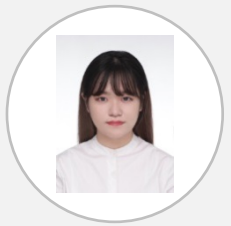


AHN SONG I

SOFTWARE

DEVELOPER

PORTPOLIO

1. Degree project

WordPress

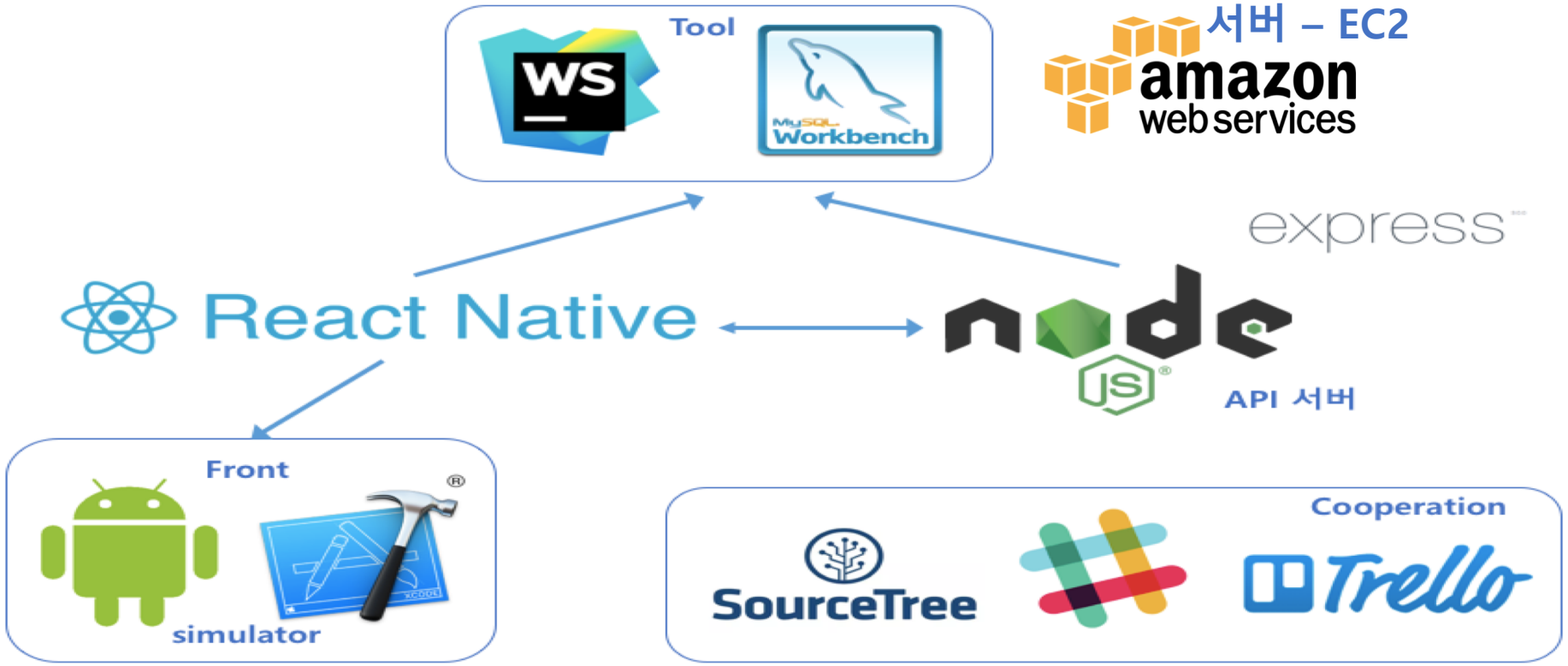
-사용 기술 : 워드프레스
-졸업 프로젝트로 진행된 웹 쇼핑몰 만들기
-플러그인(우커머스)을 이용해 쇼핑몰구현에 필요한 기술 구현
-테마를 이용해 홈페이지 꾸미기(css수정)
-졸업작품을 판매하는 사이트를 만듦으로써 실용성 추구

2. H6(group) project

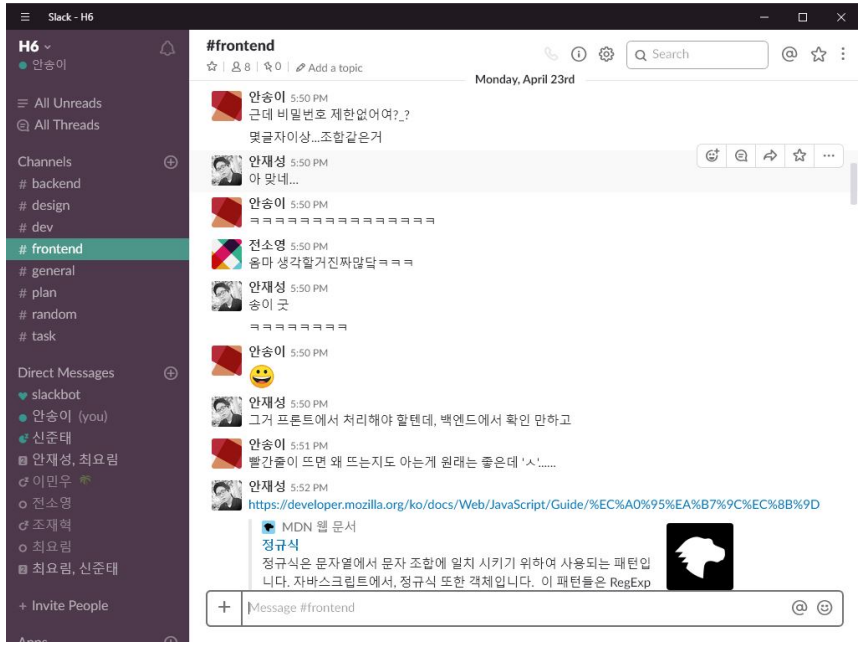
-프로젝트 목표 : 한성대학교만의 커뮤니티 앱 만들기
-사용 기술 : Node.js, React-Native, JavaScript, TypeScript, AWS
-협업 : Trello, Slack, Sourcetree
-개발진행중 : 이메일 인증과 디자인에 맞춰 프론트엔드 개발진행

-Server: 초기 api 공동 작업, 회원가입 api 구현
-Client: 회원가입 구현, 강의평가 구현중

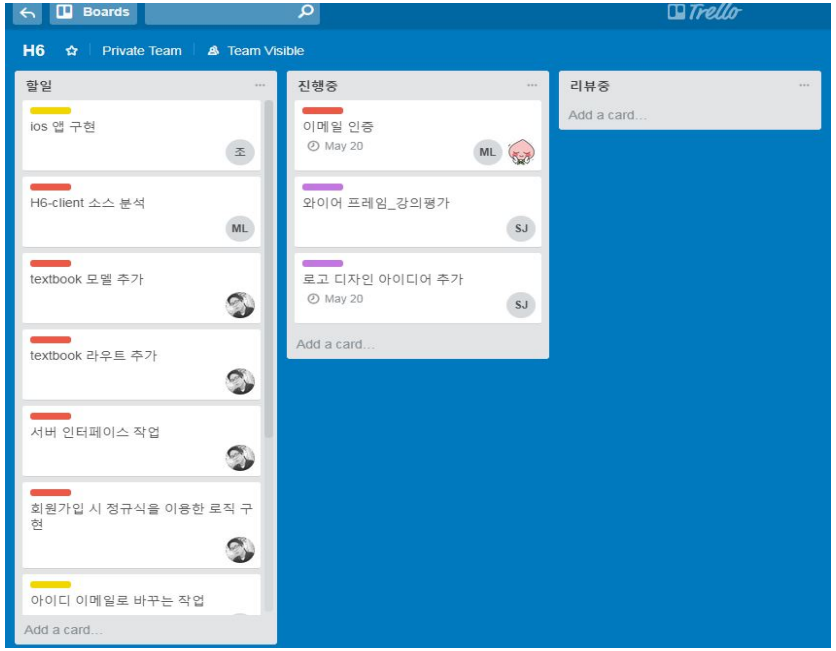
2. H6(group) project



2. H6(group) project

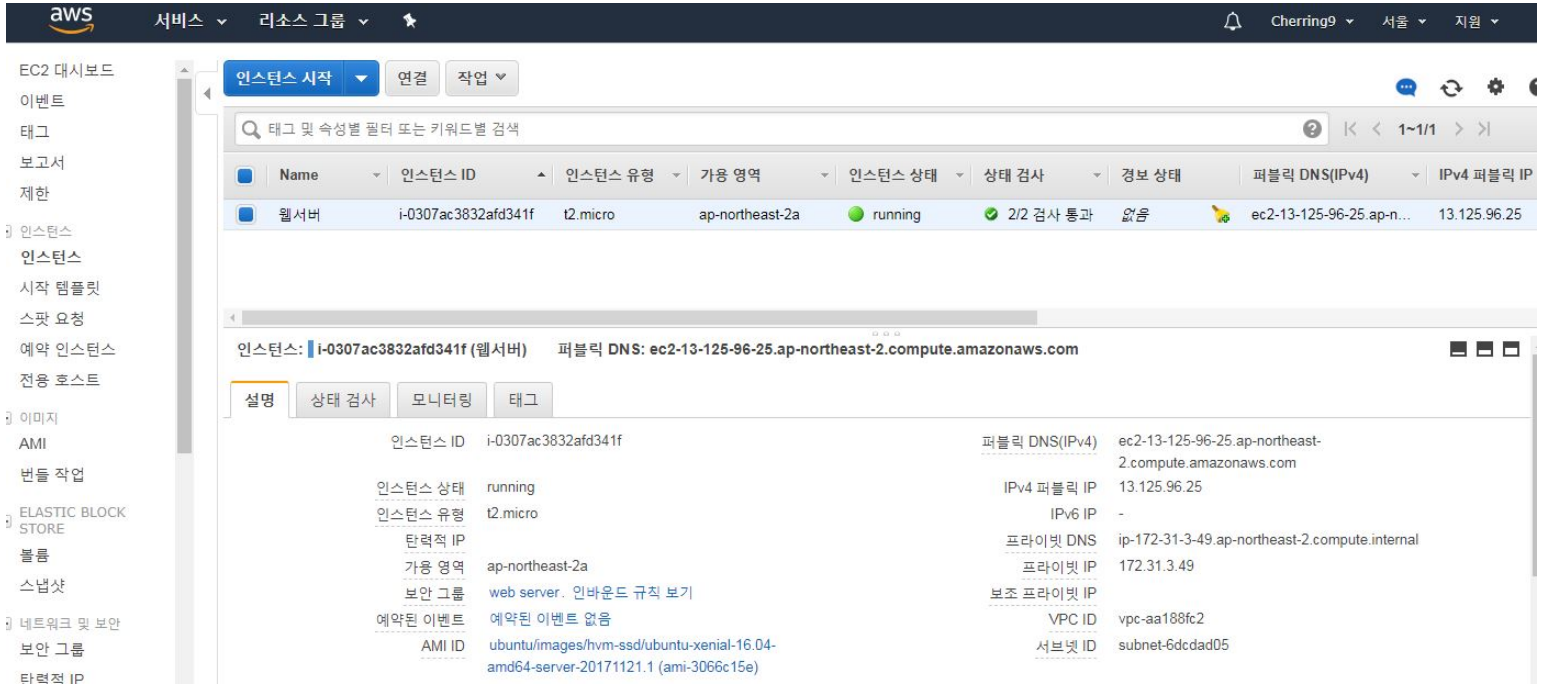


Slack, 업무 진행 상황보고, 회의



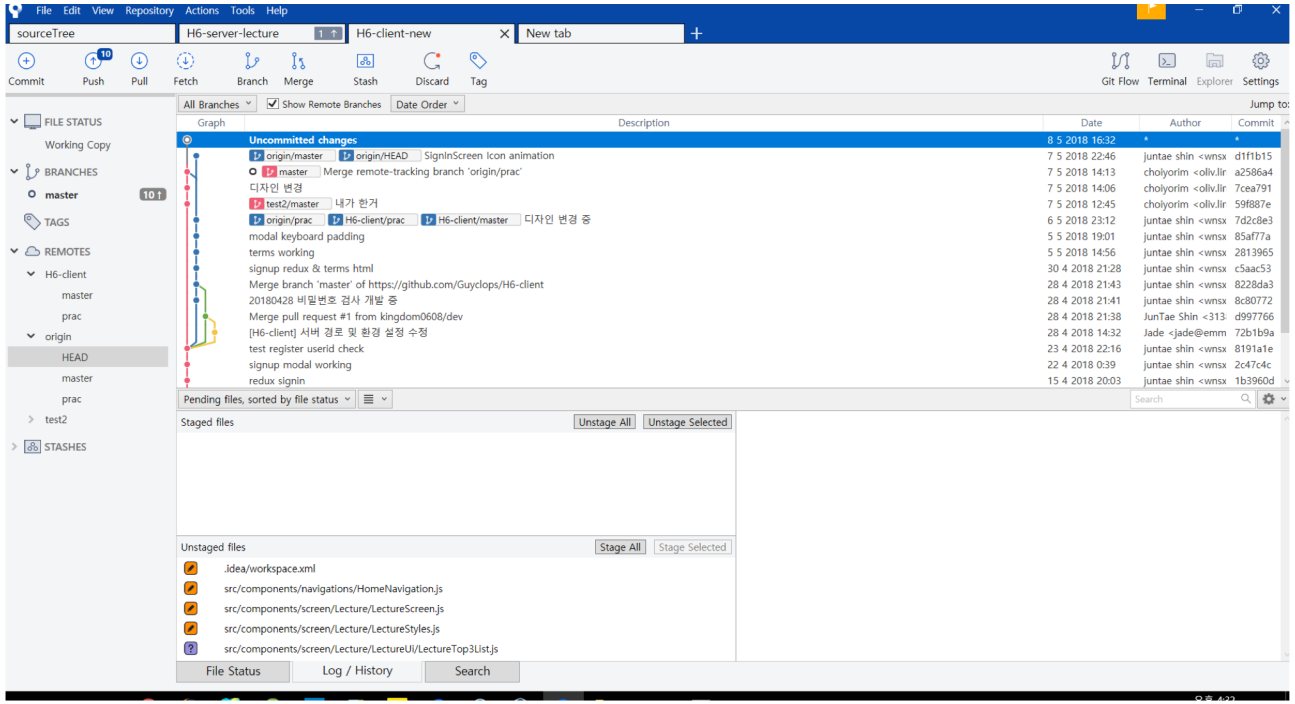
Trello, 업무 분담

2. H6(group) project



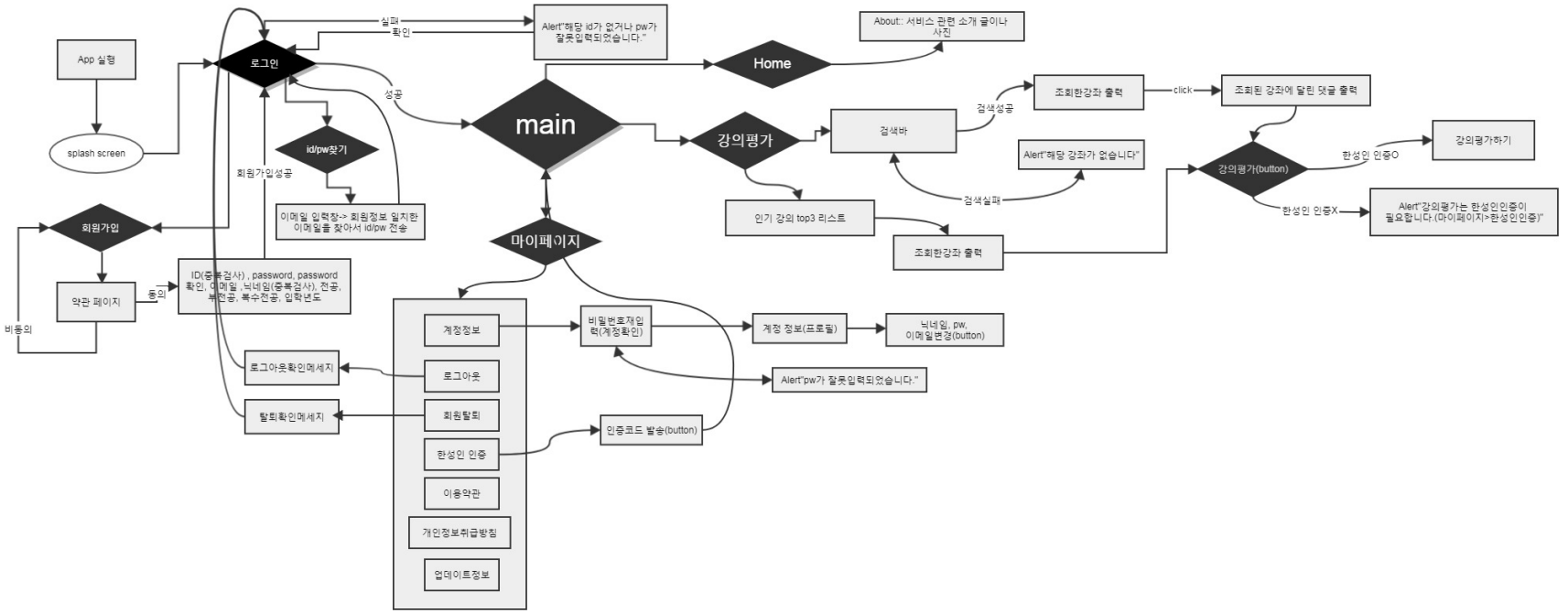
AWS, (서버작업을 의한) EC2 사용 스터디

2. H6(group) project



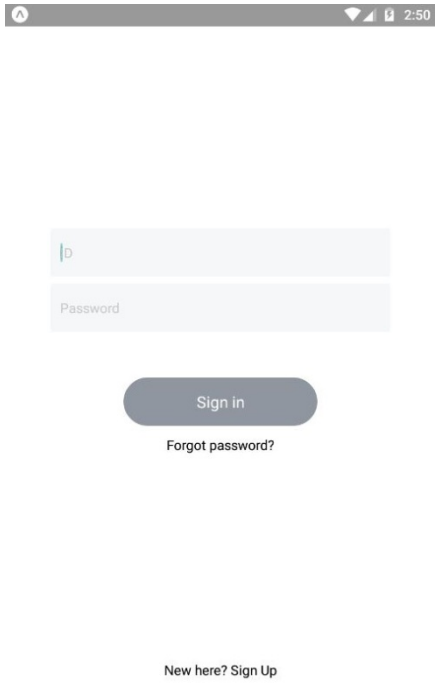
Sourcetree git 협업

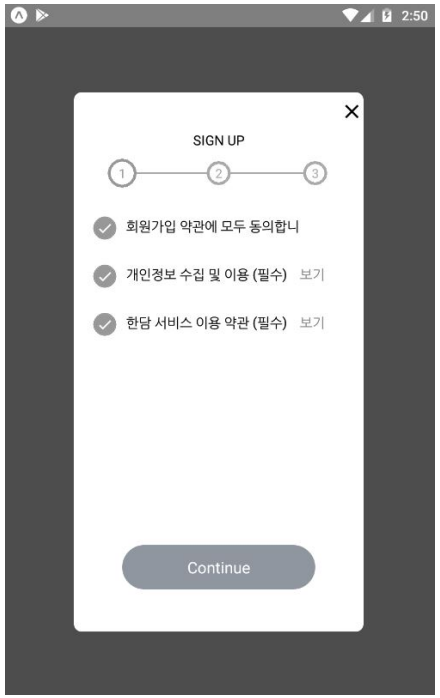
2. H6(group) project

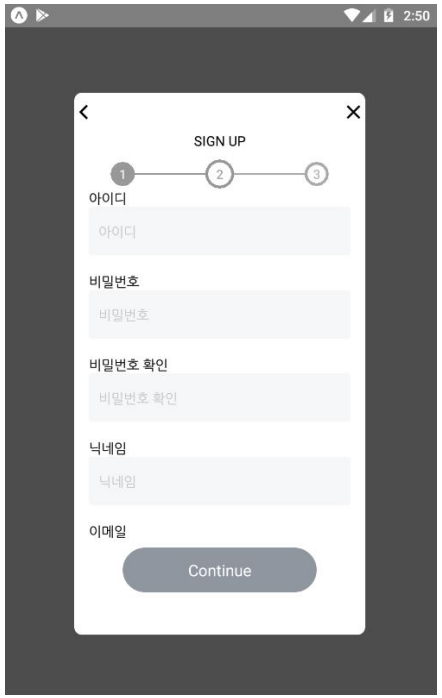


기획 - Service Flow

2. H6(group) project







Client React-native 활용하여 ios, android 모두 호환가능

2. H6(group) project

Server 구현 – 회원가입 API (User.model, User.router)

```
createUser(userData: any): Promise<any> {
    return new Promise( executor: async (resolve, reject) => {
        await pool.getConnection(async function(err, connection) {
            await connection.query(`INSERT INTO users SET ?`, [userData], function(err) {
                if (err) {
                    connection.release();
                    reject(err);
                } else {
                    connection.release();
                    resolve(userData);
                }
            })
        })
    })
}
```

Restful API, MVC 패턴에 따라 model, router, CRUD 생성

2. H6(group) project

Server 구현 – 회원가입 API (User.model, User.router)

```
export class UserRoutes {
    public userRouter: express.Router = express.Router();

    constructor() {
        this.router();
    }

    public router() {
        this.userRouter.post('/users', createUser);
        this.userRouter.get('/users', pageListUser);
        this.userRouter.get('/users/userId/:userId', getUser);
        this.userRouter.put('/users/userId/:userId', updateUser);
        this.userRouter.put('/users/userId/:userId/password', updateUserPassword);
        this.userRouter.delete('/users/userId/:userId', deleteUser);
    }
```

```
async function createUser(req, res): Promise<void> {
    const userData: any = new UserResource(req.body);
    try {
        const result: any = await user.createUser(userData);
        res.send(result);
    } catch (err) {
        res.send(err);
    }
}
```

2. H6(group) project

Redux 패턴을 이용하여 action+reducer, component 구현

```
const ROOT_URL = config.server;
const CURRENT_PAGE = 'CURRENT_PAGE';
const LECTURE_LOADING = 'LECTURE_LOADING';
const LECTURE_LIST = 'LECTURE_LIST';
const LECTURE_LIST_INIT = 'LECTURE_LIST_INIT';
const LECTURE_TOTAL = 'LECTURE_TOTAL';
const LECTURE_LIST_LENGTH = 'LECTURE_LIST_LENGTH';
const LECTURE_TEXT_VALUE = 'LECTURE_TEXT_VALUE';
const LECTURE_SEARCH_TEXT = 'LECTURE_SEARCH_TEXT';

const LECTURE_REPLY = 'LECTURE_REPLY';

const initialState = {textValue: ''...};

export const initLectureList = () => dispatch => {...};

export const onChangeTextValue = (value) => dispatch => {...};

export const handleSearchText = (value) => dispatch => {...};

export const getLectureList = (searchText, page, length) => async dispatch => {...};

// export const lectureReply = () => dispatch =>{
//     dispatch({type: LECTURE_REPLY, payload: 0});
// }

export default handleActions({...}, initialState)
```

Action 과 reducer을 합쳐 구현 한 lecture.js
reducer 를 통해 변화한 store 을 컴포넌트가 복제하여 새로 리렌더링시킨다.

2. H6(group) project

Redux 패턴을 이용하여 action+reducer, component 구현

```
class LectureScreen extends React.Component {...}

export default connect((state) => ({
        currentPage: state.lecture.currentPage,
        lectureList: state.lecture.lectureList,
        loading: state.lecture.loading,
        total: state.lecture.total,
        lectureListLength: state.lecture.lectureListLength,
        textValue: state.lecture.textValue,
        searchText: state.lecture.searchText
    }),
    (dispatch) => ({
        Lecture: bindActionCreators(lecture, dispatch)
    })
)(LectureScreen);
```

LectureScreen의
똑똑한 component
Connect는 store 등록을
해주는 메소드이다.