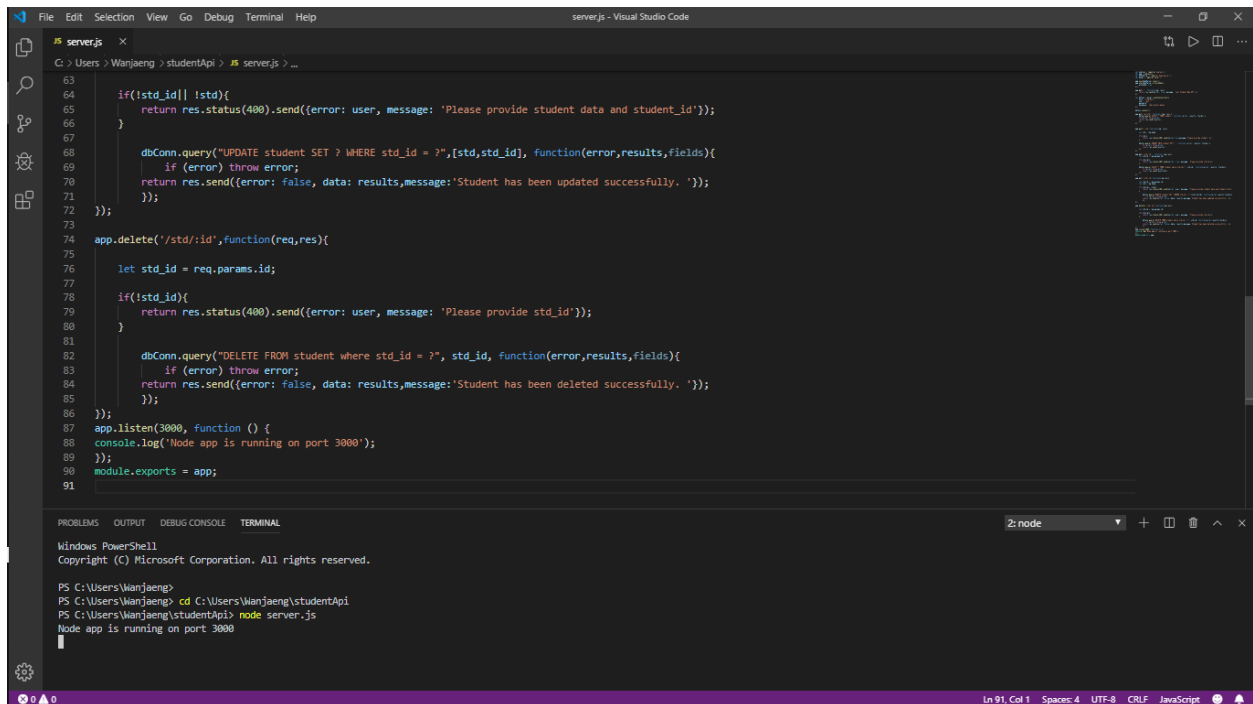


LAB 9



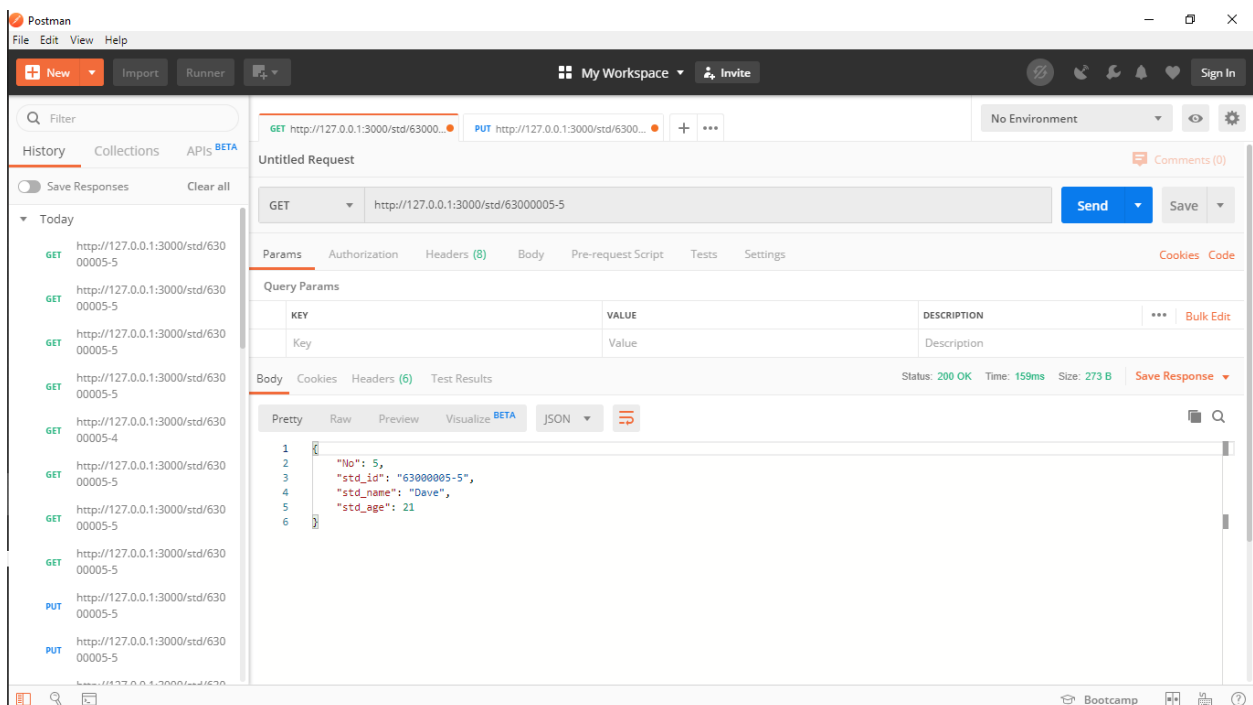
The screenshot shows the Visual Studio Code editor with a file named `server.js` open. The code defines a REST API for a student database. It includes endpoints for GET, PUT, and DELETE operations. The database is connected using `dbConn`. The terminal at the bottom shows the command `node server.js` being executed, and the output indicates that the application is running on port 3000.

```
63
64 if(!std_id || !std){
65   return res.status(400).send({error: user, message: 'Please provide student data and student_id'});
66 }
67
68 dbConn.query("UPDATE student SET ? WHERE std_id = ?",[std,std_id], function(error,results,fields){
69   if (error) throw error;
70   return res.send({error: false, data: results,message:'Student has been updated successfully. '});
71 });
72
73
74 app.delete('/std/:id',function(req,res){
75
76   let std_id = req.params.id;
77
78   if(!std_id){
79     return res.status(400).send({error: user, message: 'Please provide std_id'});
80   }
81
82   dbConn.query("DELETE FROM student where std_id = ?", std_id, function(error,results,fields){
83     if (error) throw error;
84     return res.send({error: false, data: results,message:'Student has been deleted successfully. '});
85   });
86 });
87 app.listen(3000, function () {
88   console.log('Node app is running on port 3000');
89 });
90 module.exports = app;
91
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

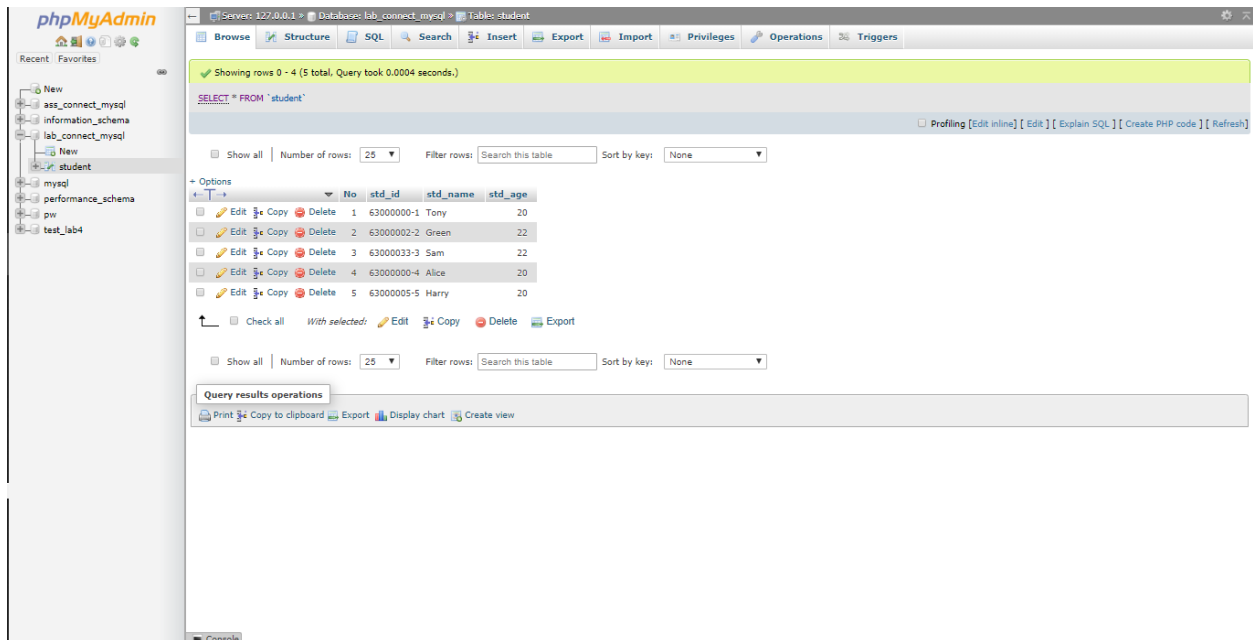
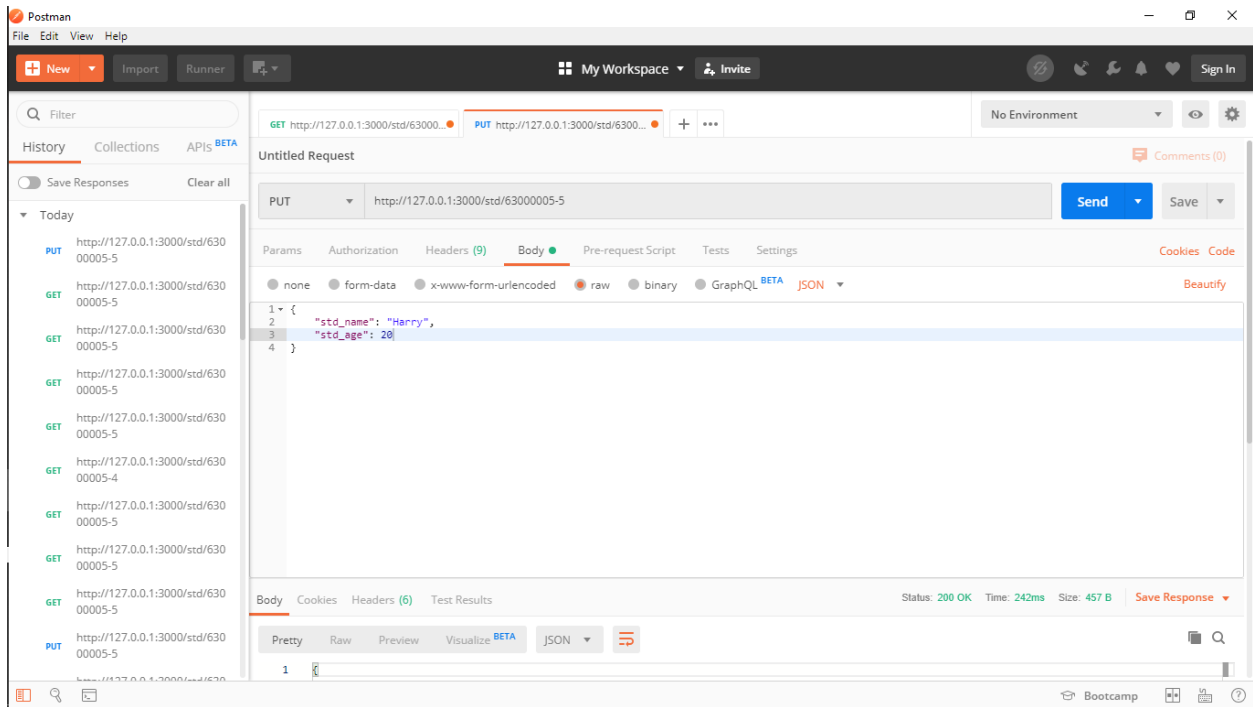
PS C:\Users\Wanjaeng>
PS C:\Users\Wanjaeng> cd C:\Users\Wanjaeng\studentApi
PS C:\Users\Wanjaeng\studentApi> node server.js
Node app is running on port 3000



The screenshot shows the Postman application with a workspace named "My Workspace". A GET request is defined with the URL `http://127.0.0.1:3000/std/63000005-5`. The request is sent, and the response is displayed in the "Body" tab. The response is a JSON object with the following structure:

```
{
  "No": 5,
  "std_id": "63000005-5",
  "std_name": "Dave",
  "std_age": 21
}
```

The response status is 200 OK, with a time of 159ms and a size of 273 B. The "Save Response" button is visible.



Postman interface showing a DELETE request to `http://127.0.0.1:3000/std/63000005-5`. The response is a JSON object indicating successful deletion.

```

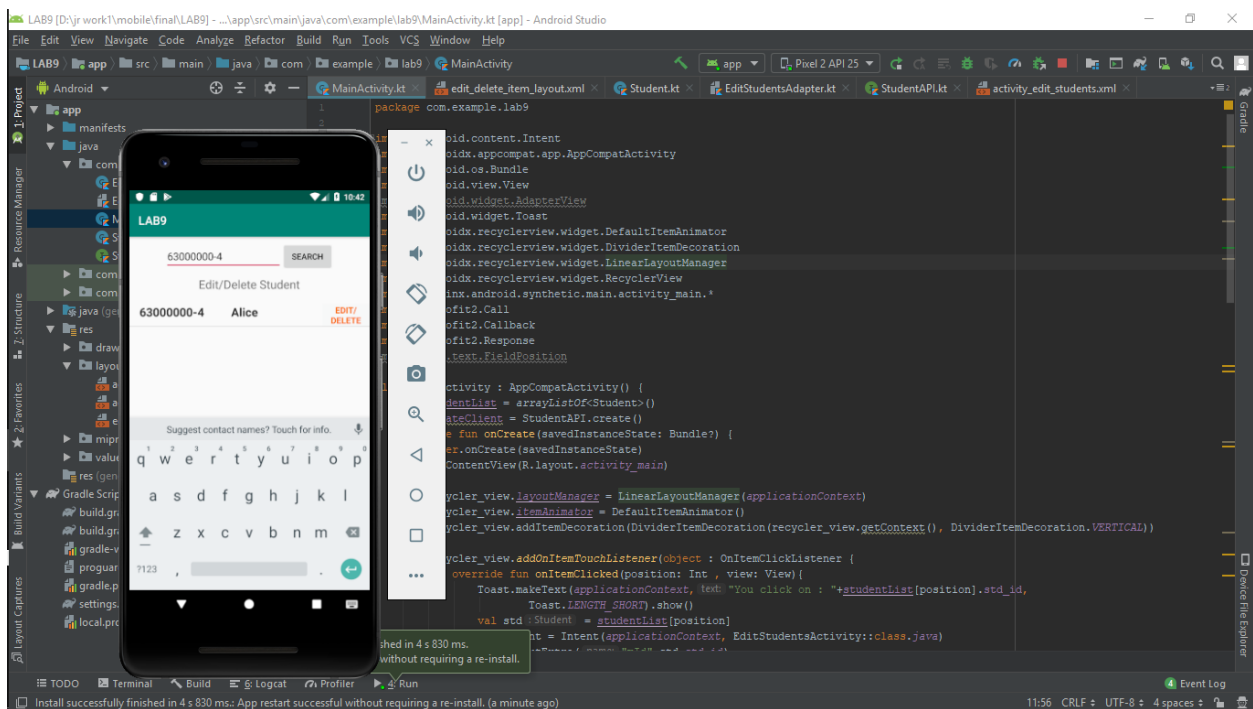
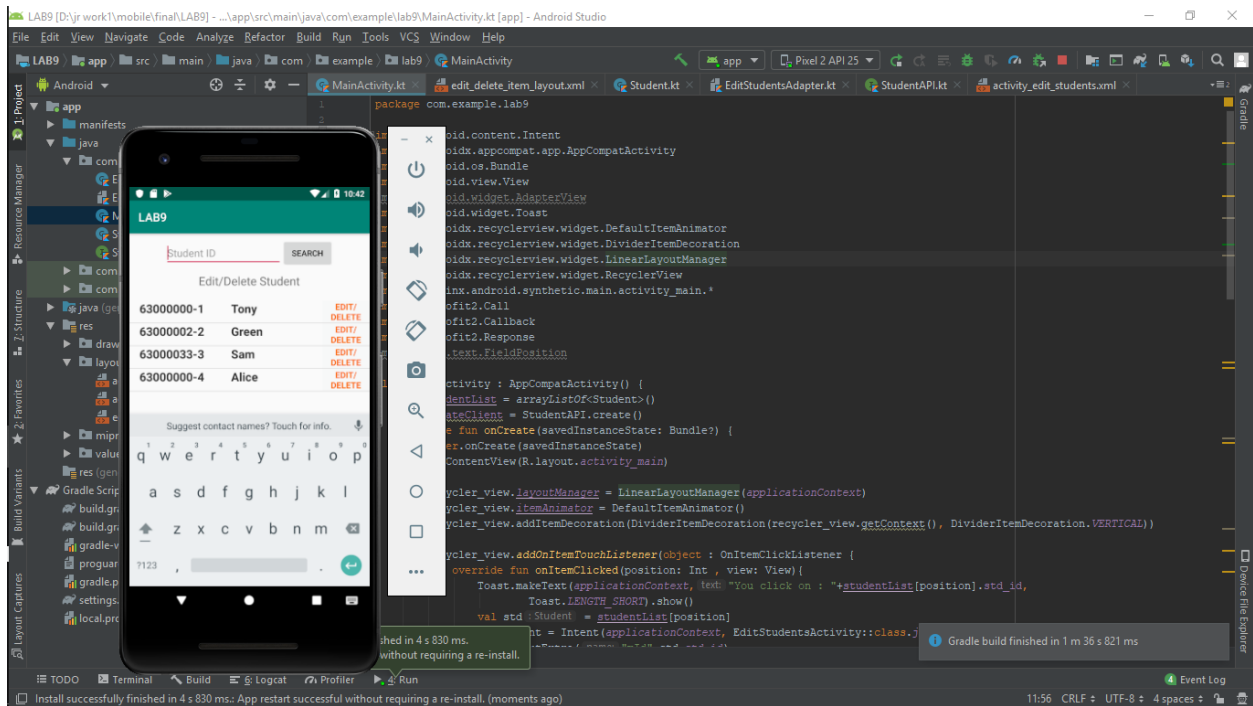
{
  "error": false,
  "data": {
    "fieldCount": 0,
    "affectedRows": 1,
    "insertId": 0,
    "serverStatus": 34,
    "warningCount": 0,
    "message": "",
    "protocol41": true,
    "changedRows": 0
  },
  "message": "Student has been deleted successfully. "
}

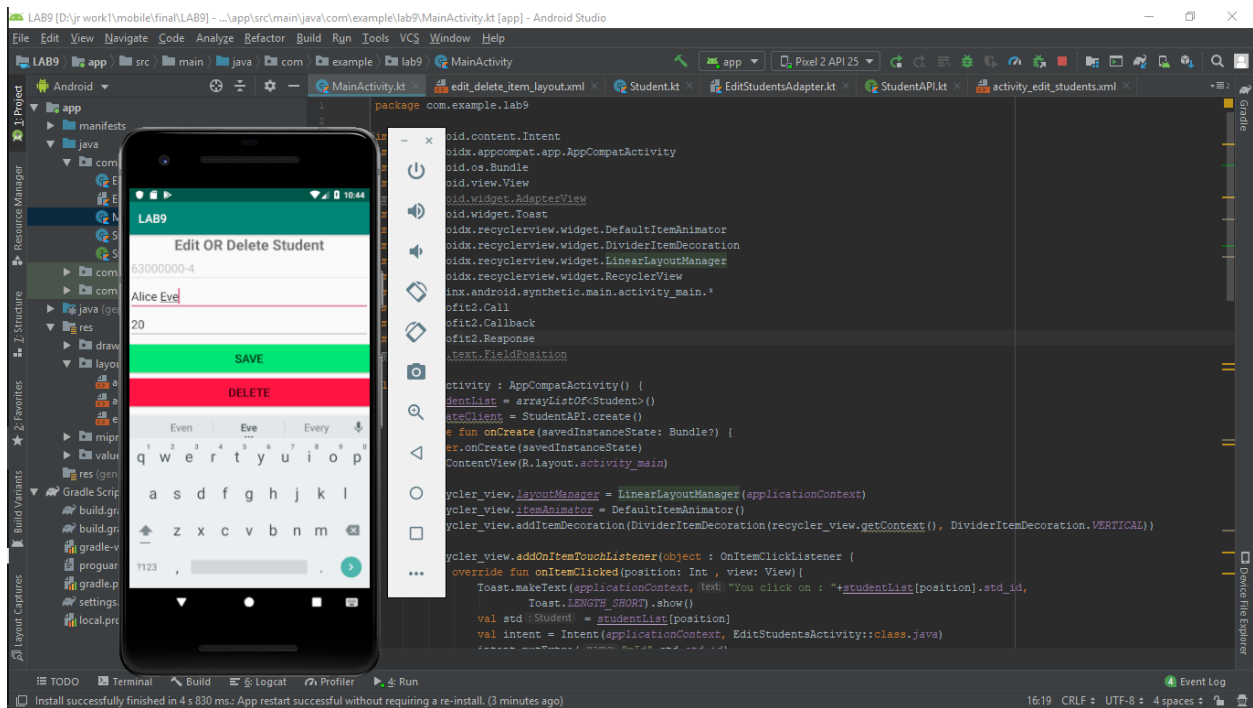
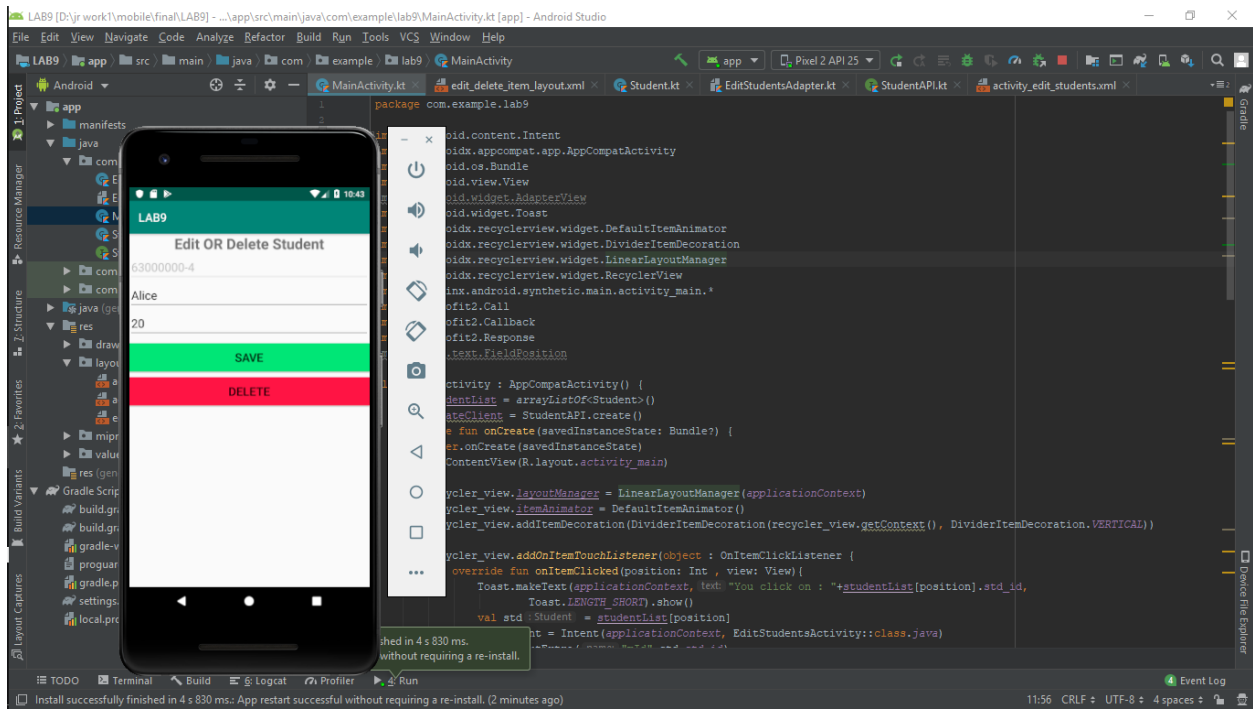
```

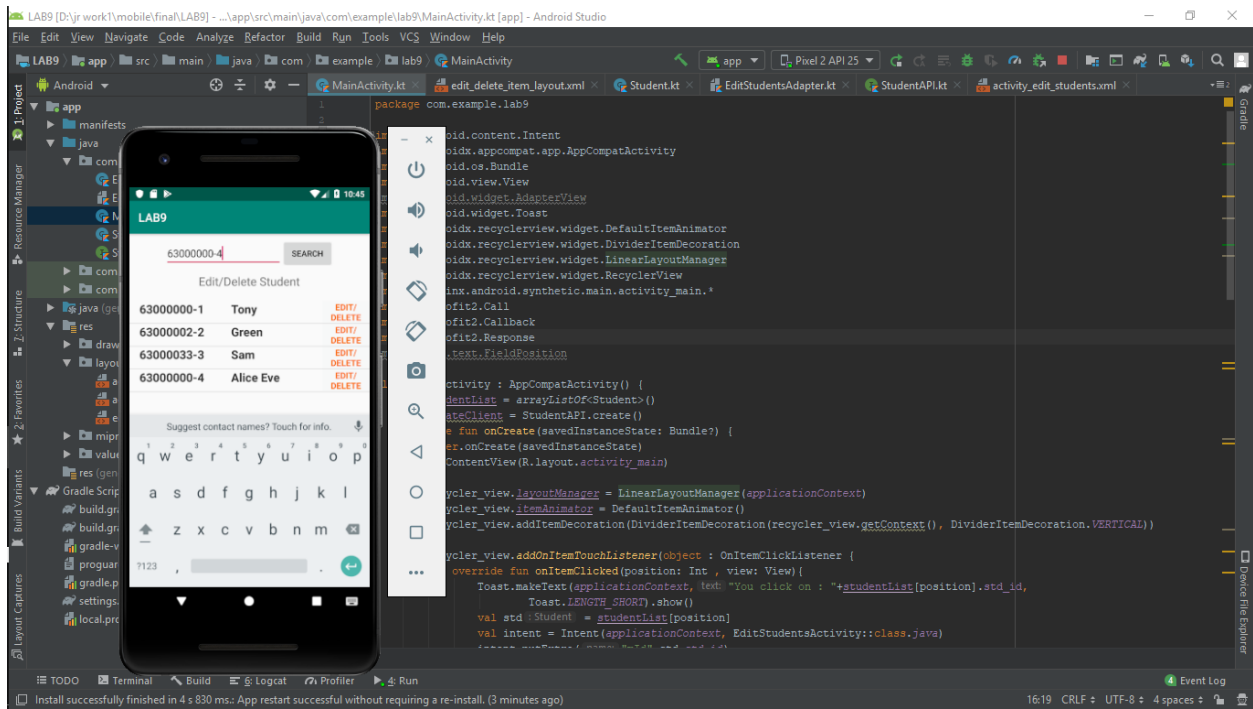
phpMyAdmin interface showing the 'student' table with 4 rows. The table columns are No, std_id, std_name, and std_age.

No	std_id	std_name	std_age
1	63000000-1	Tony	20
2	63000002-2	Green	22
3	63000033-3	Sam	22
4	63000000-4	Alice	20

ส่วน android







https://github.com/wanjaeng/LAB9_SEC1_603020802-8