic: Select all active(not selected by any student for contract) proposals of instructor with the specified id.

```
SELECT proposal.id, comment, price, date, p_id, title, deadline
FROM proposal, project
WHERE i_id = {i_id}
 AND p_id = project.id
 AND proposal.id NOT IN (SELECT proposal_id
                         FROM contract)
```

Semantic: Create contract.

```
INSERT INTO contract (date, project_id, proposal_id)
VALUES ('{date}', {project_id}, {proposal_id})
```

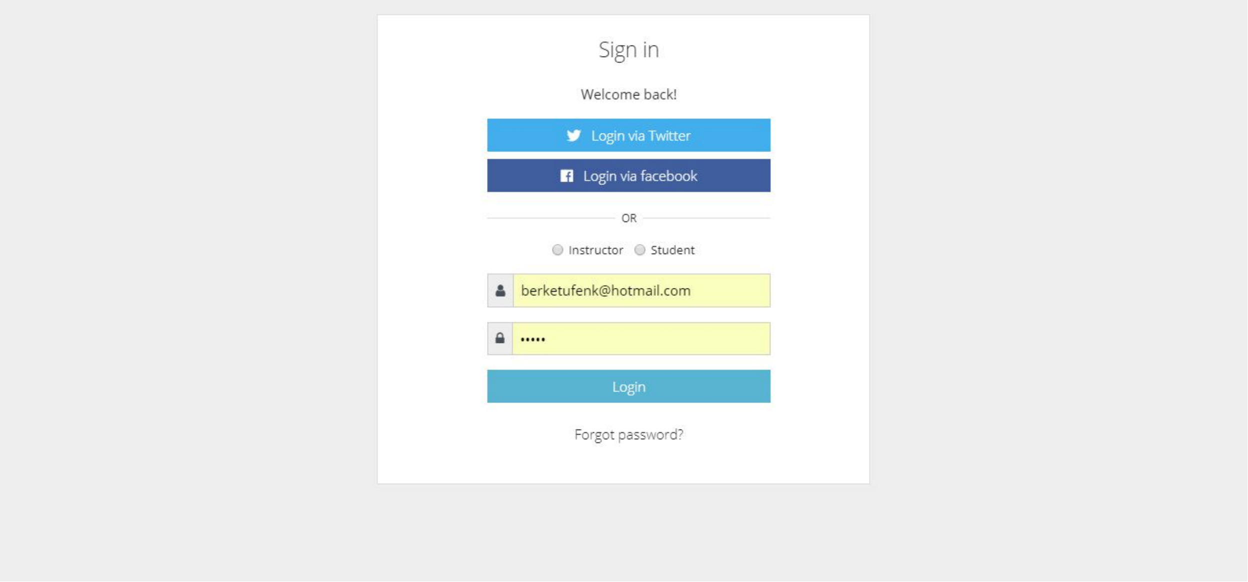
Semantic: Create rate.

```
INSERT INTO rate (c_id, score, comment)
VALUES ({c_id}, {score}, '{comment}')
```

Semantic: Update rate with the specified id.

```
UPDATE rate
SET score={score}, comment='{comment}'
WHERE c_id = {c_id}
```

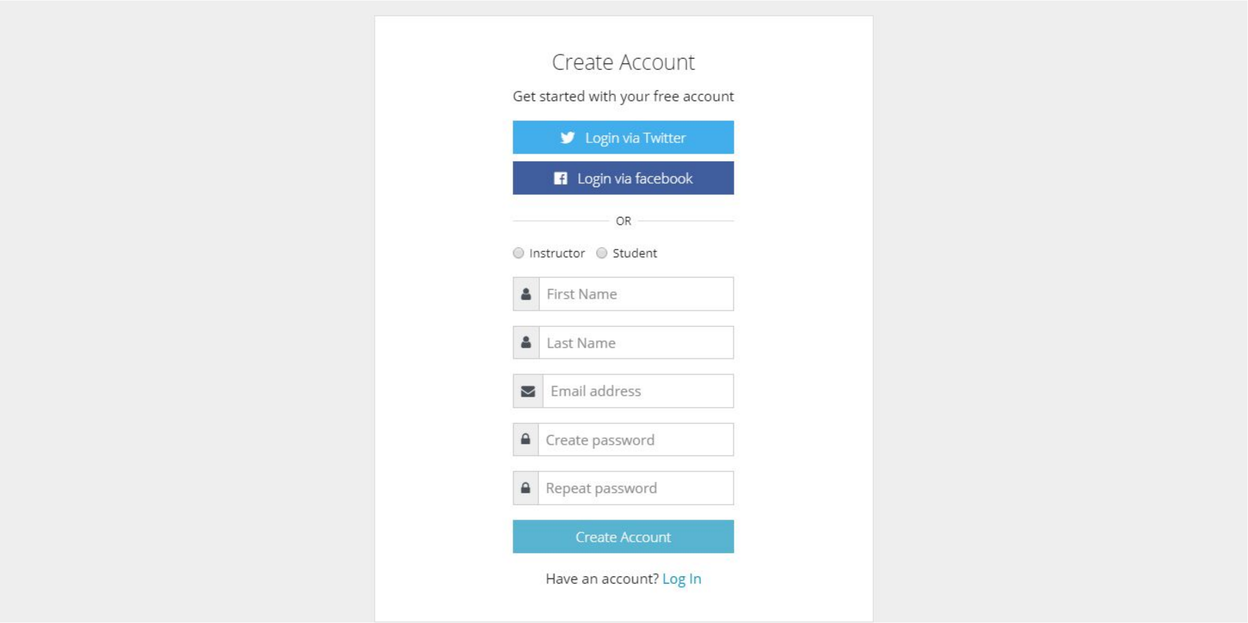
Screenshots:



Welcome back!

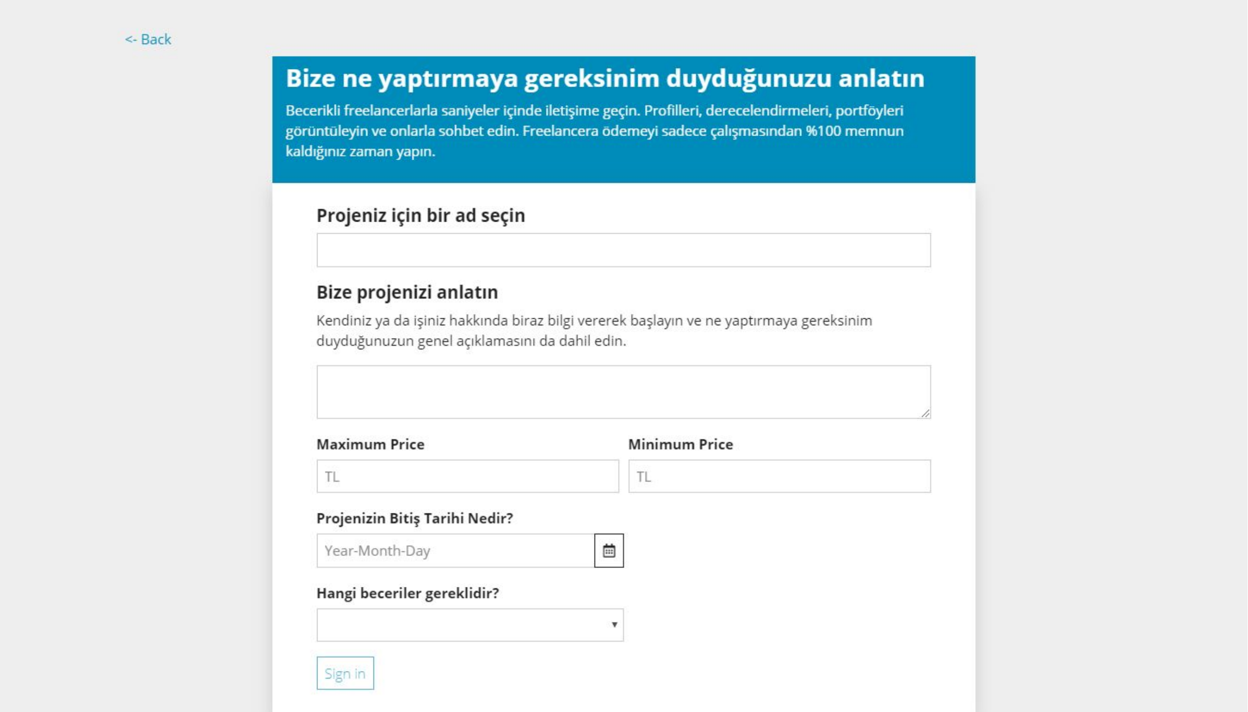
berketufenk@hotmail.com

••••••



Get started with your free account

Have an account? Log In



Bize ne yaptırmaya gereksinim duyduğunuzu anlatın

Becerikli freelancerlarla saniyeler içinde iletişime geçin. Profilleri, derecelendirmeleri, portföyleri görüntüleyin ve onlarla sohbet edin. Freelancera ödemeyi sadece çalışmasından %100 memnun kaldığınız zaman yapın.

Projeniz için bir ad seçin

Bize projenizi anlatın

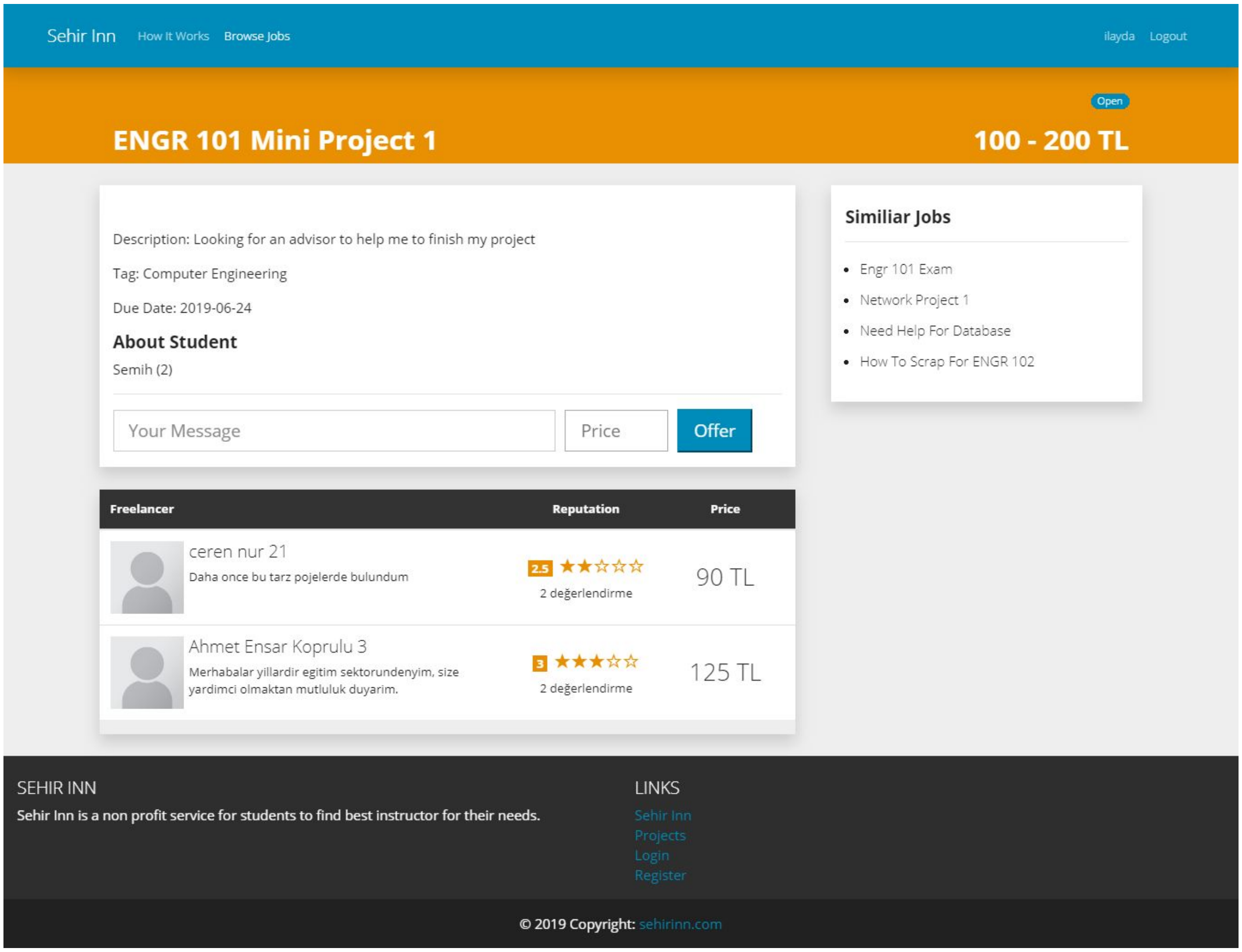
Kendiniz ya da işiniz hakkında biraz bilgi vererek başlayın ve ne yaptırmaya gereksinim duyduğunuzun genel açıklamasını da dahil edin.

Maximum Price	Minimum Price
TL	TL

Projenizin Bitiş Tarihi Nedir?

Year-Month-Day

Hangi beceriler gereklidir?



ENGR 101 Mini Project 1

100 - 200 TL

Description: Looking for an advisor to help me to finish my project

Tag: Computer Engineering

Due Date: 2019-06-24

About Student

Semih (2)

Similiar Jobs

- Engr 101 Exam
- Network Project 1
- Need Help For Database
- How To Scrap For ENGR 102

Freelancer	Reputation	Price
ceren nur 21 Daha once bu tarz pojelerde bulundum	2.5 ★★☆☆☆ 2 değerlendirme	90 TL
Ahmet Ensar Koprulu 3 Merhabalar yillardir egitim sektorundenyim, size yardimci olmaktan mutluluk duyarim.	3 ★★★☆☆ 2 değerlendirme	125 TL

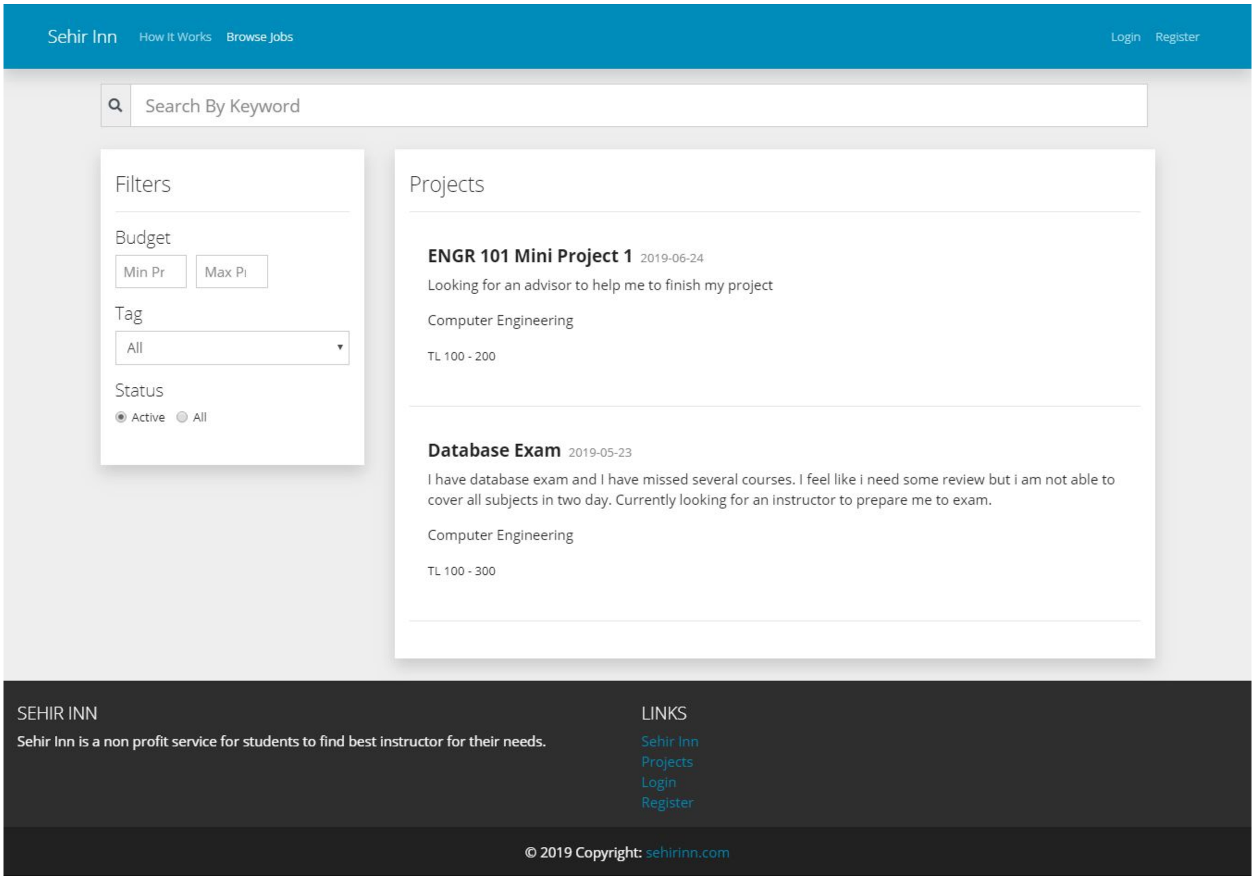
SEHIR INN

Sehir Inn is a non profit service for students to find best instructor for their needs.

LINKS

- Sehir Inn
- Projects
- Login
- Register

© 2019 Copyright: sehirinn.com



Projects

ENGR 101 Mini Project 1 2019-06-24

Looking for an advisor to help me to finish my project

Computer Engineering

TL 100 - 200

Database Exam 2019-05-23

I have database exam and I have missed several courses. I feel like i need some review but i am not able to cover all subjects in two day. Currently looking for an instructor to prepare me to exam.

Computer Engineering

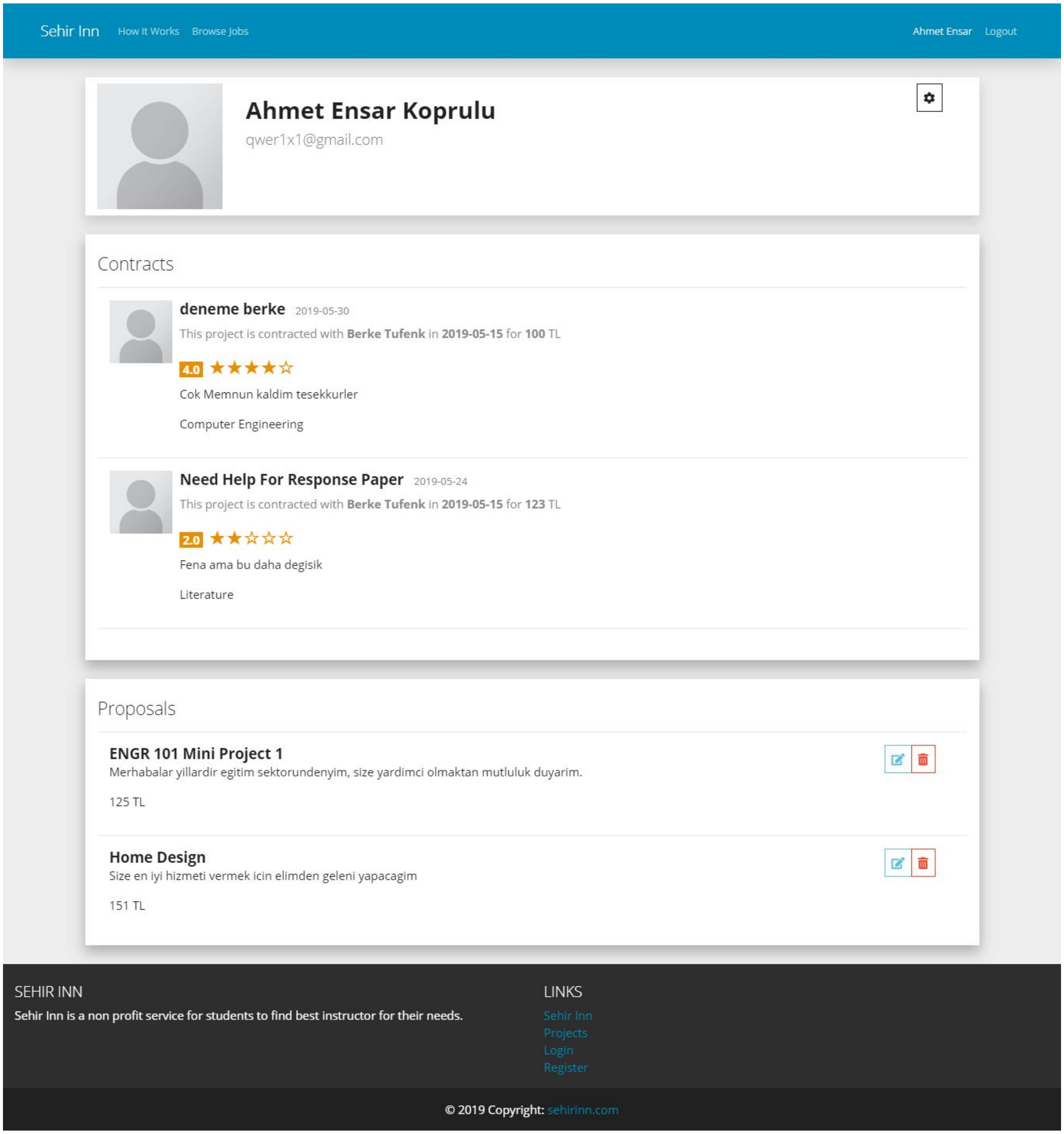
TL 100 - 300

SEHIR INN

Sehir Inn is a non profit service for students to find best instructor for their needs.

LINKS

© 2019 Copyright: sehirinn.com



Ahmet Ensar Koprulu

qwer1x1@gmail.com

Contracts

deneme berke 2019-05-30

This project is contracted with **Berke Tufenk** in **2019-05-15** for **100** TL

4.0 ★★★★☆

Cok Memnun kaldim tesekkurler

Computer Engineering

Need Help For Response Paper 2019-05-24

This project is contracted with **Berke Tufenk** in **2019-05-15** for **123** TL

2.0 ★★☆☆☆

Fena ama bu daha degisik

Literature

Proposals

ENGR 101 Mini Project 1

Merhabalar yillardir egitim sektorundenyim, size yardimci olmaktan mutluluk duyarim.

125 TL

Home Design

Size en iyi hizmeti vermek icin elimden geleni yapacagim

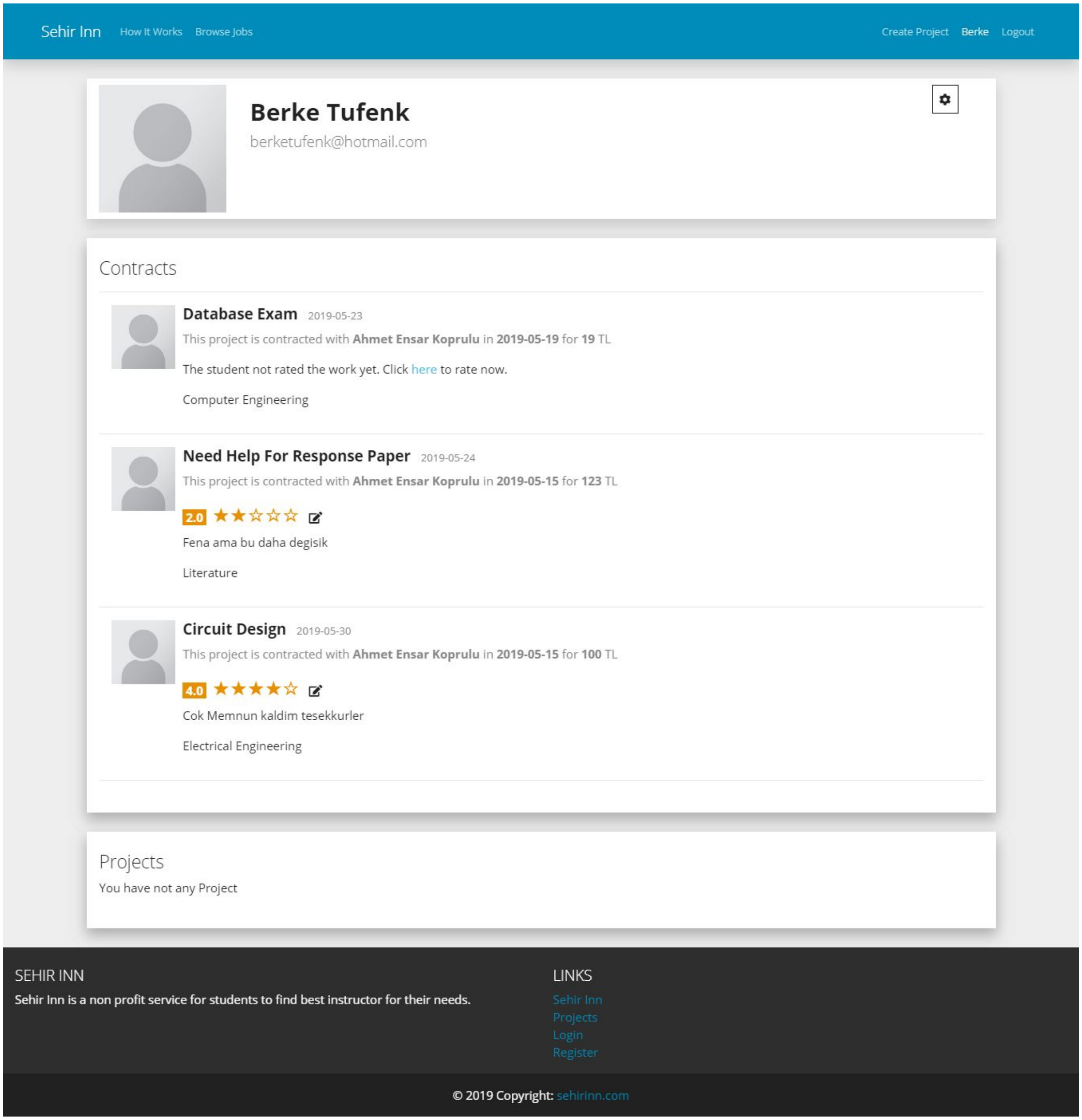
151 TL

SEHIR INN

Sehir Inn is a non profit service for students to find best instructor for their needs.

LINKS

© 2019 Copyright: sehirinn.com



Berke Tufenk

berketufenk@hotmail.com

Contracts

Database Exam 2019-05-23

This project is contracted with **Ahmet Ensar Koprulu** in **2019-05-19** for **19** TL

The student not rated the work yet. Click here to rate now.

Computer Engineering

Need Help For Response Paper 2019-05-24

This project is contracted with **Ahmet Ensar Koprulu** in **2019-05-15** for **123** TL

2.0 ★★☆☆☆

Fena ama bu daha degisik

Literature

Circuit Design 2019-05-30

This project is contracted with **Ahmet Ensar Koprulu** in **2019-05-15** for **100** TL

4.0 ★★★★☆

Cok Memnun kaldim tesekkurler

Electrical Engineering

Projects

You have not any Project

SEHIR INN

Sehir Inn is a non profit service for students to find best instructor for their needs.

LINKS

Sehir Inn
Projects
Login
Register

© 2019 Copyright: sehirinn.com