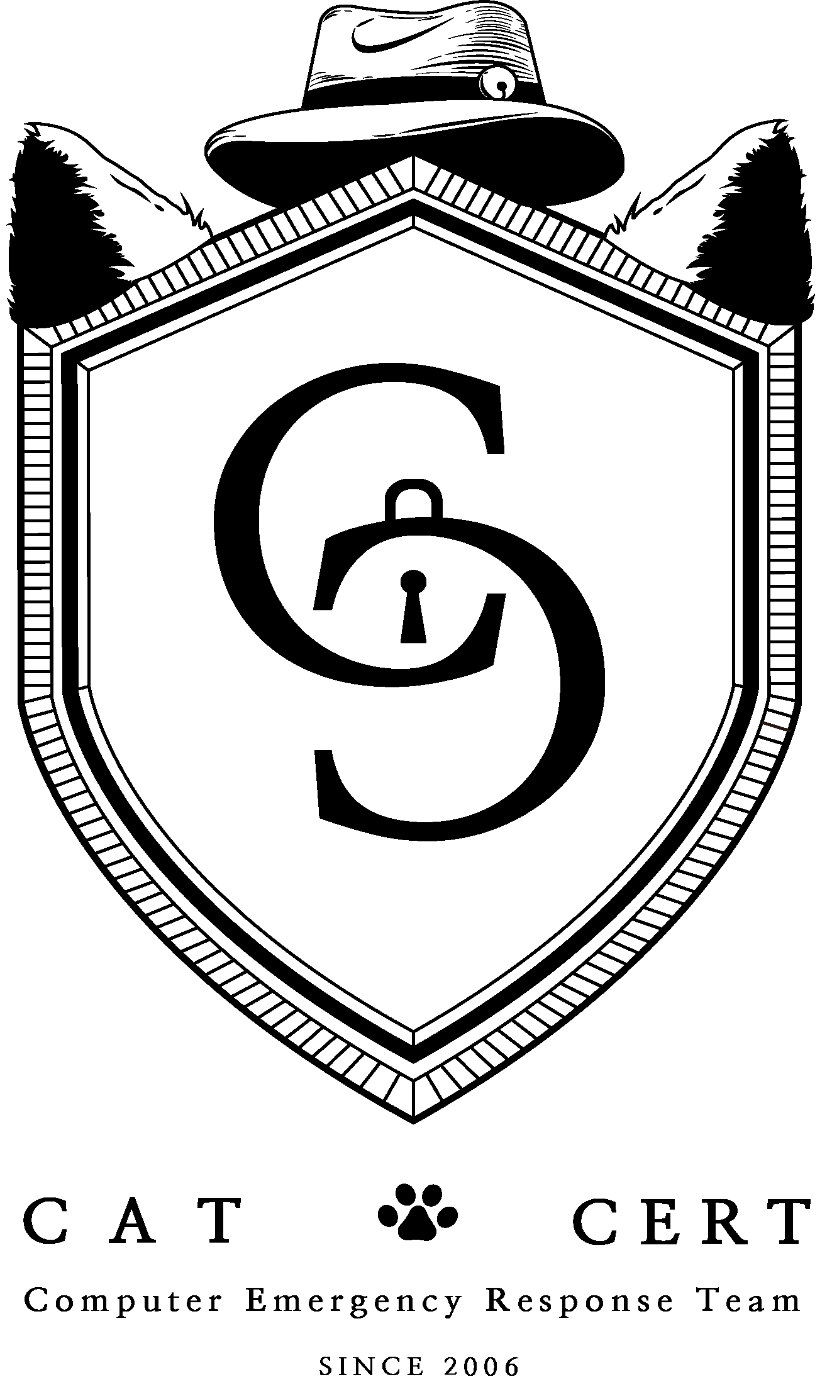
**캣서트 보고서**



202221649 인공지능학과 김민영

Web\_Study7

3.

**Node.js 방명록에 파일첨부 기능 추가**

Db.js

|  |
| --- |
| const mysql = require("mysql");  const connection = mysql.createConnection( {      host: 'localhost',      user: 'minyeong3',      password: '2022Wnsldj!',      port: 3306,      database: 'minyeong3'  });  connection.connect(function(err){      if (err) console.log(err);      else console.log('Connected!');  });  module.exports = connection; |

Index.js

|  |
| --- |
| const express = require('express');  const app = express();  const fs = require('fs');  const conn = require("./db.js");  app.set('view engine', 'ejs');  app.use('/', require('./main'));  app.listen(3000, function(){      var dir = './upload';      if (!fs.existsSync(dir)) fs.mkdirSync(dir);      console.log('listening on 3012...');  }); |

Main.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  var multer = require('multer');  const conn = require("./db.js");  var storage = multer.diskStorage({      destination(req, file, cb) {          cb(null, 'upload/');      },      filename(req, file, cb) {          cb(null, `${Date.now()}\_\_${file.originalname}`);      },  });  var upload = multer({ dest: 'upload/'});  var uploadWithOriginalFilename = multer({ storage: storage});  router.get("/", function (req, res) {      var sql = "SELECT \* FROM memo order by seq desc";      conn.query(sql, function(err, rows) {          if (err) console.log("query is not excuted. select fail!\n" + err);          else res.render(\_\_dirname +"/views/memo.ejs", {list: rows });      });  });    router.post('/uploadFileWithOriginalFileName', uploadWithOriginalFilename.single('attachment'), function(req,res){      console.log( req.body );        var ti = req.body.title;      var con = req.body.content;      var na = req.body.name;     // var fil=req.file.filename;      var sql = "INSERT INTO memo(title, content, name) VALUES (?,?,?)";      var params =[ti, con, na];        conn.query(sql, params, function(err, rows){          if (err) console.log("query is not excuted. modify fail!\n" + err);          else res.render(\_\_dirname + "/views/upload.ejs", {file:req.file, files:null});      })    });  router.get("/write",  function(req, res){      res.render(\_\_dirname +"/views/upload.ejs");  });  router.post( "/write", function ( req, res ) {      console.log( req.body );        var ti = req.body.title;      var con = req.body.content;      var na = req.body.name;      //var fil=req.body.attachment;      //console.log(req.file);      var sql = "INSERT INTO memo(title, content, name) VALUES (?,?,?)";      var params =[ti, con, na, fil];        conn.query(sql, params, function(err, rows){          if (err) console.log("query is not excuted. modify fail!\n" + err);          else res.redirect("/");      })  });  router.get("/modify/:seq", function(req, res){      var num = req.params.seq;      console.log(num);      res.render(\_\_dirname +"/views/upload.ejs");  });  router.post( "/modify/:seq", function ( req, res ) {      console.log( req.body );        var num = req.params.seq;      console.log(num);      const ti = req.body.title;      console.log(ti);      const con = req.body.content;      console.log(con);      const na = req.body.name;      console.log(na);        var sql= "update memo set title=?, content=?, name=? where seq=?";      var params =[ti, con, na, num];        conn.query(sql, params, function(err, rows){          if (err) console.log("query is not excuted. modify fail!\n" + err);          else res.redirect("/");      })  });  router.get("/delete/:seq", function(req, res){      var num = req.params.seq;      console.log(num);      var sql = "delete from memo where seq=?";      var params=[num];      conn.query(sql, params, function(err, rows){          if (err) console.log("query is not excuted. modify fail!\n" + err);          else res.redirect("/");      })  });  router.get("/download/:file", function(req,res){        var noo = req.params.file;      var file = './upload/'+noo;      console.log(file);      res.download(file);  });  module.exports = router; |

memo.ejs

|  |
| --- |
| <html>  <head>  <title> 방명록 </title>  <style>  table {     width: 100%;     text-align: center;     border-collapse: collapse;     border-top: 2px solid #ed0606;  }  th,  td {     border-bottom: 1px dashed #001763;  }  </style>  </head>  <body>  <h2> 방명록 </h2>  <div>  <table>     <thead>     <th> 제목 </th>      <th> 내용 </th>      <th> 글쓴이 </th>     </thead>  <tbody> <% for(var i of list) { %> <tr>     <td> <%= i.title %></td>     <td> <%= i.content %></td>     <td> <%= i.name %></td>     <td><a href="/download/<%=i.file %>"><button>File Download</button> </a></td>     <td><a href="/modify/<%=i.seq %>"><button>[수정]</a></button>        <a href="/delete/<%=i.seq %>"><button>[삭제]</a></button></td>     </tr> <% } %> </tbody>     <a href="/write"><button>글쓰기</button></a>  </table>  </div>  </body>  </html> |

Upload.ejs

|  |
| --- |
| <!doctype html>  <head>  <meta charset="UTF-8">  <style>          table.table2 {              border-collapse: separate;              border-spacing: 1px;              text-align: left;              line-height: 1.5;              border-top: 1px solid #ccc;              margin: 20px 10px;          }          table.table2 tr {              width: 50px;              padding: 10px;              font-weight: bold;              vertical-align: top;              border-bottom: 1px solid #ccc;          }          table.table2 td {              width: 100px;              padding: 10px;              vertical-align: top;              border-bottom: 1px solid #ccc;          }      </style>  <title>방명록</title>  </head>  <body>          <h1><a href="/">방명록</a></h1>    <form action="/uploadFileWithOriginalFilename" enctype="multipart/form-data" method="post">      <table style="padding-top:50px" align=center width=auto border=0 cellpadding=2>          <tr>              <td style="height:40; float:center; background-color:#3C3C3C">                  <p style="font-size:15px; text-align:center; color:white; margin-top:15px; margin-bottom:15px"><b>글 작성하기</b></p>              </td>          </tr>          <tr>              <td bgcolor=white>                  <table class="table2">                     <tr>                      <td> 제목 <input type="text" name="title"/><br></td>                      </tr>                       <tr>                      <td> <textarea rows="10" cols="50" name="content" placeholder='1000자 이상 입력하세요.'></textarea>                      </tr>                      <tr>                          <td> 이름 <input type="text" name="name"/><br></td>                      </tr>              </table>              <center>        <input type="file" name="attachment">      <button type="submit" class="btn btn-primary"> Upload</button>  </form>  </center>  </td>  </tr>  </form>  </table>  </body>  </html> |

Download.js

|  |
| --- |
| var express = require('express');  var router = express.Router();  const path = require('path');  var appRoot = process.env.PWD;  var uploadPath = path.join(appRoot, "upload");  router.get('/', function(req, res){      res.render("index");  });  router.get('/download', function(req,res){      const file = './upload';      console.log(file);      res.download(file);  });  module.exports = router; |

텍스트이(가) 표시된 사진

자동 생성된 설명



게시판 프로젝트

*-회원가입*

*-로그인 & 로그아웃*

*-회원정보수정*

*-탈퇴*

*-비밀번호찾기*

*-글쓰기*

*-글수정*

*-글삭제*

*-글목록(로그인 시에만 보이게)*

*-글검색*

*-글보기*

*-페이징*

*-비밀글*

*-조회수(새로고침 해도 안올라가게)*

*-댓글(쓰기,삭제,수정, 댓글 수 표시,* 비밀 댓글,)

*-답글(쓰기,삭제,수정,비밀글,* 댓글의 답글)

*-좋아요 & 싫어요*

*-모든 글에 파일 업로드*