Part1: Create Express Framework with Given Directory Structure

=> create final_407410272 express framework

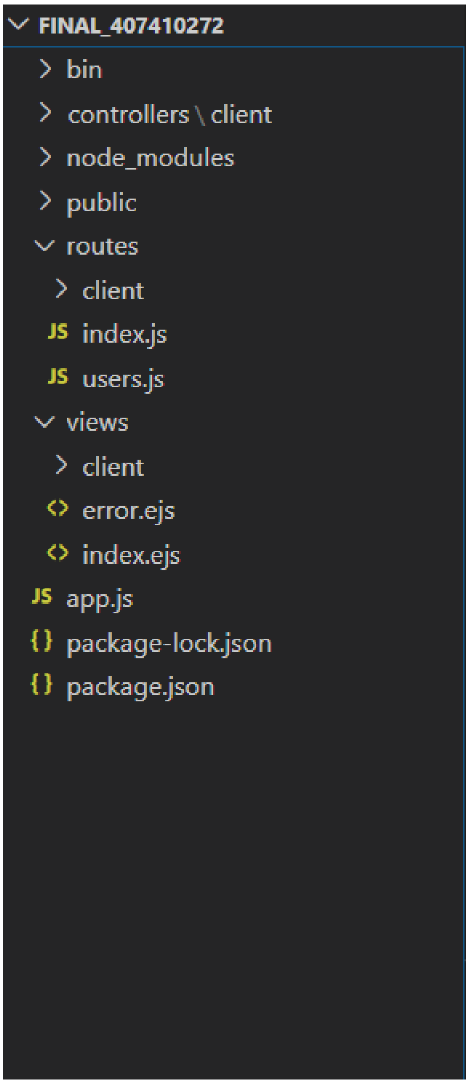
```
$ express --view=ejs final_407410272
$ cd final_407410272
$ code .
```

=> install node modules ,and change node to nodemon

```
$ npm install
$ npm install mysql2
```

```
// package.json

{
  "name": "final-407410272",
  "version": "0.0.0",
  "private": true,
  "scripts": {
    "start": "nodemon ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "morgan": "~1.9.1"
  }
}
```

=> create folders - models, controllers, controllers/client, routes/client, views/client



```
FINAL_407410272
  > bin
  > controllers \ client
  > node_modules
  > public
  v routes
    > client
    JS index.js
    JS users.js
  v views
    > client
    <> error.ejs
    <> index.ejs
  JS app.js
  {} package-lock.json
  {} package.json
```

Part 2 -- Make Lonely Work in new Express Framework

=> use VS code to open final_407410272 folder

=> use VS code to open lonely folder

=> copy files between two VS Code

=> npm start on final_407410272 in order to correct errors between moving files

=> delete all files in public folder

Copy Files From lonely to Final_407410272

(1)

utils/database.js(lonely) => utils/database.js (final_407410272)

(2)

models/aboutModel.js (lonely) => models/aboutModel.js (final_407410272)

(3)

controllers/aboutController.js (lonely) => controllers/client/aboutController.js (final_407410272)

(4)

routes/about.js (lonely) => routes/client/about.js (final_407410272)

(5)

views/about.ejs (lonely) => views/client/about.ejs(final_407410272)

(6)

public/* (lonely) => public/* (final_407410272)

* Change the path that does not match

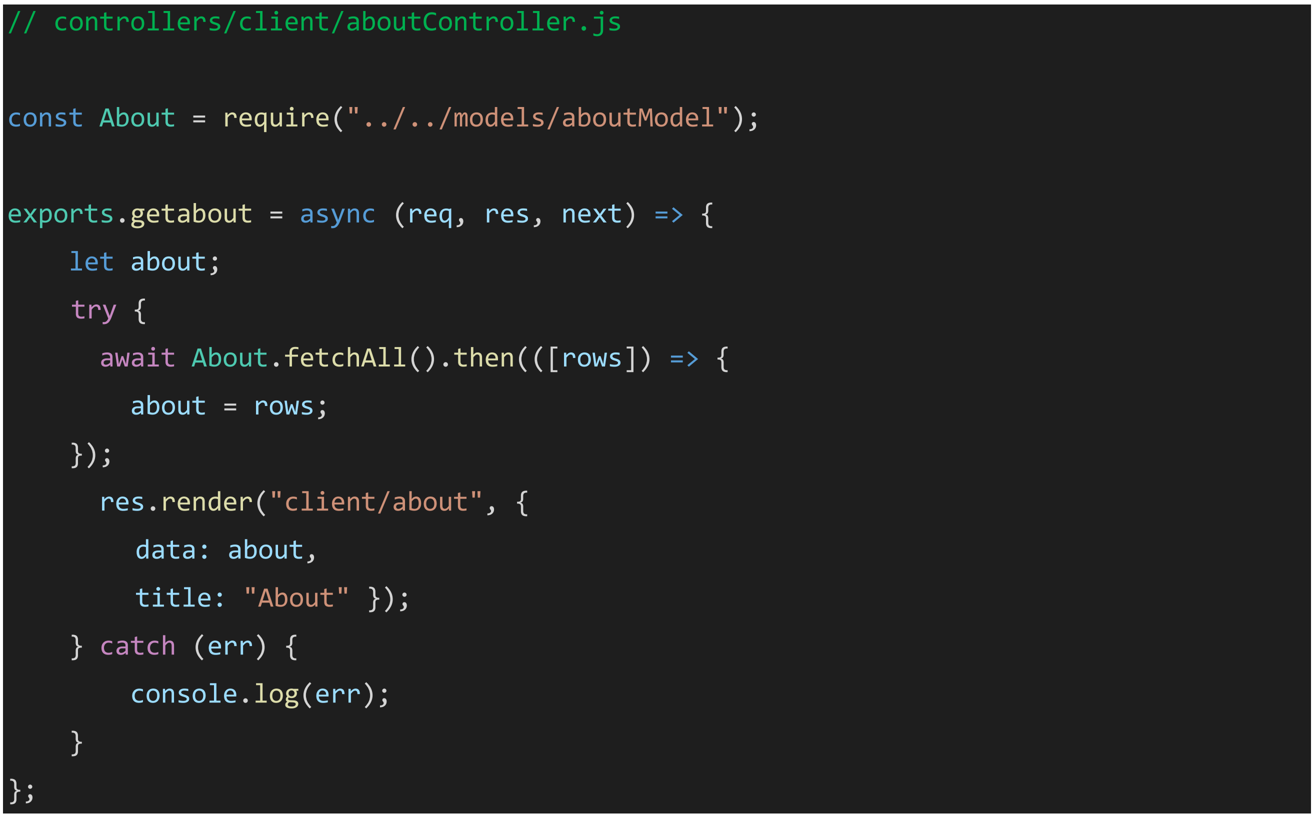
=> add /about route in app.js

```
// app.js

const lonelyRouter = require('./routes/client/lonely');
const aboutRouter = require('./routes/client/about');

app.use('/', indexRouter);
app.use('/users', usersRouter);
app.use('/lonely', lonelyRouter);
app.use('/about',aboutRouter);
```

=> change the path of aboutController.js

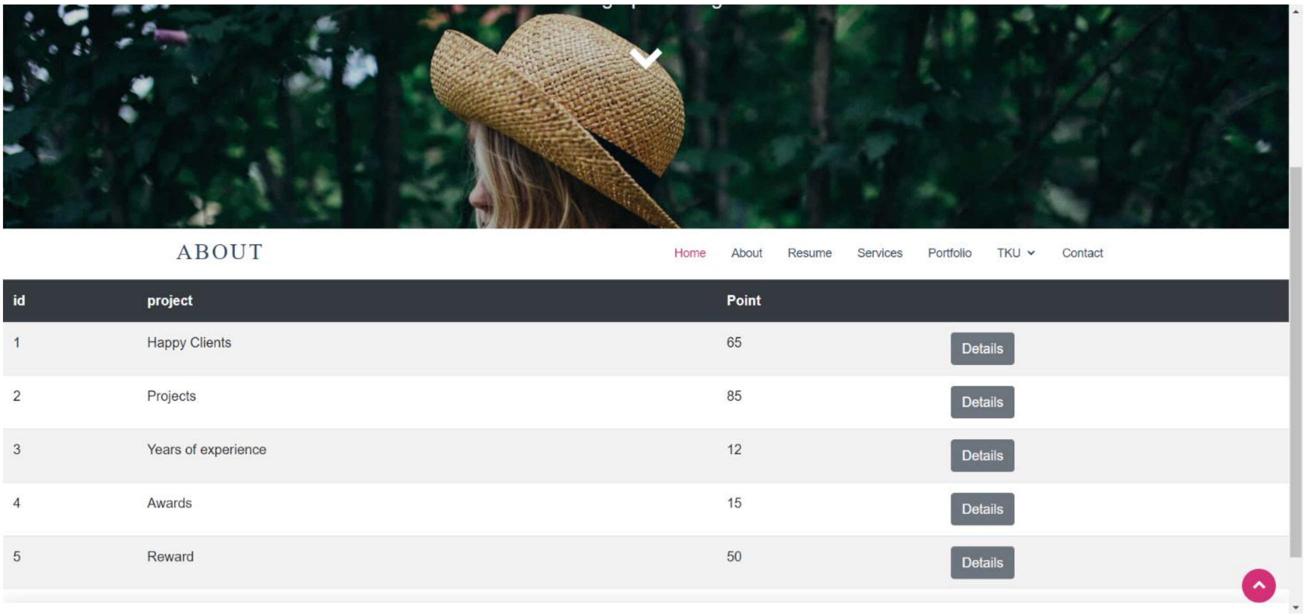
```
//routes/client/about.js

const aboutController = require("../../controllers/client/aboutController");
router.get("/", aboutController.getabout);
```

=> change the path of aboutModel.js and about.ejs



```
// controllers/client/aboutController.js

const About = require("../../models/aboutModel");

exports.getabout = async (req, res, next) => {
    let about;
    try {
      await About.fetchAll().then(([rows]) => {
        about = rows;
    });
      res.render("client/about", {
         data: about,
         title: "About" });
    } catch (err) {
        console.log(err);
    }
};
```

=> Working



id	project	Point	
1	Happy Clients	65	Details
2	Projects	85	Details
3	Years of experience	12	Details
4	Awards	15	Details
5	Reward	50	Details

Part 3 -- Use Postman to Test CRUD

=> add route /admlonely, with its handler

```
// app.js

const lonelyRouter = require('./routes/client/lonely');
const aboutRouter = require('./routes/client/about');
const admaboutRouter = require('./routes/_about');

app.use('/lonely', lonelyRouter);
app.use('/about',aboutRouter);
app.use('/admabout',admaboutRouter);
```

READ get   /admabout/_about/:id

=> copy views/client/aboout.ejs to views/_about.ejs

=> In aboutModel.js, add a function findAboutById(id)

```
// models/aboutModel.js

const db = require("../utils/database");
const about = class About {
    constructor(id, project, Point) {
        this.id = id;
        this.project = project;
        this.Point = Point;
    }

// READ
static fetchAll() {
    return db.execute("SELECT * FROM about");
}

static findById(id) {
    return db.execute('SELECT * FROM about where id = ?', [id]);
}
};
```

=> In _aboutController.js, add a function getAboutById(req,res)

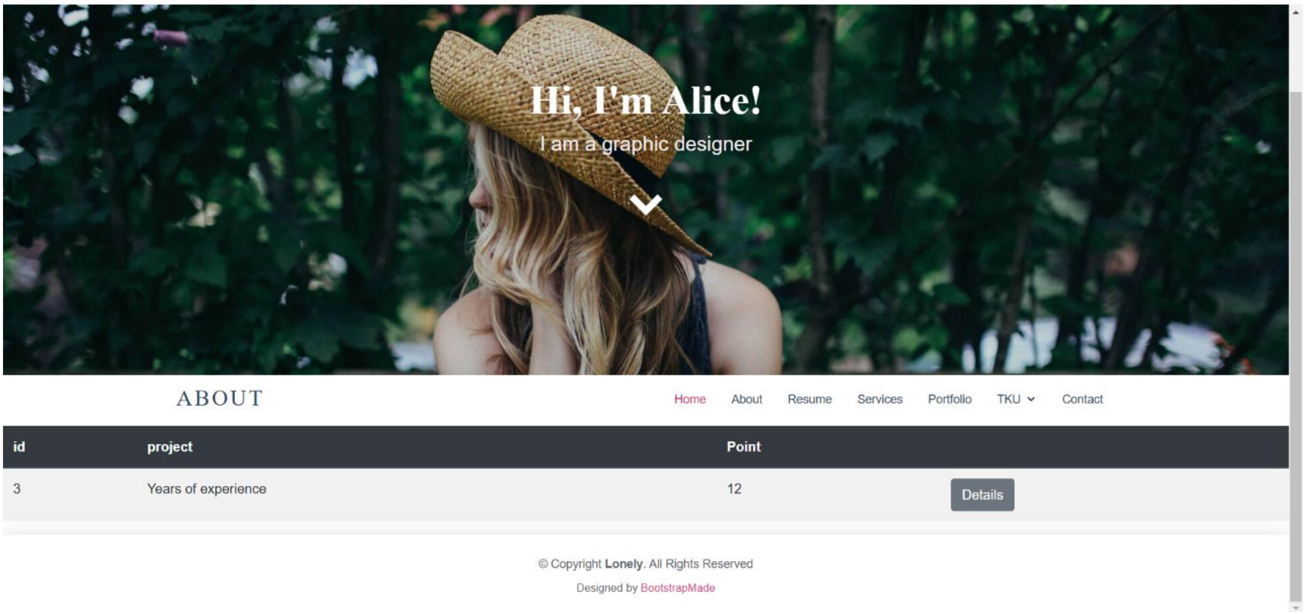
```
// controllers/_aboutController.js

const About = require("../models/aboutModel");

//READ
exports.getabout = async (req, res, next) => {
    let about;
    try {
      await About.fetchAll().then(([rows]) => {
        about = rows;
    });
      res.render("_about", {
        data: about,
        title: "About" });
    } catch (err) {
        console.log(err);
    }
};
exports.getAboutById = async (req, res) => {
    let about;
    try {
        await About.findById(req.params.id).then(([rows]) => {
            about = rows;
        });
        res.render('_about', {
          data: about,
          title: 'About',
        });
    } catch (err) {
        console.log(err);
    }
};
```

=> In routes/_about.js, add route /_about/:id, and its handler

```
const _aboutController = require('../controllers/_aboutController');

// READ
router.get('/_about', _aboutController.getabout);
router.get('/_about/:id', _aboutController.getAboutById);
```

=> Use Chrome to test

=> localhost:3000/admabout/_about/3



Hi, I'm Alice!

I am a graphic designer

ABOUT

id	project	Point	
3	Years of experience	12	Details

© Copyright **Lonely**. All Rights Reserved

Designed by BootstrapMade

UPDATE post /admabout/_about/update

=> In aboutModel.js , add a function updateById(req,res)

```
// aboutModel.js
 // Update
 static updateById(req, res) {
    const id = req.body.id;
    const project = req.body.project;
    const Point = req.body.Point;
    console.log('updateById()', id, project, Point);
    return db.execute(
        'update about set project = ?, Point = ? where id =?', [project, Point, id]

    );
 }
};
```

=> In _aboutController.js, add a function updateAboutById(req, res)

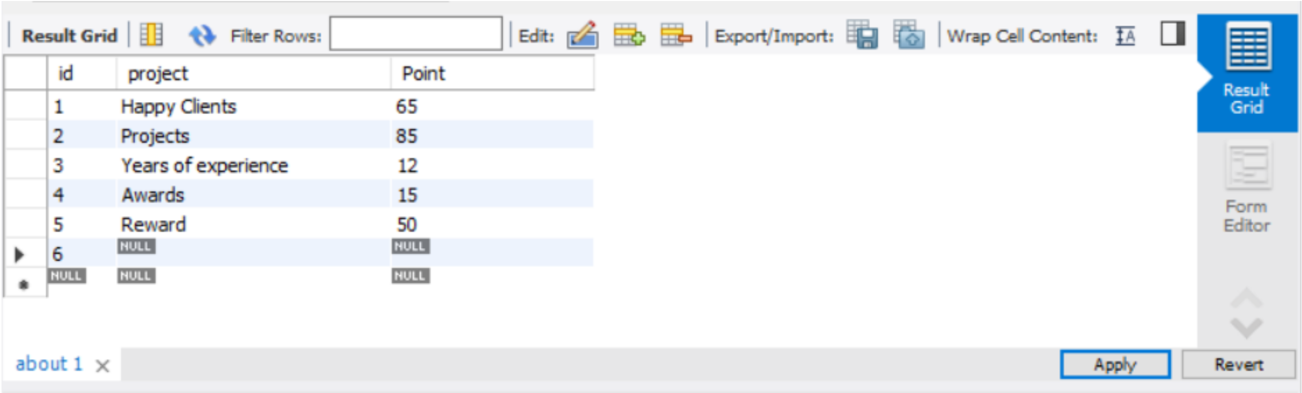
```
// controllers/_aboutController.js

// Update
exports.updateAboutById = async (req, res) => {
    try {
        await About.updateById(req, res).then(([rows]) => {
            res.redirect('/admabout/_about');
        });
    } catch (err) {
        console.log(err);
    }
};
```

=> In _about.js, add route/_about/update, and its handler

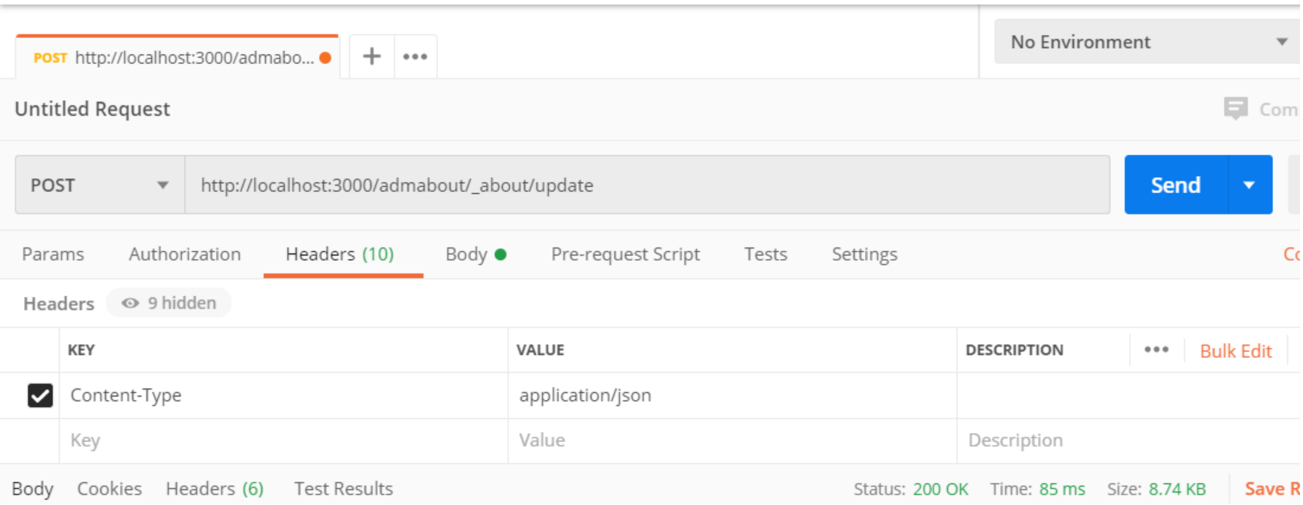
```
// routes/_about

// UPDATE
router.post('/_about/update', _aboutController.updateAboutById);
```

=> Use Postman to test

=> Use Workbench to add a record, id="6", other fields empty



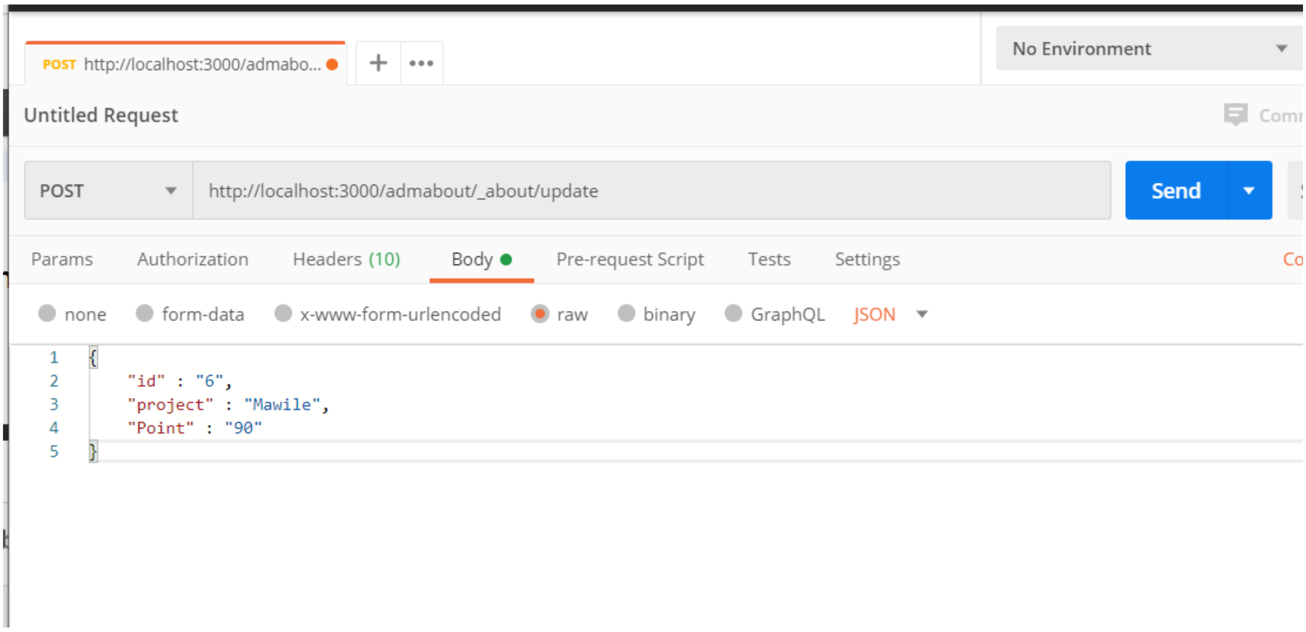
id	project	Point
1	Happy Clients	65
2	Projects	85
3	Years of experience	12
4	Awards	15
5	Reward	50
6	NULL	NULL
NULL	NULL	NULL



POST http://localhost:3000/admabout/_about/update

	KEY	VALUE	DESCRIPTION
☑	Content-Type	application/json	
	Key	Value	Description

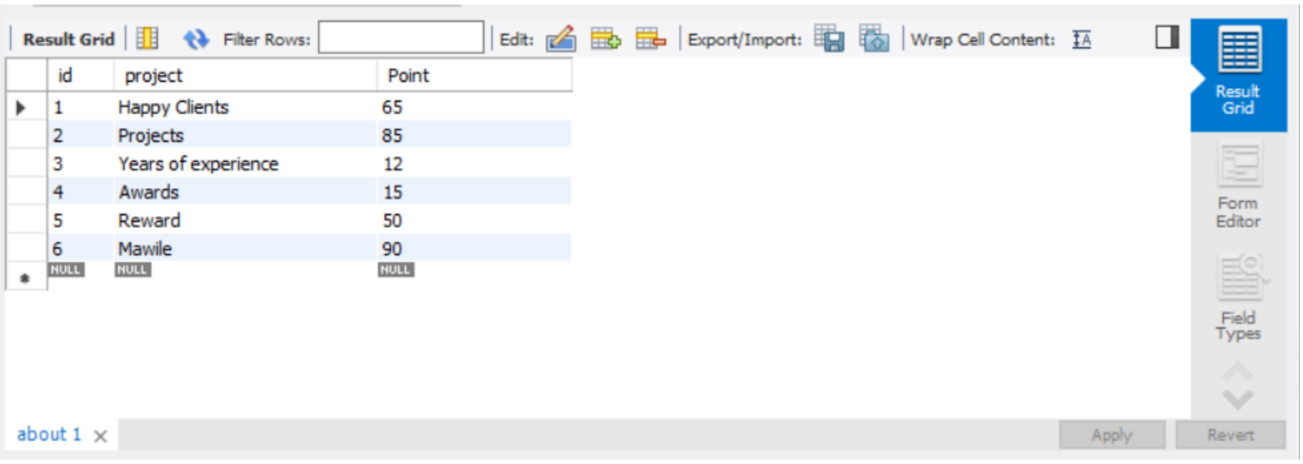
Status: 200 OK Time: 85 ms Size: 8.74 KB



POST http://localhost:3000/admabout/_about/update

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

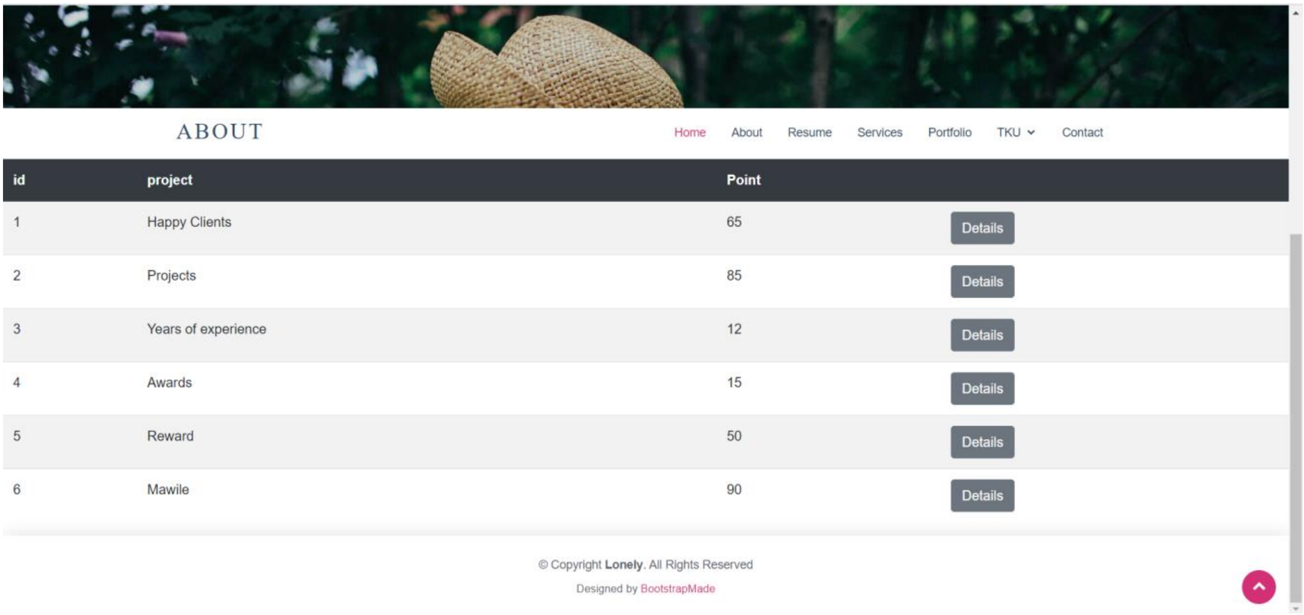
```
{
    "id" : "6",
    "project" : "Mawile",
    "Point" : "90"
}
```



id	project	Point
1	Happy Clients	65
2	Projects	85
3	Years of experience	12
4	Awards	15
5	Reward	50
6	Mawile	90
NULL	NULL	NULL

=> Check Chrome: localhost:3000/admabout/_about

=> it shows the updated "Mawile"



id	project	Point	
1	Happy Clients	65	Details
2	Projects	85	Details
3	Years of experience	12	Details
4	Awards	15	Details
5	Reward	50	Details
6	Mawile	90	Details

© Copyright **Lonely**. All Rights Reserved

Designed by BootstrapMade

CREATE post /admabout/_about/create

=> In aboutModel.js, add a function create(req,res)

```
// models/aboutModel.js

 // CREATE
 static create(req, res) {
    return db.execute(
        'INSERT INTO about (project, Point) VALUES (?, ?)',
        [req.body.project, req.body.Point]
       );
}
```

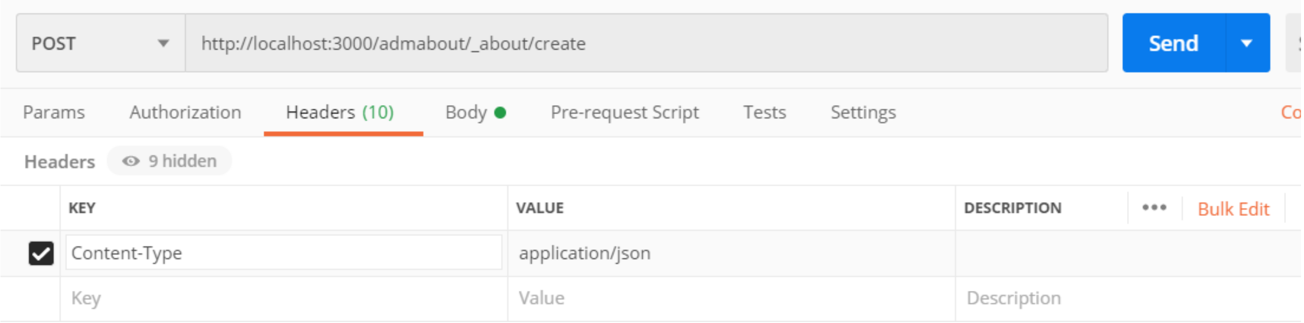
=> In _aboutController.js, add a function Create About(req,res)

```
// controllers/_aboutControllerjs
// CREATE
exports.createAbout = async (req, res) => {
    try {
        console.log('createAbout', req.body);
        await About.create(req, res).then(([rows]) => {
            res.redirect('/admabout/_about');
        });
    } catch (err) {
        console.log(err);
    }
};
```

=> In _about.js, add route/_about/create About, and its handler

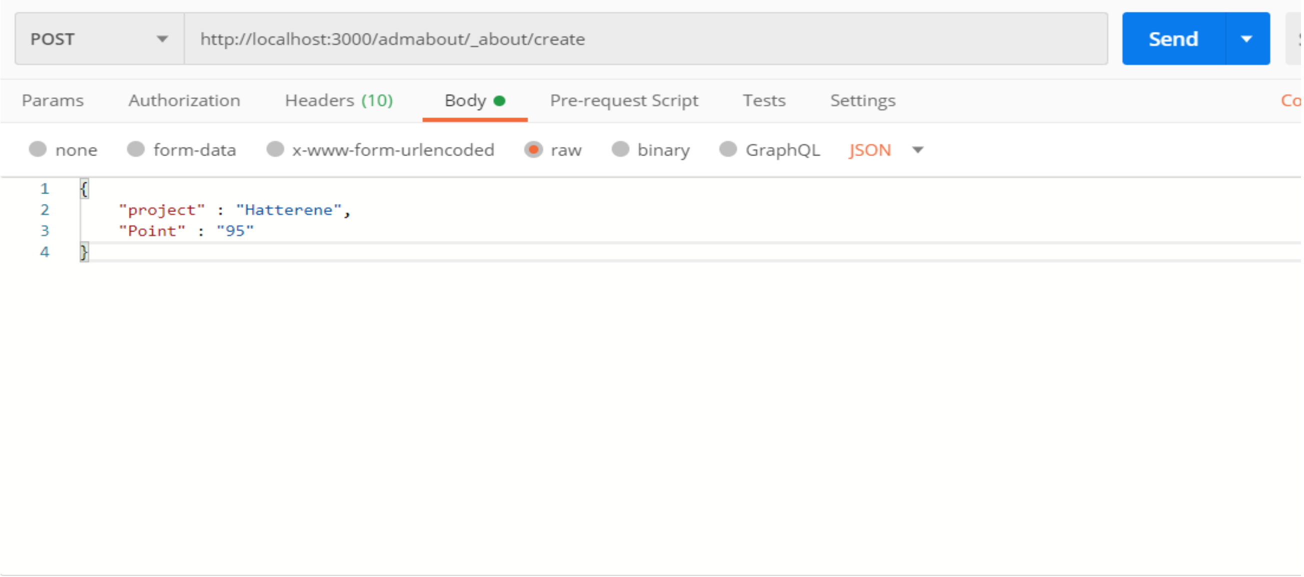
```
// routes/_aboutjs
// CREATE
router.post('/_about/create', _aboutController.createAbout);
```

=> Use Postman to test



POST http://localhost:3000/admabout/_about/create

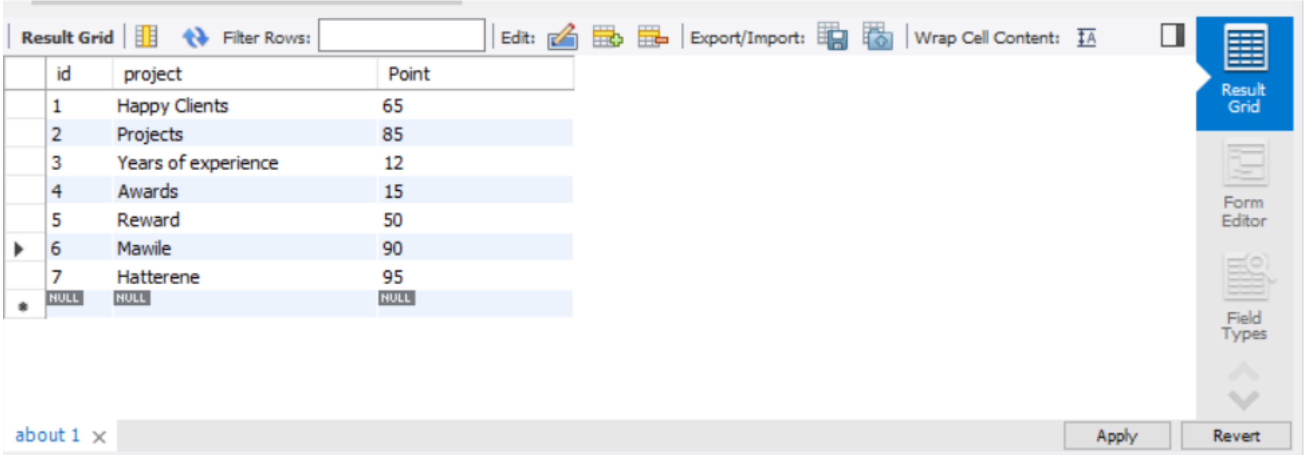
	KEY	VALUE	DESCRIPTION
☑	Content-Type	application/json	
	Key	Value	Description



POST http://localhost:3000/admabout/_about/create

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

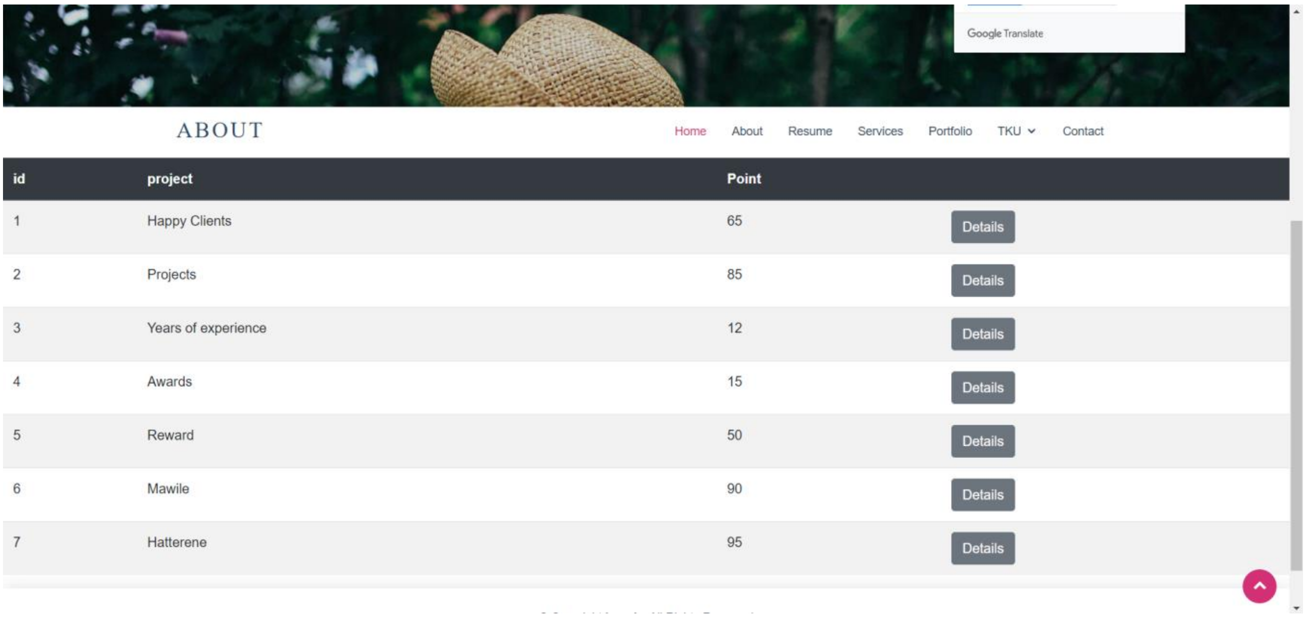
```
{
    "project" : "Hatterene",
    "Point" : "95"
}
```



id	project	Point
1	Happy Clients	65
2	Projects	85
3	Years of experience	12
4	Awards	15
5	Reward	50
6	Mawile	90
7	Hatterene	95
NULL	NULL	NULL

=> Check Chrome:localhost:3000/admabout/_about

=> it shows the newly added "Hatterene"



id	project	Point	
1	Happy Clients	65	Details
2	Projects	85	Details
3	Years of experience	12	Details
4	Awards	15	Details
5	Reward	50	Details
6	Mawile	90	Details
7	Hatterene	95	Details

DELETE get /admabout/_about/delete/:id

=> In aboutModel.js, add a function deleteById(id)

```
// models/aboutModel.js
// DELETE
 static deleteById(id) {
    return db.execute(
    'DELETE FROM about WHERE id = ?', [id]
 );
    }
};
```

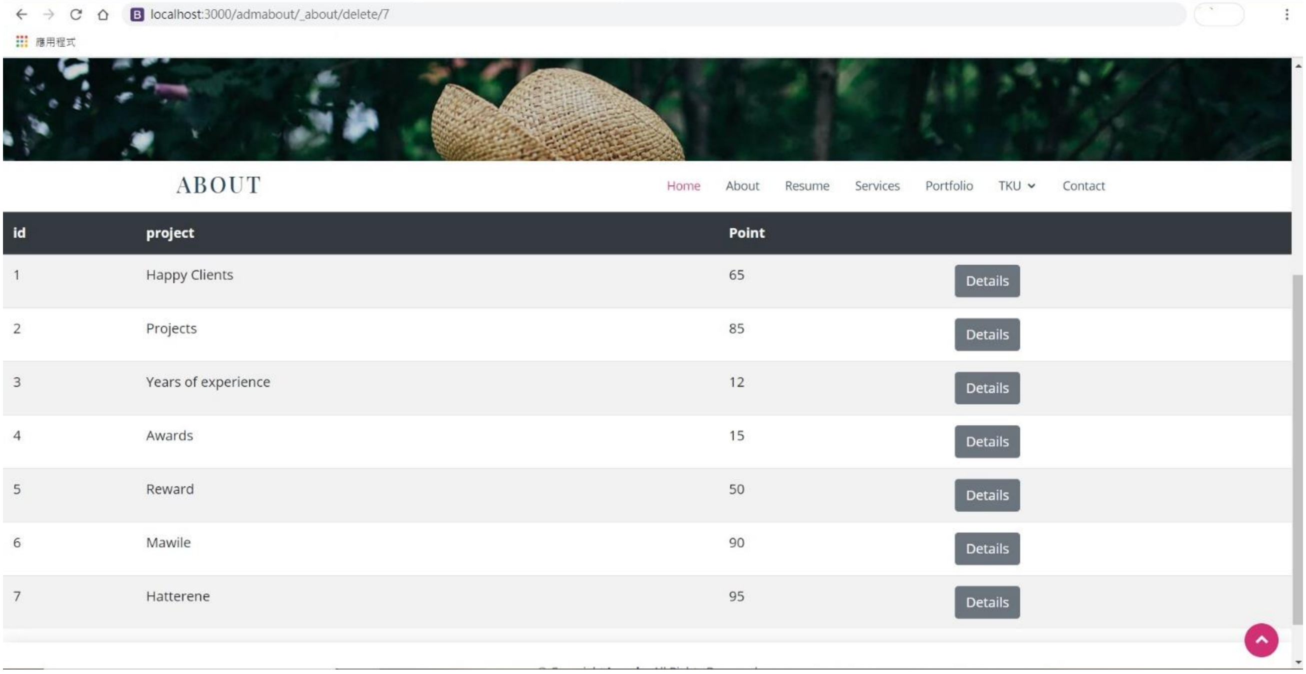
=> In _aboutController.js, add a function deleteAboutById(req, res)

```
// controllers/_aboutController.js

// DELETE
exports.deleteAboutById = async (req, res) => {
    try {
        await About.deleteById(req.params.id).then(([rows]) => {
            res.redirect('/admabout/_about');
        });
    } catch (err) {
        console.log(err);
    }
};
```

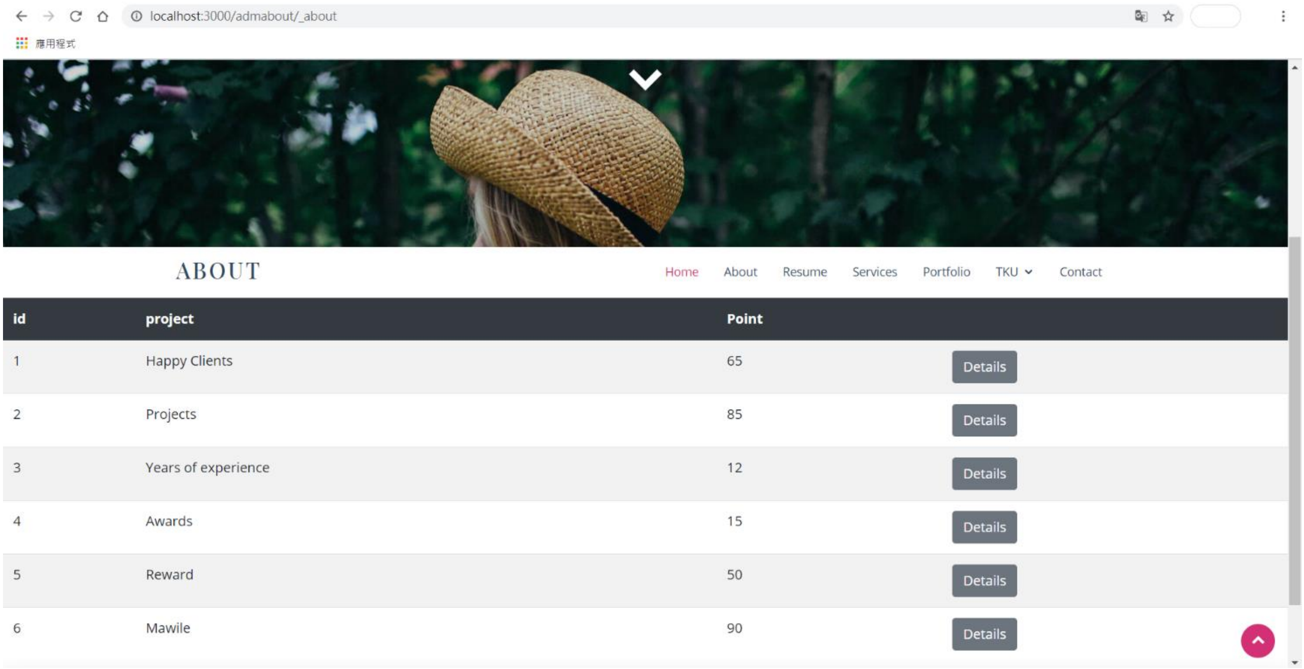
=> In _about.js, add route /_about/delete/:id, and its handler

```
// routes/_about.js

// DELETE
router.get('/_about/delete/:id', _aboutController.deleteAboutById);
```

=> Use Chrome test, delete id=7

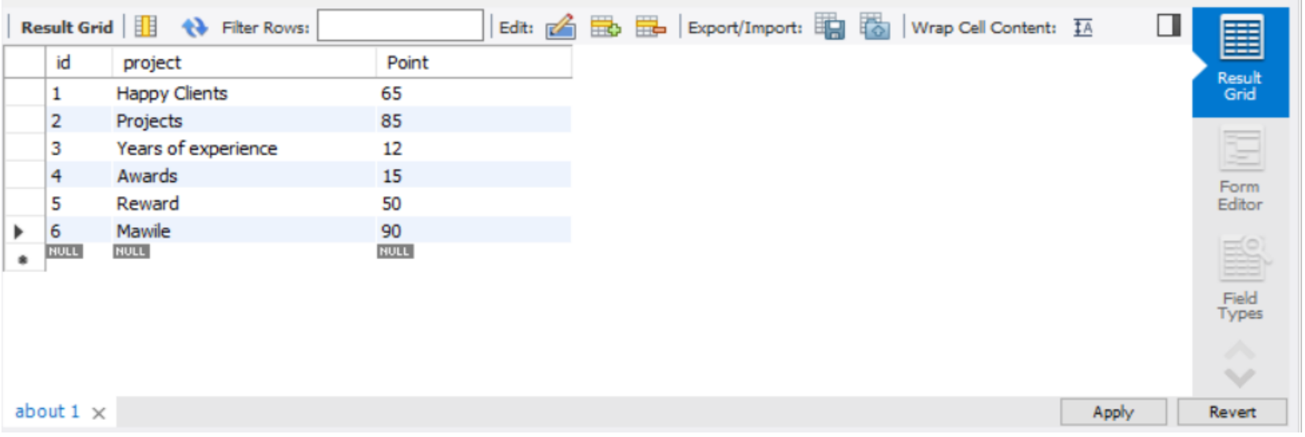


id	project	Point	
1	Happy Clients	65	Details
2	Projects	85	Details
3	Years of experience	12	Details
4	Awards	15	Details
5	Reward	50	Details
6	Mawile	90	Details
7	Hatterene	95	Details



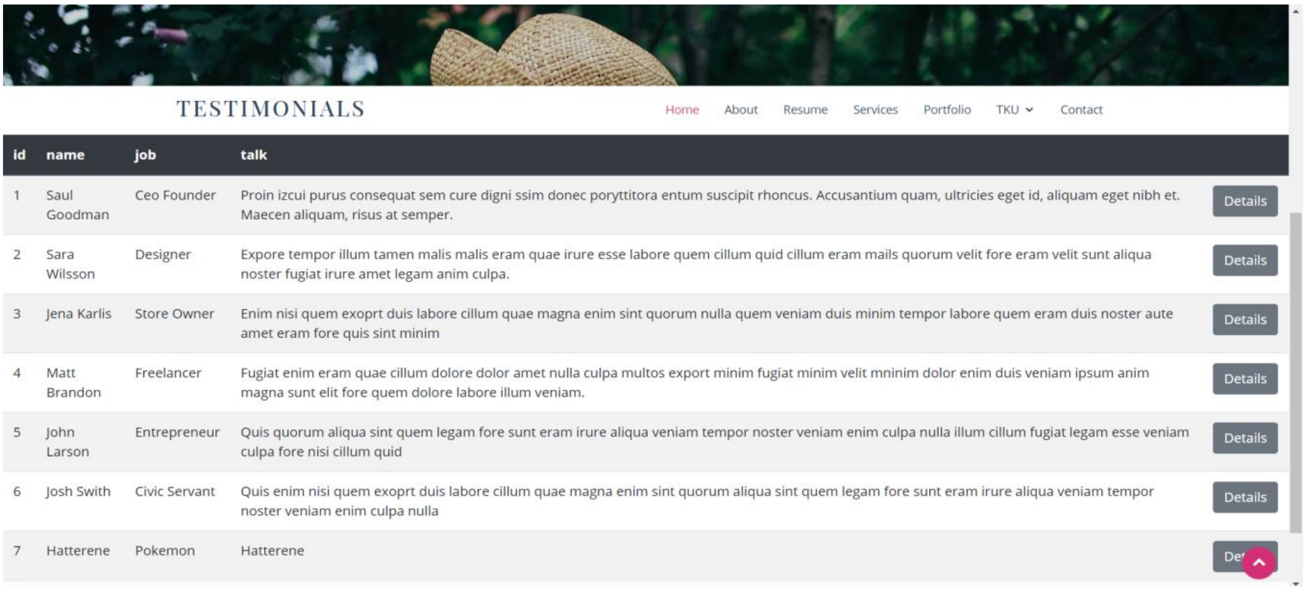
id	project	Point	
1	Happy Clients	65	Details
2	Projects	85	Details
3	Years of experience	12	Details
4	Awards	15	Details
5	Reward	50	Details
6	Mawile	90	Details

=> check with Workbench, id=7 being deleted



id	project	Point
1	Happy Clients	65
2	Projects	85
3	Years of experience	12
4	Awards	15
5	Reward	50
6	Mawile	90
NULL	NULL	NULL

Table testimonials testimonials 也以上述方法完成



id	name	job	talk	
1	Saul Goodman	Ceo Founder	Proin izcui purus consequat sem cure digni ssim donec poryttitora entum suscipit rhoncus. Accusantium quam, ultricies eget id, aliquam eget nibh et. Maecen aliquam, risus at semper.	Details
2	Sara Wilsson	Designer	Expore tempor illum tamen malis malis eram quae irure esse labore quem cillum quid cillum eram mails quorum velit fore eram velit sunt aliqua noster fugiat irure amet legam anim culpa.	Details
3	Jena Karlis	Store Owner	Enim nisi quem exoprt duis labore cillum quae magna enim sint quorum nulla quem veniam duis minim tempor labore quem eram duis noster aute amet eram fore quis sint minim	Details
4	Matt Brandon	Freelancer	Fugiat enim eram quae cillum dolore dolor amet nulla culpa multos export minim fugiat minim velit mninim dolor enim duis veniam ipsum anim magna sunt elit fore quem dolore labore illum veniam.	Details
5	John Larson	Entrepreneur	Quis quorum aliqua sint quem legam fore sunt eram irure aliqua veniam tempor noster veniam enim culpa nulla illum cillum fugiat legam esse veniam culpa fore nisi cillum quid	Details
6	Josh Swith	Civic Servant	Quis enim nisi quem exoprt duis labore cillum quae magna enim sint quorum aliqua sint quem legam fore sunt eram irure aliqua veniam tempor noster veniam enim culpa nulla	Details
7	Hatterene	Pokemon	Hatterene	Det

Lonely with foreign key

=> add pid
=> add cid
=> add two foreign keys as follows

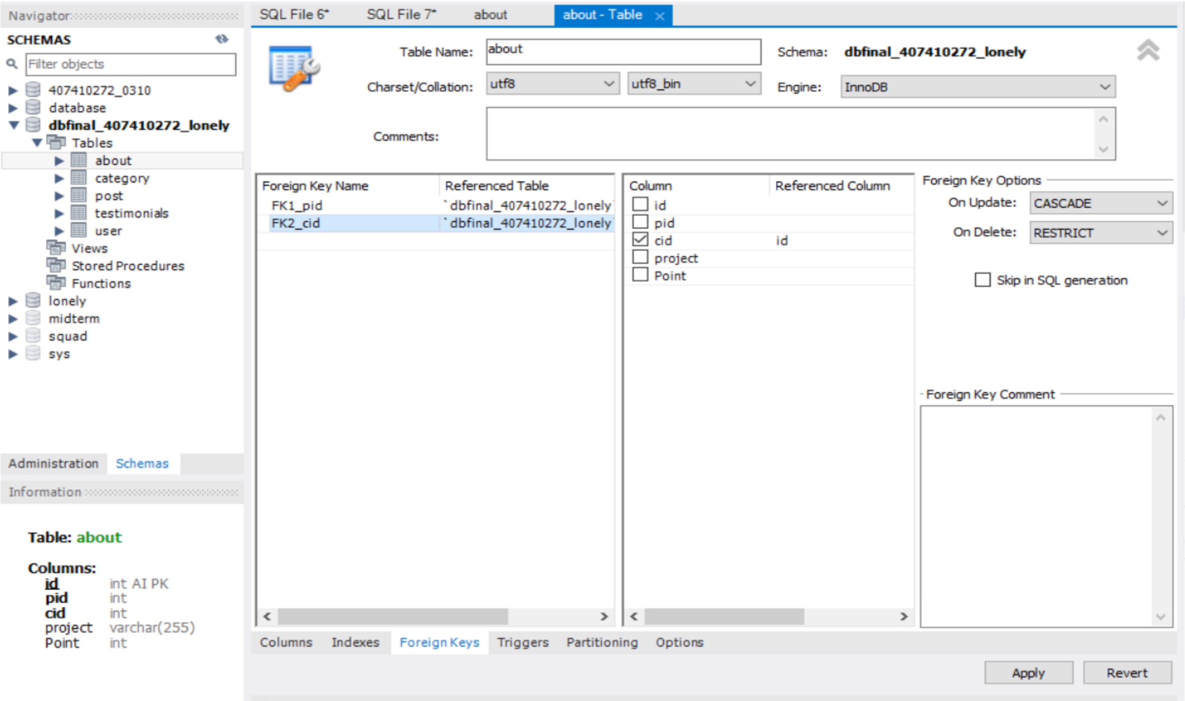


Table Name: about

Schema: dbfinal_407410272_lonely

Charset/Collation: utf8 utf8_bin

Engine: InnoDB

Comments:

Foreign Key Name	Referenced Table
FK1_pid	`dbfinal_407410272_lonely`
FK2_cid	`dbfinal_407410272_lonely`

Column	Referenced Column
□ id	
□ pid	
☑ cid	id
□ project	
□ Point	

Foreign Key Options

On Update: CASCADE

On Delete: RESTRICT

□ Skip in SQL generation

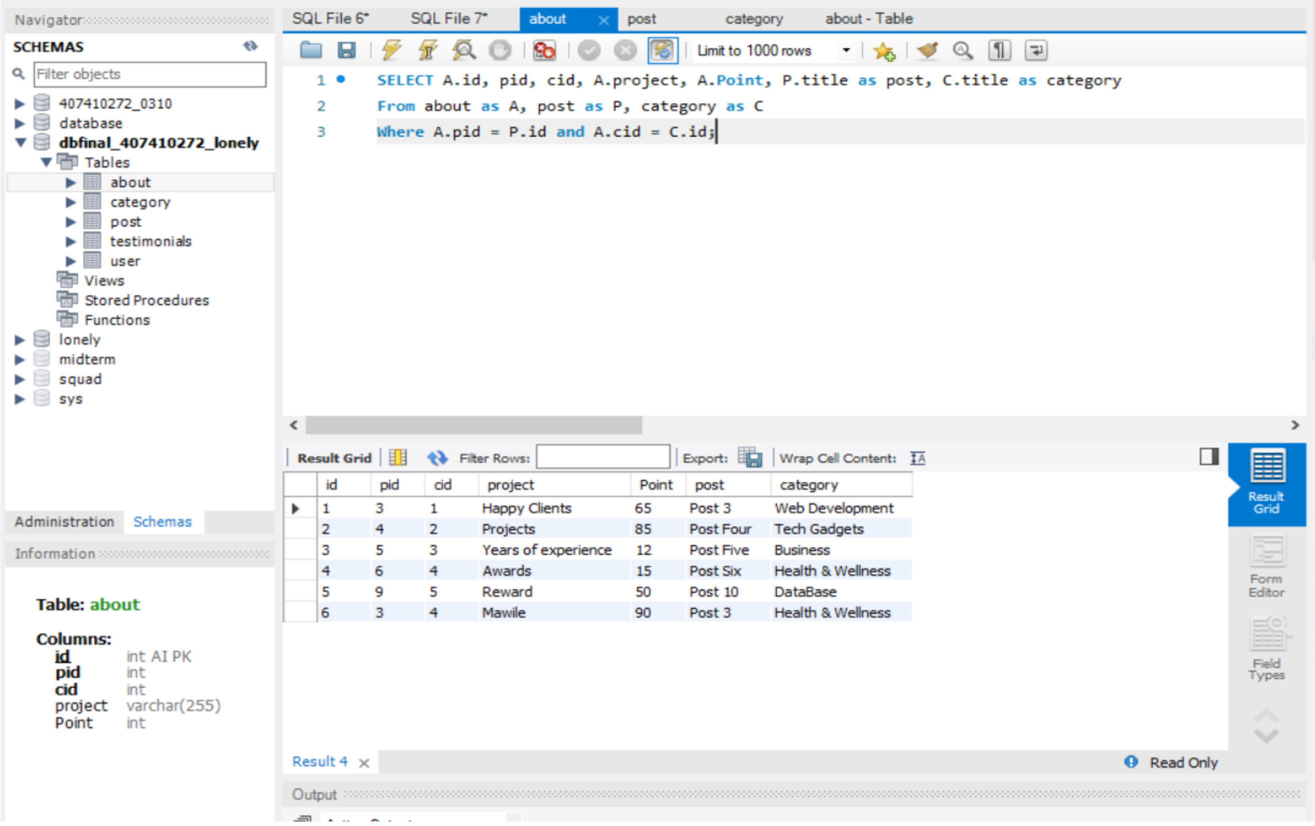
Foreign Key Comment

Table: about

Columns:

id	int AI PK
pid	int
cid	int
project	varchar(255)
Point	int

=> make a SQL to get about info with two additional fields, post and category as below



```
SELECT A.id, pid, cid, A.project, A.Point, P.title as post, C.title as category
From about as A, post as P, category as C
Where A.pid = P.id and A.cid = C.id;
```

id	pid	cid	project	Point	post	category
1	3	1	Happy Clients	65	Post 3	Web Development
2	4	2	Projects	85	Post Four	Tech Gadgets
3	5	3	Years of experience	12	Post Five	Business
4	6	4	Awards	15	Post Six	Health & Wellness
5	9	5	Reward	50	Post 10	DataBase
6	3	4	Mawile	90	Post 3	Health & Wellness

Table: about

Columns:

id	int AI PK
pid	int
cid	int
project	varchar(255)
Point	int

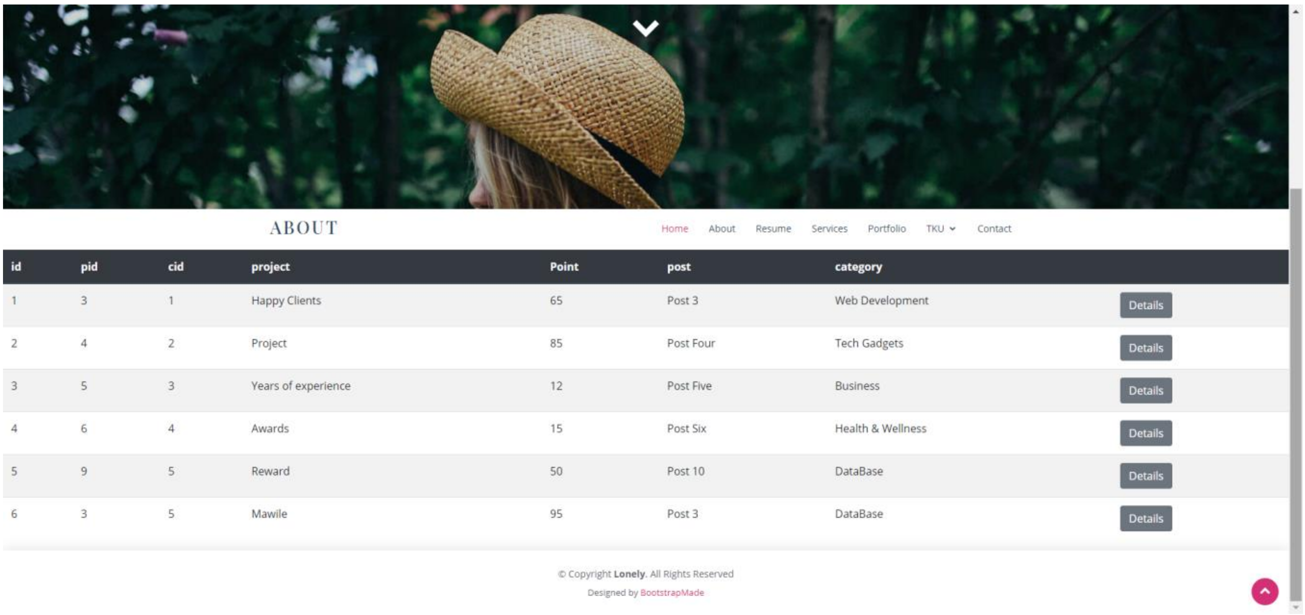
=> make changes in aboutModel.js

```
// models/aboutModel.js

const about = class About {
    constructor(id, pid, cid, project, Point) {
        this.id = id;
        this.pid = pid;
        this.cid = cid;
        this.project = project;
        this.Point = Point;
    }
 // CREATE
 static create(req, res) {
    return db.execute(
        'INSERT INTO about (pid, cid, project, Point) VALUES (?, ?, ?, ?)',
        [req.body.pid, req.body.cid, req.body.project, req.body.Point]);
}
// READ
static fetchAll() {
    return db.execute("SELECT A.id, pid, cid, A.project, A.Point, P.title as post, C.title as category\nFrom `about` as A, post as P, category as C\nWhere A.pid = P.id and A.cid = C.id\n order by A.id "
    );
}

 // Update
 static updateById(req, res) {
    const id = req.body.id;
    const pid = req.body.pid;
    const cid = req.body.cid;
    const project = req.body.project;
    const Point = req.body.Point;
    console.log('updateById()', id, pid, cid, project, Point);
    return db.execute(
        'update about set pid = ?, cid = ?, project = ?, Point = ? where id =?', [pid, cid, project, Point, id]    );
 }
```

=> change about.ejs accordingly

```
// views/ _about.ejs

<table class="table table-striped">
      <thead class="thead-dark">
        <tr>
          <th>id</th>
          <th>pid</th>
          <th>cid</th>
          <th>project</th>
          <th>Point</th>
          <th>post</th>
          <th>category</th>
          <th></th>
        </tr>
      </thead>

      <tbody>
      <% for(let i=0; i < data.length; i++){ %>
        <tr>
          <td><%= data[i].id %></td>
          <td><%= data[i].pid %></td>
          <td><%= data[i].cid %></td>
          <td><%= data[i].project %></td>
          <td><%= data[i].Point %></td>
          <td><%= data[i].post %></td>
          <td><%= data[i].category %></td>
          <td>
            <a href="/admabout/_about/details/<%= data[i].id %>"
 class="btn btn-secondary">
              <i class="fas fa-angle-double-right"></i>
              Details
            </a>
          </td>
        </tr>
        <% } %>
      </tbody>
</table>
```

=> correct from the Chrome



ABOUT

id	pid	cid	project	Point	post	category	
1	3	1	Happy Clients	65	Post 3	Web Development	Details
2	4	2	Project	85	Post Four	Tech Gadgets	Details
3	5	3	Years of experience	12	Post Five	Business	Details
4	6	4	Awards	15	Post Six	Health & Wellness	Details
5	9	5	Reward	50	Post 10	DataBase	Details
6	3	5	Mawile	95	Post 3	DataBase	Details

© Copyright **Lonely**. All Rights Reserved
Designed by BootstrapMade

Update a about

=> modify findAboutById(id)

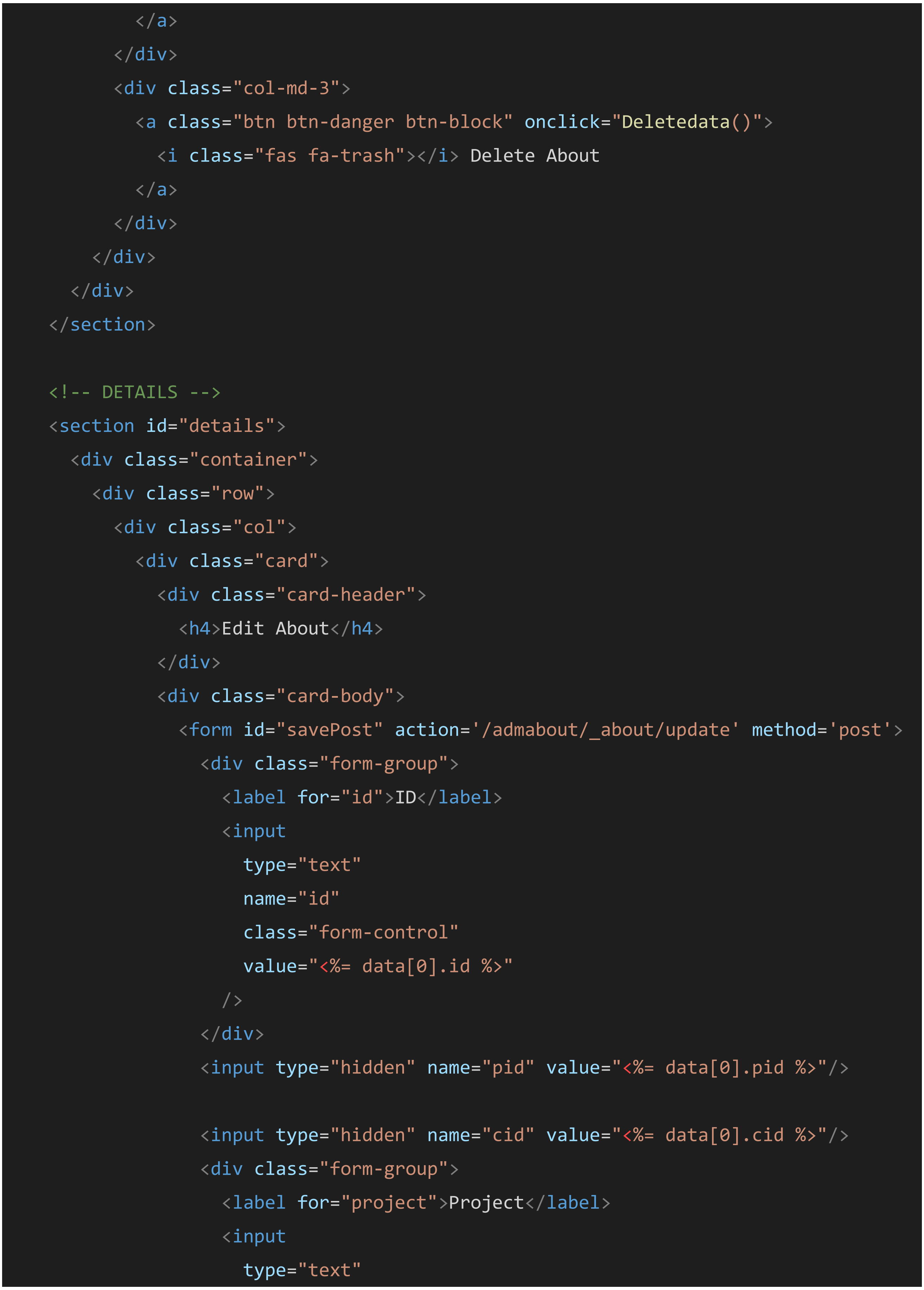
```
// models/aboutModel.js

static findById(id) {
    return db.execute("SELECT A.id, pid, cid, A.project, A.Point, P.title as post, C.title as category\nFrom `about` as A, post as P, category as C\nWhere A.pid = P.id and A.cid = C.id\n and A.id =?", [id]
);
}
```

=> modify getDetails()

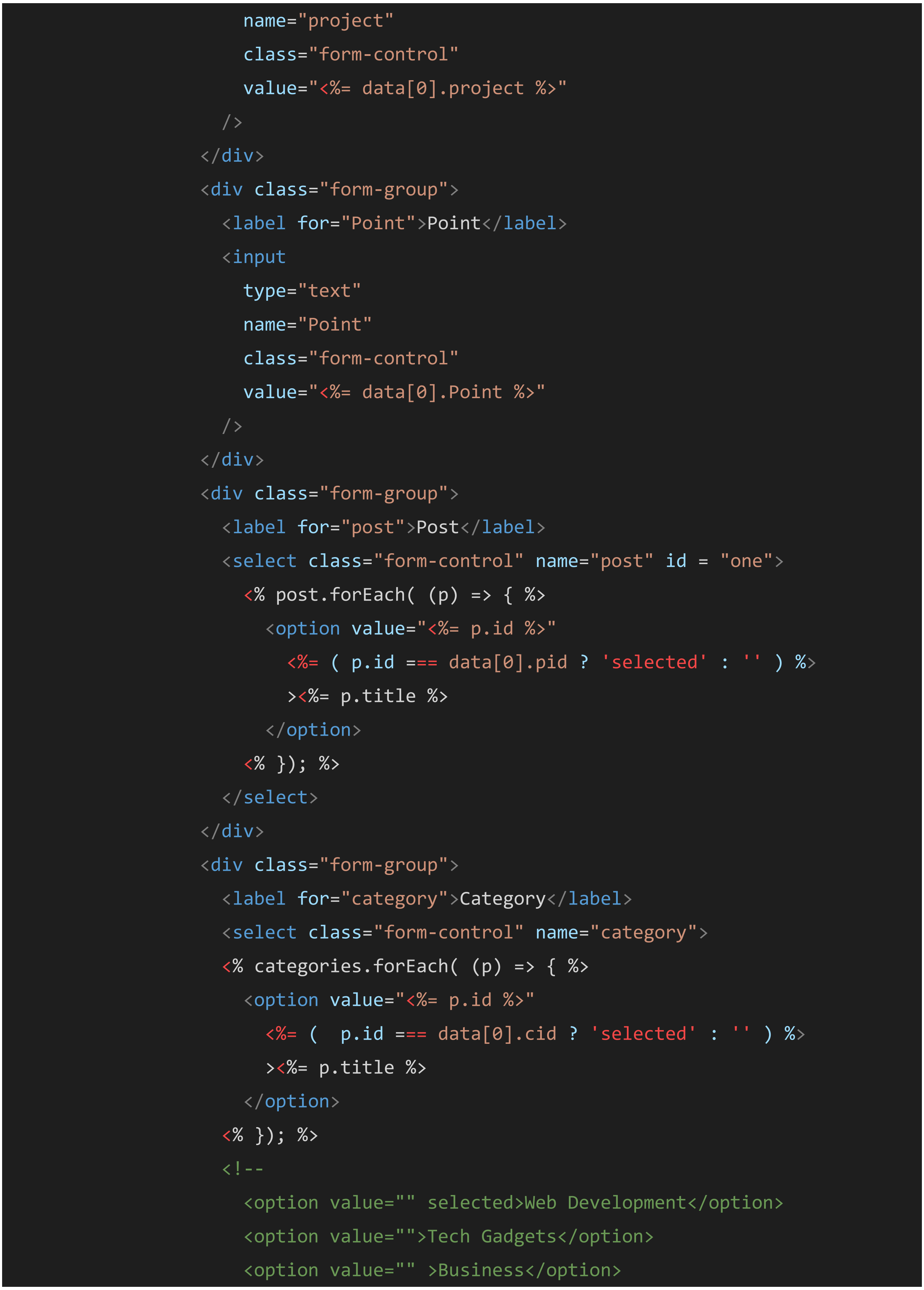
```
// controllers/_aboutcontroller.js

const About = require("../models/aboutModel");
const Post = require('../models/postModel');
const Category = require('../models/categoryModel');

exports.getDetails = async (req, res, next) => {
    let about;
    let post;
    let categories;
    try {
        console.log('id', req.params.id);
        await About.findAboutById(req.params.id).then(([rows]) => {
            about = rows;
            console.log('about', JSON.stringify(about));
        });
        await Post.fetchAll().then(([rows]) => {
            post = rows;
            // console.log('post', JSON.stringify(post));
        });
        await Category.fetchAll().then(([rows]) => {
            categories = rows;
            // console.log('category', JSON.stringify(categories));
        });
        res.render('details', {
            data: about,
            post,
            categories,
            title: 'About Details',
        });
    } catch (err) {
        console.log(err);
    }
};
```

=> modify details.ejs

```
<!-- ACTIONS -->
    <section id="actions" class="py-4 mb-4 bg-light">
      <div class="container">
        <div class="row">
          <div class="col-md-3">
            <a href="http://localhost:3000/admabout/_about" class="btn btn-light btn-block">
              <i class="fas fa-arrow-left"></i> Back To _About
            </a>
          </div>
          <div class="col-md-3">
            <a class="btn btn-success btn-block" onclick="savePost()">
              <i class="fas fa-check"></i> Save Changes
```



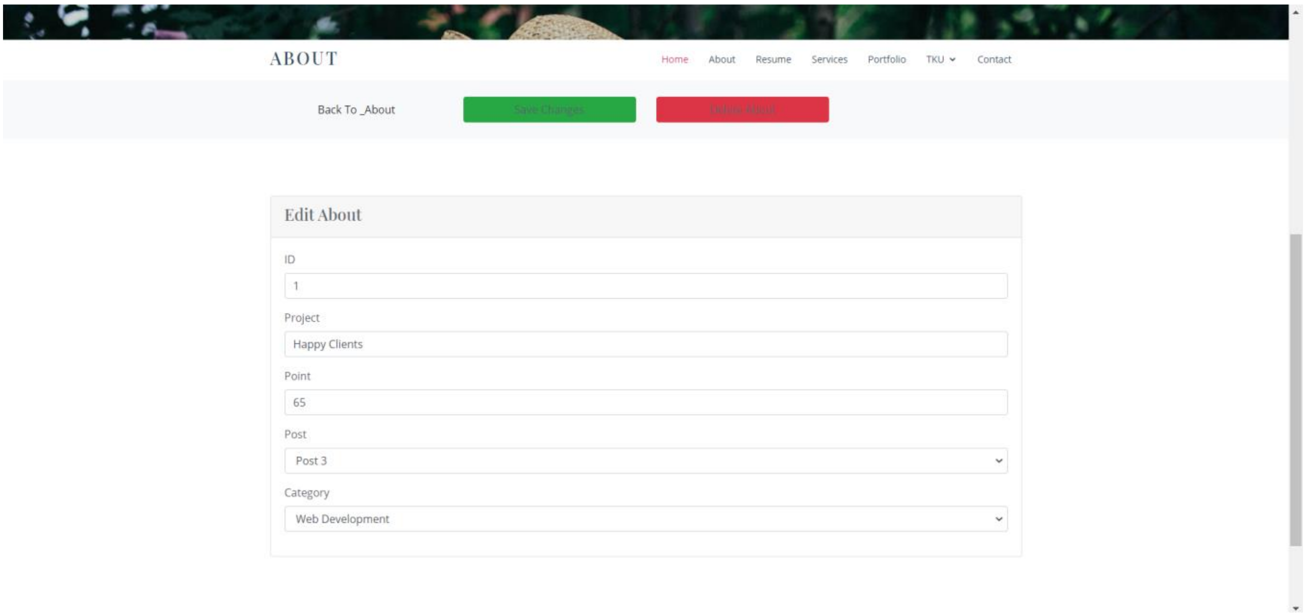
```
        </a>
      </div>
      <div class="col-md-3">
        <a class="btn btn-danger btn-block" onclick="Deletedata()">
          <i class="fas fa-trash"></i> Delete About
        </a>
      </div>
    </div>
  </div>
</section>

<!-- DETAILS -->
<section id="details">
  <div class="container">
    <div class="row">
      <div class="col">
        <div class="card">
          <div class="card-header">
            <h4>Edit About</h4>
          </div>
          <div class="card-body">
            <form id="savePost" action='/admabout/_about/update' method='post'>
              <div class="form-group">
                <label for="id">ID</label>
                <input
                  type="text"
                  name="id"
                  class="form-control"
                  value="<%= data[0].id %>"
                />
              </div>
              <input type="hidden" name="pid" value="<%= data[0].pid %>"/>

              <input type="hidden" name="cid" value="<%= data[0].cid %>"/>
              <div class="form-group">
                <label for="project">Project</label>
                <input
                  type="text"
```



```
    name="project"
    class="form-control"
    value="<%= data[0].project %>"
  />
</div>
<div class="form-group">
  <label for="Point">Point</label>
  <input
    type="text"
    name="Point"
    class="form-control"
    value="<%= data[0].Point %>"
  />
</div>
<div class="form-group">
  <label for="post">Post</label>
  <select class="form-control" name="post" id = "one">
    <% post.forEach( (p) => { %>
      <option value="<%= p.id %>"
        <%= ( p.id === data[0].pid ? 'selected' : '' ) %>
        ><%= p.title %>
      </option>
    <% }); %>
  </select>
</div>
<div class="form-group">
  <label for="category">Category</label>
  <select class="form-control" name="category">
  <% categories.forEach( (p) => { %>
    <option value="<%= p.id %>"
      <%= (  p.id === data[0].cid ? 'selected' : '' ) %>
      ><%= p.title %>
    </option>
  <% }); %>
  <!--
    <option value="" selected>Web Development</option>
    <option value="">Tech Gadgets</option>
    <option value="" >Business</option>
```

```
                      <option value="">Health & Wellness</option>
                    -->
                    </select>
                  </div>
                </form>
              </div>
            </div>
          </div>
        </div>
      </div>
    </section>
```

=> Working in Chrome



Edit About

ID

1

Project

Happy Clients

Point

65

Post

Post 3

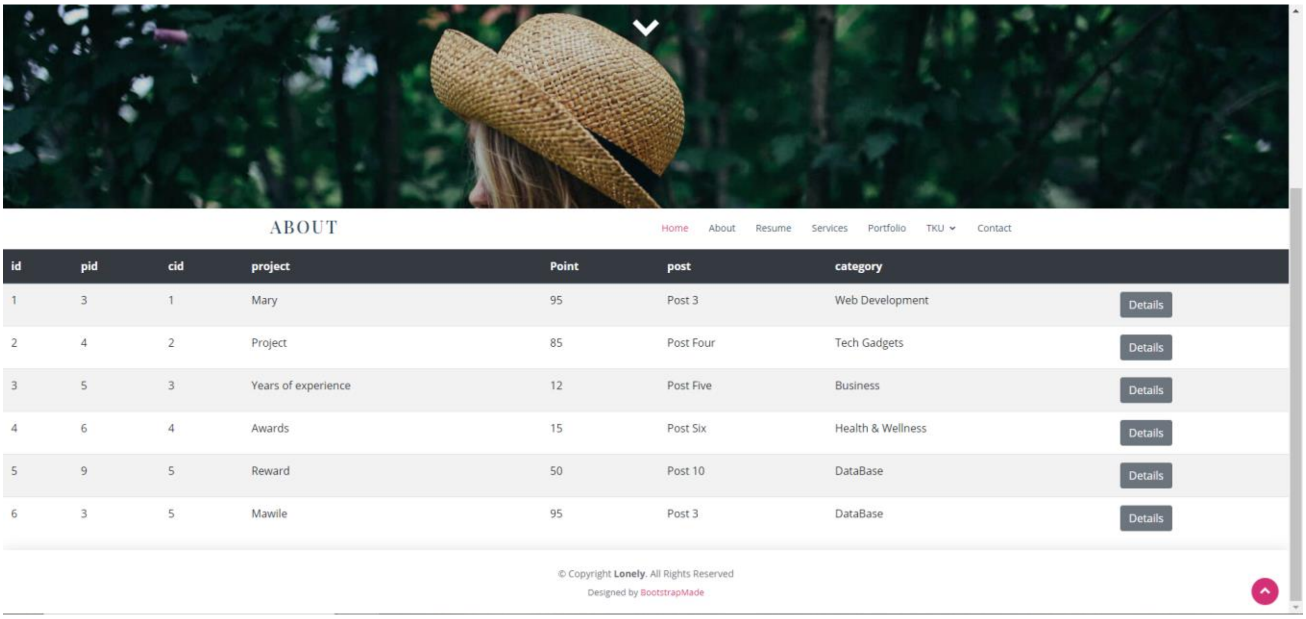
Category

Web Development

=> update some info as follows to check if is working

Project: Mary

Point: 95



id	pid	cid	project	Point	post	category	
1	3	1	Mary	95	Post 3	Web Development	Details
2	4	2	Project	85	Post Four	Tech Gadgets	Details
3	5	3	Years of experience	12	Post Five	Business	Details
4	6	4	Awards	15	Post Six	Health & Wellness	Details
5	9	5	Reward	50	Post 10	DataBase	Details
6	3	5	Mawile	95	Post 3	DataBase	Details

© Copyright **Lonely**. All Rights Reserved
Designed by BootstrapMade

遇到的問題

第一個遇到的問題: 字打錯導致程式跑錯，解決方法是花時間一行一行檢查哪裡錯字。

第二個遇到的問題: detail 更改資料時，資料無法更改，解決方法是檢查原先打好的程式，發現因為一直反覆更改，導致原本能跑得程式變成不能跑，修改後就成功了。

第三個遇到的問題: detail 更改資料時，_about 裡的 detail 的 post 跟 category 更改後值無法更改其他值更改後正常，而解決方法目前我還沒找到。